A VISUAL REFERENCE · Org-mode

Org-mode, one change at a time

From a first paragraph to a reproducible report. Complete source. A small change. Its actual result.

Edition 3 - original example edition: 10 October 2026. Prepared for Lukas C. Bossert. Native export and editor results.

The Org file is the canonical source. Read the HTML guide beside your work, or download its PDF and source for offline reading.

> **Last changes: 2026-10-11.** Unified tour–chapter–example IDs, numbered chapter anchors and updated direct links, downloads and comparisons. Reader pages now omit publishing instructions and highlight PDF downloads.

208
worked examples

17
topic families

1
canonical Org source

READING THE TOUR

Three ways to see a change

Document

Org source becomes exported content: HTML fragments or real PDF pages.

Editor

Source, cursor and command produce actual buffer text, not a simulated editor.

Code and data

A source block produces a result or file. Babel evaluates; tangling extracts.

Read from the beginning or jump to a stable example ID. Each example includes its complete input and captured result. Copy link preserves the address of the guide you are reading, including its current domain and path.

Scope. This edition focuses on **Doom Emacs without Evil**. Standard Org / Emacs keys are shown on every example; the keyboard reference explains non-Evil Doom prefixes and contextual bindings. Command names remain usable through `M-x`. `C` means Control, `M` Meta (usually Alt/Option), and `S` Shift. Cursor position matters: commands may behave differently inside a table, a list or a heading.

Shared defaults. Examples start with no contents list, section numbers, author, date or HTML postamble. Displayed options override these settings. Full-document examples enable explicit metadata. Agenda examples use 12 October 2026. Task completion logging is off. Each example has its own file, working directory and Org buffer; Emacs starts with `-Q`.

Comparisons. A teal stripe marks source lines changed from the named comparison. HTML Before / After buttons switch both source and result; the PDF shows the After state. Some examples change a command rather than the source, and some refactorings deliberately produce the same output.

Doom Emacs without Evil

This tour assumes Doom's evil module is disabled, with :lang org enabled. Standard Org / Emacs keys appear on every example. Doom's non-Evil leader is C-c and its localleader is C-c l; these are configurable through doom-leader-alt-key and doom-localleader-alt-key. No modal editing is required. Press C-h k before a sequence to inspect the binding in your actual buffer. The computed examples run in plain Emacs -Q; Doom-specific aliases are documented separately and are not execution tests of your configuration.

Keys	Command / prefix	Use
Keys	Command / prefix	Use
C-c	Doom leader	Global Doom command prefix; wait to see the available menu.
C-c l	Doom localleader	Mode-specific Doom commands in an Org buffer.
M-x command RET	execute-extended-command	Run any interactive command by name.
C-h k, then keys	describe-key	Inspect the current binding; Doom may use a richer help frontend.
C-h f	describe-function	Read a command's documentation.
C-h m	describe-mode	Inspect active modes and their bindings.
C-x C-f / C-x C-s	find-file / save-buffer	Open / save your .org file.
C-SPC	set-mark-command	Start a region; move point to select text.
C-_ or C-x u	undo	Undo an edit.

Doom Emacs without Evil · continued

Keyboard reference continued. Commands are available by name through M-x; context-specific keys require the indicated buffer.

Keys	Command / prefix	Use
`C-g`	`keyboard-quit`	Cancel a prompt or unfinished key sequence.
`C-u`	`universal-argument`	Prefix the next command; its meaning depends on that command.
`M-: expression RET`	`eval-expression`	Evaluate a Lisp expression shown by an inspect example.

Doom shortcuts in an Org buffer

These aliases use Doom's standard non-Evil prefixes and the :lang org module. They are read from Doom's current module source (10 October 2026); your configured keys may differ. C-c l introduces the local menu: the letters after it are ordinary sequential keypresses. The spreadsheet examples include the recalculation prefix explicitly, because the unprefixed command updates only the current row.

Keys	Command / prefix	Use
Keys	Command	Use
C-c l e	org-export-dispatch	Export menu; continue with h h for HTML or l p for PDF.
C-c l '	org-edit-special	Formula editor in a table; source editor inside a block.
C-c l b f	org-table-edit-formulas	Open table formulas directly.
C-c l b r / C-u C-c l b r	org-table-recalculate	Current row / whole table.
C-c l b R	org-table-recalculate-buffer-tables	Recalculate every table in the buffer once.
C-c l b t f / C-c l b t o	org-table-toggle-formula-debugger / org-table-toggle-coordinate-overlays	Formula debugger / coordinate grid.
C-c l b a	org-table-align	Align the current table.
C-c l t / C-c l x	org-todo / org-toggle-checkbox	Change task state / checkbox state.
C-c l q / C-c l o	org-set-tags-command / org-set-property	Edit tags / a property.

Doom shortcuts in an Org buffer · continued

Keyboard reference continued. Commands are available by name through M-x; context-specific keys require the indicated buffer.

Keys	Command / prefix	Use
C-c l d s / C-c l d d	org-schedule / org-deadline	Schedule / deadline.
C-c l d t / C-c l d T	org-time-stamp / org-time-stamp-inactive	Active / inactive timestamp.
C-c l l l / C-c n l	org-insert-link / org-store-link	Insert / store a link.
C-c n a a / C-c n t	org-agenda / org-todo-list	Agenda view / unfinished tasks.
C-c l #	org-update-statistics-cookies	Refresh completion statistics.
C-c l s n / C-c l s N	org-narrow-to-subtree / widen	Focus on a subtree / show the full buffer.
C-c l s s m	org-sparse-tree	Interactive tag/property match.

Structure, text and links

C means Control, M means Meta (usually Alt/Option), S means Shift. RET is Return; arrow names refer to arrow keys. A sequence such as C-c C-e h h is pressed in order. Org's editing commands depend on point: the same arrow shortcut can edit a heading, list item or table. On keyboards where C-SPC or modifier-arrow combinations are intercepted, use M-x with the command name.

Keys	Command / prefix	Use
Keys	Command	Use
TAB / S-TAB	org-cycle / org-shifttab	Fold current heading / cycle the whole outline.
M-RET	org-meta-return	Insert a heading or list item.
M-LEFT / M-RIGHT	org-metaleft / org-metaright	Promote / demote heading or item at point.
M-S-LEFT / M-S-RIGHT	org-shiftmetaleft / org-shiftmetaright	Change the level of an entire subtree.
M-UP / M-DOWN	org-metaup / org-metadown	Move the current subtree or item.
C-c C-n / C-c C-p	org-next-visible-heading / org-previous-visible-heading	Navigate visible headings.
C-c C-u	outline-up-heading	Go to the parent heading.
C-c ^, then a	org-sort	Sort sibling headings alphabetically at a heading.
C-x n s / C-x n w	org-narrow-to-subtree / widen	Focus on a subtree / restore the whole buffer.

Structure, text and links · continued

Keyboard reference continued. Commands are available by name through M-x; context-specific keys require the indicated buffer.

Keys	Command / prefix	Use
`C-c C-x C-f`	`org-emphasize`	Add an emphasis marker to selected text; choose *, /, ~, = or +.
`C-c C-l / C-c C-o`	`org-insert-link / org-open-at-point`	Insert or edit / follow a link.
`C-c C-x f`	`org-footnote-action`	Create a footnote or jump between reference and definition.
`C-c C-x C-v`	`org-toggle-inline-images`	Show or hide inline image previews.

Tables and formula workflow

Inside a table, C-c C-c aligns it. On a #+TBLFM line, the same keys apply that formula set. Use C-c ' to edit formulas in a separate buffer: each assignment gets its own line, so long formulas remain manageable. If you see #ERROR, toggle the debugger with C-c {, then recalculate and inspect the substitutions. Arrow shortcuts act on whole rows or columns only when point is inside the table.

Keys	Command / prefix	Use
Keys	Command	Use
TAB / S-TAB / RET	org-table-next-field / org-table-previous-field / org-table-next-row	Align and move through fields.
C-c C-c	org-ctrl-c-ctrl-c	Align table; on #+TBLFM, apply that formula line.
M-LEFT / M-RIGHT	org-table-move-column-left / org-table-move-column-right	Move the current column.
M-UP / M-DOWN	org-table-move-row-up / org-table-move-row-down	Move the current row.
M-S-RIGHT / M-S-LEFT	org-table-insert-column / org-table-delete-column	Insert / delete a column.
M-S-DOWN / M-S-UP	org-table-insert-row / org-table-kill-row	Insert / delete a row.
C-c -	org-table-insert-hline	Insert a horizontal separator below the row.
C-c = / C-u C-c =	org-table-eval-formula	Edit a column / field formula.
C-c * / C-u C-c *	org-table-recalculate	Recalculate current row / whole table.

Tables and formula workflow · continued

Keyboard reference continued. Commands are available by name through M-x; context-specific keys require the indicated buffer.

Keys	Command / prefix	Use
`C-u C-u C-c *`	`org-table-iterate`	Recalculate repeatedly until stable or the limit is reached.
`C-c '`	`org-table-edit-formulas`	Open the formula editor when point is inside a table.
`C-c C-c / C-u C-c C-c`	`org-table-fedit-finish`	In formula editor: save / save and recalculate.
`C-c C-q`	`org-table-fedit-abort`	In formula editor: leave without applying changes.
`C-c ? / C-c }`	`org-table-field-info / org-table-toggle-coordinate-overlays`	Show current field coordinates / coordinate grid.
`C-c {`	`org-table-toggle-formula-debugger`	Toggle formula debugging, then recalculate.

Tasks, dates, agenda and properties

Agenda commands read org-agenda-files. To include your current file temporarily, use C-c [; use C-c] to remove it. Day-view keys below apply inside the agenda buffer. The guide fixes its agenda examples to 12 October 2026; your interactive agenda normally opens at the current date. M-x entry points work even if you have customized Doom's global menu.

Keys	Command / prefix	Use
Keys	Command	Use
C-c C-t	org-todo	Change the TODO keyword of a heading.
C-c ,	org-priority	Set a heading's priority.
M-S-RET	org-insert-todo-heading	New TODO heading; in a list, a checkbox item.
C-c C-c	org-ctrl-c-ctrl-c	Toggle checkbox at point / refresh statistics cookie.
C-c . / C-c !	org-time-stamp / org-time-stamp-inactive	Insert active / inactive timestamp.
C-c C-s / C-c C-d	org-schedule / org-deadline	Set scheduled date / deadline.
C-c [/ C-c]	org-agenda-file-to-front / org-remove-file	Add / remove current file from agenda files.
M-x org-agenda RET a	org-agenda	Open the agenda view.
d / w / j	org-agenda-day-view / org-agenda-week-view / org-agenda-goto-date	In agenda: day / week / chosen date.

Tasks, dates, agenda and properties · continued

Keyboard reference continued. Commands are available by name through M-x; context-specific keys require the indicated buffer.

Keys	Command / prefix	Use
`n / p / RET / q`	`org-agenda-next-line / org-agenda-previous-line / org-agenda-switch-to / org-agenda-quit`	Navigate / visit item / quit agenda.
`M-x org-agenda RET t`	`org-todo-list`	Show unfinished tasks across agenda files.
`C-c C-q`	`org-set-tags-command`	Add or remove heading tags.
`C-c C-x p`	`org-set-property`	Set a named property on the current heading.
`C-c / m`	`org-sparse-tree`	Show entries matching a tag/property expression.

Export and Babel

Use the export dispatcher from the Org buffer. For a body-only HTML fragment like the guide previews, enable C-b in that dispatcher before h h; for a complete document, leave body-only off. The PDF examples require a working LaTeX installation. In Doom, enable the required Babel language support through its Org module/ configuration. Execute only code you intend to run; tangling writes source files without running them.

Keys	Command / prefix	Use
Keys	Command	Use
C-c C-e	org-export-dispatch	Open the export menu.
C-c C-e h h / h o	org-html-export-to-html	From dispatcher: HTML file / HTML file and browser.
C-c C-e l l / l p / l o	org-latex-export-to-latex / org-latex-export-to-pdf	LaTeX source / PDF / PDF and viewer.
C-b / C-s	Export dispatcher toggles	Body-only / current subtree; press within the dispatcher.
C-c C-, s	org-insert-structure-template	Insert a source block; enter the language.
C-c '	org-edit-special	Edit a source block in its language mode.
C-c ' / C-c C-k	org-edit-src-exit / org-edit-src-abort	In source editor: accept changes / abort editing.
C-c C-c	org-babel-execute-src-block (contextual dispatch)	Evaluate the source block at point.
C-c C-v b	org-babel-execute-buffer	Evaluate source blocks throughout the current buffer.

Export and Babel · continued

Keyboard reference continued. Commands are available by name through M-x; context-specific keys require the indicated buffer.

Keys	Command / prefix	Use
`C-c C-v t`	`org-babel-tangle`	Extract tangled files from the buffer.
`C-c C-v v`	`org-babel-expand-src-block`	Inspect an expanded source block without executing it.
`C-c C-v g`	`org-babel-goto-named-src-block`	Jump to a named source block.
`C-c C-v r`	`org-babel-goto-named-result`	Jump to a named result.

ORG-01-000 · 8 EXAMPLES

First document

Begin with text. Introduce structure and metadata only when they change the result.

ORG-01-001 Start with a paragraph

ORG-01-002 Separate two paragraphs

ORG-01-003 Add a heading

ORG-01-004 Nest a heading

ORG-01-005 Add a document title

ORG-01-006 Add an author

ORG-01-007 Choose an explicit date

ORG-01-008 Add a short abstract

Start with a paragraph

KEYS (NO EVIL) C-c C-e h h · export HTML (org-export-dispatch)
DOOM ALIAS C-c l e h h · export HTML

An ordinary line of text is already a valid Org document.

COMPLETE ORG SOURCE

```
Research begins with a question.
```

HTML EXPORT

Research begins with a question.

Generated by native Org. The frame and typography belong to this guide.

Plain text

No wrapper or document environment is required.

Separate two paragraphs

Compare with ORG-01-001 · Start with a paragraph

KEYS (NO EVIL) C-c C-e h h · export HTML (org-export-dispatch)
DOOM ALIAS C-c l e h h · export HTML

A blank line starts a new paragraph.

COMPLETE ORG SOURCE

```
Research begins with a question.

Write down what you observe.
```

HTML EXPORT

Research begins with a question.

Write down what you observe.

Generated by native Org. The frame and typography belong to this guide.

`Blank line`	Separates paragraphs; a single source newline normally does not.

Add a heading

Compare with ORG-01-002 · Separate two paragraphs

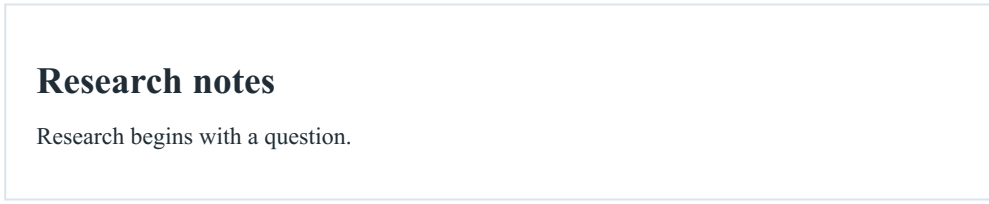
KEYS (NO EVIL) M-RET · insert heading; M-RIGHT · demote; C-c C-e h h · export
DOOM ALIAS C-c l e h h · export HTML

A star followed by a space introduces a top-level heading.

COMPLETE ORG SOURCE

```
* Research notes
Research begins with a question.
```

HTML EXPORT



Generated by native Org. The frame and typography belong to this guide.

*	Level-one heading; the space after the star matters.

Nest a heading

Compare with ORG-01-003 · Add a heading

KEYS (NO EVIL) M-RET · insert heading; M-RIGHT · demote; C-c C-e h h · export
DOOM ALIAS C-c l e h h · export HTML

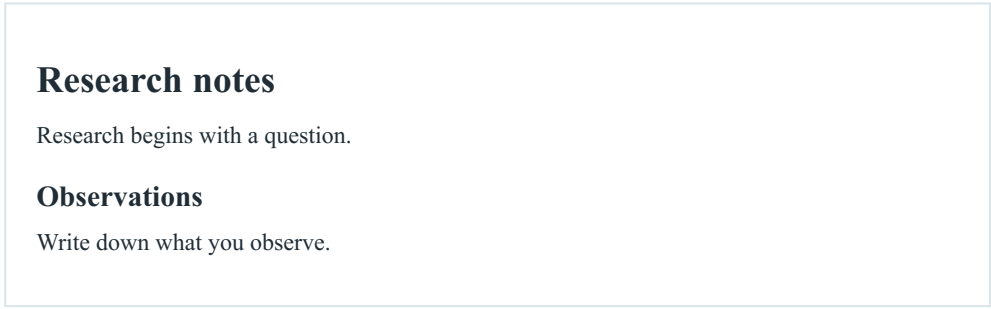
Add a second star to make a child heading.

COMPLETE ORG SOURCE

```
* Research notes
Research begins with a question.

** Observations
Write down what you observe.
```

HTML EXPORT



Research notes

Research begins with a question.

Observations

Write down what you observe.

Generated by native Org. The frame and typography belong to this guide.

**	Level-two heading inside the preceding level-one section.

Add a document title

Compare with ORG-01-004 · Nest a heading

KEYS (NO EVIL) C-c C-e h h · export HTML (org-export-dispatch)
DOOM ALIAS C-c l e h h · export HTML

The title is document metadata, separate from the outline. This example uses a full HTML document so its title is visible.

COMPLETE ORG SOURCE

```
#+TITLE: Field notebook
* Research notes
Research begins with a question.
```

FULL HTML DOCUMENT

Field notebook

Research notes

Research begins with a question.

Generated by native Org. The frame and typography belong to this guide.

`#+TITLE:`	Document title; the guide normally previews only the document body.

Add an author

Compare with ORG-01-005 · Add a document title

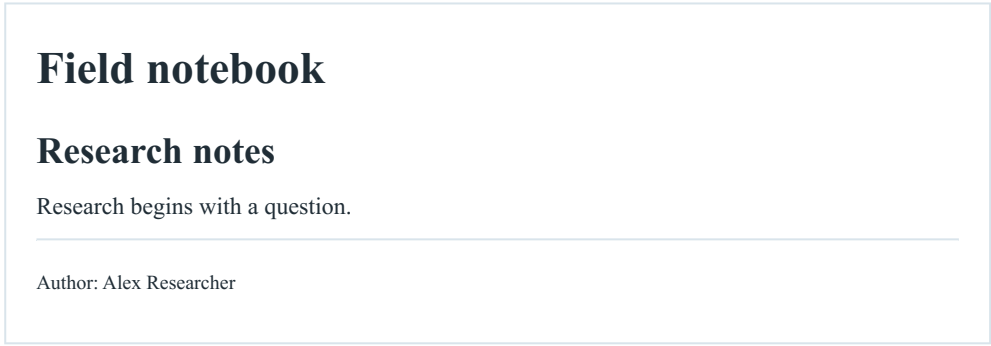
KEYS (NO EVIL) C-c C-e h h · export HTML (org-export-dispatch)
DOOM ALIAS C-c l e h h · export HTML

Add an author to the same document. The full HTML export puts author metadata in its postamble.

COMPLETE ORG SOURCE

```
#+TITLE: Field notebook
#+AUTHOR: Alex Researcher
* Research notes
Research begins with a question.
```

FULL HTML DOCUMENT



Generated by native Org. The frame and typography belong to this guide.

`#+AUTHOR:`	Author string used by the exporter.

Choose an explicit date

Compare with ORG-01-006 · Add an author

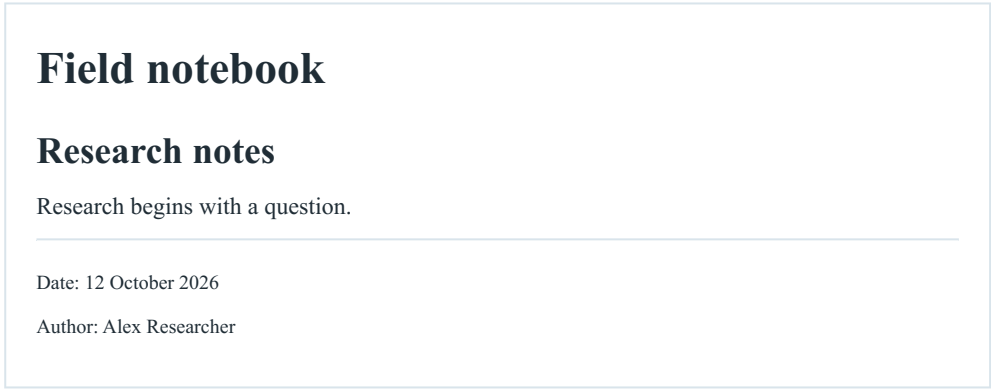
KEYS (NO EVIL) C-c C-e h h · export HTML (org-export-dispatch)
DOOM ALIAS C-c l e h h · export HTML

An explicit date makes the document independent of the export day.

COMPLETE ORG SOURCE

```
#+TITLE: Field notebook
#+AUTHOR: Alex Researcher
#+DATE: 12 October 2026
* Research notes
Research begins with a question.
```

FULL HTML DOCUMENT



Field notebook

Research notes

Research begins with a question.

Date: 12 October 2026

Author: Alex Researcher

Generated by native Org. The frame and typography belong to this guide.

`#+DATE:`	Literal document date; an empty value suppresses the date.

Add a short abstract

KEYS (NO EVIL) C-c C-e h h · export HTML (org-export-dispatch)
DOOM ALIAS C-c l e h h · export HTML

An abstract block identifies introductory summary content.

COMPLETE ORG SOURCE

```
#+TITLE: Field notebook
#+BEGIN_ABSTRACT
A compact record of a small study.
#+END_ABSTRACT
* Research notes
Research begins with a question.
```

HTML EXPORT

A compact record of a small study.

Research notes

Research begins with a question.

Generated by native Org. The frame and typography belong to this guide.

`#+BEGIN_ABSTRACT`	Starts a special block named abstract.
`#+END_ABSTRACT`	Closes the block; HTML exports a container, while LaTeX uses an environment.

ORG-02-000 · 10 EXAMPLES

Text and lists

Change a small piece of syntax and inspect the rendered text beside it.

ORG-02-001 Make a word bold

ORG-02-002 Use italic emphasis

ORG-02-003 Show inline code

ORG-02-004 Show literal text

ORG-02-005 Strike out an old statement

ORG-02-006 Create a bullet list

ORG-02-007 Nest a list item

ORG-02-008 Number the items

ORG-02-009 Use a description list

ORG-02-010 Preserve line breaks in verse

Make a word bold

KEYS (NO EVIL) C-c C-x C-f * · org-emphasize (select text first); C-c C-e h h · export
DOOM ALIAS C-c l e h h · export HTML

Use paired asterisks around a word.

COMPLETE ORG SOURCE

```
A *reliable* workflow.
```

HTML EXPORT

A **reliable** workflow.

Generated by native Org. The frame and typography belong to this guide.

`*text*`	Emphasis exported as strong text.

Use italic emphasis

KEYS (NO EVIL) C-c C-x C-f / · org-emphasize (select text first); C-c C-e h h · export
DOOM ALIAS C-c l e h h · export HTML

Replace the emphasis markers.

COMPLETE ORG SOURCE

```
A /reliable/ workflow.
```

HTML EXPORT

A *reliable* workflow.

Generated by native Org. The frame and typography belong to this guide.

`/text/`	Emphasis exported as italic text.

Show inline code

KEYS (NO EVIL) C-c C-x C-f ~ · org-emphasize (select text first); C-c C-e h h · export
DOOM ALIAS C-c l e h h · export HTML

Mark short pieces of code inside a sentence.

COMPLETE ORG SOURCE

```
Run ~make~ to rebuild.
```

HTML EXPORT

Run make to rebuild.

Generated by native Org. The frame and typography belong to this guide.

`~text~`	Inline code; internal markup is not interpreted.

Show literal text

KEYS (NO EVIL) C-c C-x C-f = · org-emphasize (select text first); C-c C-e h h · export
DOOM ALIAS C-c l e h h · export HTML

Protect syntax that would otherwise be interpreted.

COMPLETE ORG SOURCE

```
Keep =*this*= literal.
```

HTML EXPORT

Keep *this* literal.

Generated by native Org. The frame and typography belong to this guide.

`=text=`	Verbatim text; the stars remain visible.

Strike out an old statement

KEYS (NO EVIL) C-c C-x C-f + · org-emphasize (select text first); C-c C-e h h · export
DOOM ALIAS C-c l e h h · export HTML

Mark a superseded phrase.

COMPLETE ORG SOURCE

```
The result is +final+ provisional.
```

HTML EXPORT

The result is ~~final~~ provisional.

Generated by native Org. The frame and typography belong to this guide.

`+text+`	Strike-through formatting.

Create a bullet list

KEYS (NO EVIL) M-RET · new item; M-RIGHT / M-LEFT · change nesting; C-c C-e h h · export
DOOM ALIAS C-c l e h h · export HTML

Start each item with a hyphen and a space.

COMPLETE ORG SOURCE

```
Materials:
- Notebook
- Pencil
```

HTML EXPORT

Materials:

- Notebook
- Pencil

Generated by native Org. The frame and typography belong to this guide.

-	Unordered list item.

Nest a list item

KEYS (NO EVIL) M-RET · new item; M-RIGHT / M-LEFT · change nesting; C-c C-e h h · export
DOOM ALIAS C-c l e h h · export HTML

Indent an item beneath its parent.

COMPLETE ORG SOURCE

```
Materials:
- Notebook
  - Blank pages
- Pencil
```

HTML EXPORT

Materials:

- Notebook
  - Blank pages
- Pencil

Generated by native Org. The frame and typography belong to this guide.

`Indentation` Keeps the child item inside the preceding list item.

Number the items

KEYS (NO EVIL) M-RET · new item; M-RIGHT / M-LEFT · change nesting; C-c C-e h h · export
DOOM ALIAS C-c l e h h · export HTML

Use ordered markers when sequence matters.

COMPLETE ORG SOURCE

```
1. Collect
2. Check
3. Publish
```

HTML EXPORT

1. Collect
2. Check
3. Publish

Generated by native Org. The frame and typography belong to this guide.

1. Starts an ordered list; Org can renumber it while editing.

Use a description list

KEYS (NO EVIL) M-RET · new item; M-RIGHT / M-LEFT · change nesting; C-c C-e h h · export
DOOM ALIAS C-c l e h h · export HTML

Associate terms with definitions.

COMPLETE ORG SOURCE

```
- Input :: Raw observations
- Output :: Checked observations
```

HTML EXPORT

Input
: Raw observations

Output
: Checked observations

Generated by native Org. The frame and typography belong to this guide.

`term :: definition` — Description-list syntax.

Preserve line breaks in verse

KEYS (NO EVIL) C-c C-e h h · export HTML (org-export-dispatch)
DOOM ALIAS C-c l e h h · export HTML

A verse block preserves intentional line breaks and indentation.

COMPLETE ORG SOURCE

```
#+BEGIN_VERSE
First line
  Second line
Third line
#+END_VERSE
```

HTML EXPORT

First line
  Second line
Third line

Generated by native Org. The frame and typography belong to this guide.

`verse block` For text whose line structure matters.

ORG-03-000 · 8 EXAMPLES

Outline and navigation

These are real buffer transformations. A command operates at the stated cursor location.

ORG-03-001 Demote one heading

ORG-03-002 Demote a whole subtree

ORG-03-003 Promote a whole subtree

ORG-03-004 Move a subtree down

ORG-03-005 Sort sibling headings

ORG-03-006 Focus on one subtree

ORG-03-007 Widen the buffer again

ORG-03-008 Fold a subtree

Demote one heading

KEYS (NO EVIL) M-RIGHT · org-do-demote

Make Methods one level deeper. Its child is not demoted with it.

COMPLETE ORG SOURCE

```
* Study
Introduction.
** Methods
Collect data.
*** Instruments
A sensor.
** Results
Checked data.
* Appendix
Details.
```

EDITOR BUFFER

```
* Study
Introduction.
*** Methods
Collect data.
*** Instruments
A sensor.
** Results
Checked data.
* Appendix
Details.
```

Actual text captured from Emacs; syntax colours are applied by the guide.

ACTION

M-RIGHT · org-do-demote

`(org-do-demote)`

Cursor at the beginning of: ** Methods

`org-do-demote`	Moves the current heading one level down.
`M-RIGHT`	Standard Emacs binding; command names also work through M-x.

Demote a whole subtree

Compare with ORG-03-001 · Demote one heading

KEYS (NO EVIL) M-S-RIGHT · org-demote-subtree
DOOM ALIAS C-c l s l · demote subtree

Move Methods and its descendants together.

COMPLETE ORG SOURCE

```
* Study
Introduction.
** Methods
Collect data.
*** Instruments
A sensor.
** Results
Checked data.
* Appendix
Details.
```

EDITOR BUFFER

```
* Study
Introduction.
*** Methods
Collect data.
**** Instruments
A sensor.
** Results
Checked data.
* Appendix
Details.
```

Actual text captured from Emacs; syntax colours are applied by the guide.

ACTION

M-S-RIGHT · org-demote-subtree

`(org-demote-subtree)`

Cursor at the beginning of: ** Methods

org-demote-subtree	Preserves relative depths inside the subtree.
M-S-RIGHT	Meta + Shift + Right in standard bindings.

Promote a whole subtree

Compare with ORG-03-002 · Demote a whole subtree

KEYS (NO EVIL) M-S-LEFT · org-promote-subtree
DOOM ALIAS C-c l s h · promote subtree

Move Methods and its child one level up.

COMPLETE ORG SOURCE

```
* Study
Introduction.
** Methods
Collect data.
*** Instruments
A sensor.
** Results
Checked data.
* Appendix
Details.
```

EDITOR BUFFER

```
* Study
Introduction.
* Methods
Collect data.
** Instruments
A sensor.
** Results
Checked data.
* Appendix
Details.
```

Actual text captured from Emacs; syntax colours are applied by the guide.

ACTION

M-S-LEFT · org-promote-subtree

`(org-promote-subtree)`

Cursor at the beginning of: ** Methods

`org-promote-subtree`	Promotes the heading and every descendant.
`M-S-LEFT`	Standard binding.

Move a subtree down

KEYS (NO EVIL) M-DOWN · org-move-subtree-down
DOOM ALIAS C-c l s j · move subtree down

Swap Methods with the next sibling, Results. The Instruments child moves with Methods.

COMPLETE ORG SOURCE

```
* Study
Introduction.
** Methods
Collect data.
*** Instruments
A sensor.
** Results
Checked data.
* Appendix
Details.
```

EDITOR BUFFER

```
* Study
Introduction.
** Results
Checked data.
** Methods
Collect data.
*** Instruments
A sensor.
* Appendix
Details.
```

Actual text captured from Emacs; syntax colours are applied by the guide.

ACTION

M-DOWN · org-move-subtree-down

`(org-move-subtree-down)`

Cursor at the beginning of: ** Methods

`org-move-subtree-down`	Moves across one sibling without losing child content.

Sort sibling headings

KEYS (NO EVIL) C-c ^, then a · org-sort-entries
DOOM ALIAS C-c l s S, then a · sort

Sort the direct children of Study alphabetically.

COMPLETE ORG SOURCE

```
* Study
** Zebra
Last alphabetically.
** Alpha
First alphabetically.
** Beta
In the middle.
```

EDITOR BUFFER

```
* Study
** Alpha
First alphabetically.
** Beta
In the middle.
** Zebra
Last alphabetically.
```

Actual text captured from Emacs; syntax colours are applied by the guide.

ACTION

C-c ^, then a · org-sort-entries

`(org-sort-entries nil ?a)`

Cursor at the beginning of: * Study

`org-sort-entries`	Sorts sibling entries.
`?a`	The Lisp character a requests ascending alphabetical order.

Focus on one subtree

KEYS (NO EVIL) C-x n s · org-narrow-to-subtree
DOOM ALIAS C-c l s n · narrow to subtree

Narrow the buffer to Methods. The other text still exists but is outside the accessible region.

COMPLETE ORG SOURCE

```
* Study
Introduction.
** Methods
Collect data.
*** Instruments
A sensor.
** Results
Checked data.
* Appendix
Details.
```

ACTION

C-x n s · org-narrow-to-subtree

```
(org-narrow-to-subtree)
```

Cursor at the beginning of: ** Methods

EDITOR BUFFER

```
** Methods
Collect data.
*** Instruments
A sensor.
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`org-narrow-to-subtree`	Restricts the accessible buffer region.
`C-x n s`	Standard binding; C-x n w widens again.

Widen the buffer again

Compare with ORG-03-006 · Focus on one subtree

KEYS (NO EVIL) C-x n s, then C-x n w
DOOM ALIAS C-c l s n, then C-c l s N · narrow / widen

First narrow to Methods, then widen. The full buffer becomes accessible again.

COMPLETE ORG SOURCE

```
* Study
Introduction.
** Methods
Collect data.
*** Instruments
A sensor.
** Results
Checked data.
* Appendix
Details.
```

EDITOR BUFFER

```
* Study
Introduction.
** Methods
Collect data.
*** Instruments
A sensor.
** Results
Checked data.
* Appendix
Details.
```

Actual text captured from Emacs; syntax colours are applied by the guide.

ACTION

C-x n s, then C-x n w

```
(org-narrow-to-subtree)
(widen)
```

Cursor at the beginning of: ** Methods

`widen`	Removes narrowing without editing the document.

Fold a subtree

KEYS (NO EVIL) TAB · org-cycle

Hide the body and descendants of Methods. The preview serializes visible buffer text and uses an ellipsis for each hidden region.

COMPLETE ORG SOURCE

```
* Study
Introduction.
** Methods
Collect data.
*** Instruments
A sensor.
** Results
Checked data.
* Appendix
Details.
```

VISIBLE EDITOR TEXT

```
* Study
Introduction.
** Methods …
** Results
Checked data.
* Appendix
Details.
```

Actual text captured from Emacs; syntax colours are applied by the guide.

ACTION

TAB · org-cycle

```
(org-fold-show-all)
(org-cycle)
```

Cursor at the beginning of: ** Methods

`org-cycle`	Cycles the visibility of a heading.
`TAB`	With point on a heading, cycle its visibility.

ORG-04-000 · 8 EXAMPLES

Links and references

Make destinations explicit and keep internal references stable when headings change.

ORG-04-001 Insert a web link

ORG-04-002 Give a link readable text

ORG-04-003 Link to another heading

ORG-04-004 Use a stable custom target

ORG-04-005 Link to an inline target

ORG-04-006 Add a footnote

ORG-04-007 Write an inline footnote

ORG-04-008 Link to a local file

Insert a web link

KEYS (NO EVIL) C-c C-l · insert/edit link; C-c C-o · open link; C-c C-e h h · export
DOOM ALIAS C-c l l l · insert/edit link; C-c n l · store link

A bracket link can display its destination directly.

COMPLETE ORG SOURCE

```
[[https://orgmode.org/]]
```

HTML EXPORT

https://orgmode.org/

Generated by native Org. The frame and typography belong to this guide.

`[[URL]]`	A link without a separate description.

Give a link readable text

KEYS (NO EVIL) C-c C-l · insert/edit link; C-c C-o · open link; C-c C-e h h · export
DOOM ALIAS C-c l l l · insert/edit link; C-c n l · store link

Add a description while keeping the same destination.

COMPLETE ORG SOURCE

```
[[https://orgmode.org/][Org website]]
```

HTML EXPORT

Org website

Generated by native Org. The frame and typography belong to this guide.

`[[URL][description]]`	The description is shown; the URL remains the destination.

Link to another heading

KEYS (NO EVIL) C-c C-l · insert/edit link; C-c C-o · open link; C-c C-e h h · export
DOOM ALIAS C-c l l l · insert/edit link; C-c n l · store link

A star inside the link targets a heading by its text.

COMPLETE ORG SOURCE

```
See [[*Results][the findings]].

* Methods
Collect data.
* Results
The findings are ready.
```

HTML EXPORT

See the findings.

Methods

Collect data.

Results

The findings are ready.

Generated by native Org. The frame and typography belong to this guide.

`[[*Heading]]`	A fuzzy link to a heading in the same file.

Use a stable custom target

KEYS (NO EVIL) C-c C-l · insert/edit link; C-c C-o · open link; C-c C-e h h · export
DOOM ALIAS C-c l l l · insert/edit link; C-c n l · store link

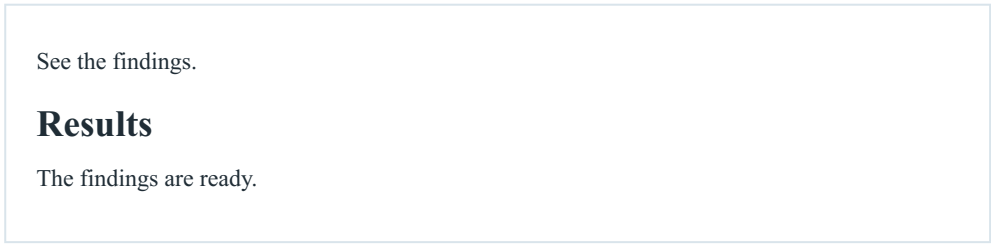
The link survives a change to the displayed heading text.

COMPLETE ORG SOURCE

```
See [[#findings][the findings]].

* Results
:PROPERTIES:
:CUSTOM_ID: findings
:END:
The findings are ready.
```

HTML EXPORT



Generated by native Org. The frame and typography belong to this guide.

`CUSTOM_ID`	Explicit local target for this heading.
`[[#findings]]`	Links to that custom target.

Link to an inline target

KEYS (NO EVIL) C-c C-l · insert/edit link; C-c C-o · open link; C-c C-e h h · export
DOOM ALIAS C-c l l l · insert/edit link; C-c n l · store link

An explicit target can sit inside ordinary document content.

COMPLETE ORG SOURCE

```
See [[sampling][the sampling step]].

<<sampling>>
Take three readings.
```

HTML EXPORT

See the sampling step.

Take three readings.

Generated by native Org. The frame and typography belong to this guide.

`<<sampling>>`	Creates a named target.
`[[sampling][...]]`	References that target.

Add a footnote

KEYS (NO EVIL) C-c C-x f · org-footnote-action; C-c C-e h h · export
DOOM ALIAS C-c l f · footnote action

Separate supporting detail from the main sentence.

COMPLETE ORG SOURCE

```
A small sample[fn:sample] is enough here.

[fn:sample] Three observations are used for illustration.
```

HTML EXPORT

A small sample[1] is enough here.

Footnotes:

1

Three observations are used for illustration.

Generated by native Org. The frame and typography belong to this guide.

`[fn:sample]`	Named footnote reference and definition marker.

Write an inline footnote

KEYS (NO EVIL) C-c C-x f · org-footnote-action; C-c C-e h h · export
DOOM ALIAS C-c l f · footnote action

Put the definition directly beside its reference.

COMPLETE ORG SOURCE

```
A small sample[fn::Three observations are used for illustration.] is enough here.
```

HTML EXPORT

A small sample[1] is enough here.

Footnotes:

1

Three observations are used for illustration.

Generated by native Org. The frame and typography belong to this guide.

`[fn::text]`	Anonymous inline footnote; the exporter assigns a number.

Link to a local file

KEYS (NO EVIL) C-c C-l · insert/edit link; C-c C-o · open link; C-c C-e h h · export
DOOM ALIAS C-c l l l · insert/edit link; C-c n l · store link

Use a relative path to keep the project portable. The HTML contains the small sample file as an embedded download.

COMPLETE ORG SOURCE

```
Download [[file:assets/readings.csv][the readings]].
```

HTML EXPORT

Download the readings.

Generated by native Org. The frame and typography belong to this guide.

`file:`	Local-file link.
`Relative path`	Resolved relative to the Org file; the ZIP includes the asset.

ORG-05-000 · 12 EXAMPLES

Tables and calculations

Edit and recalculate real Org tables. Source formulas remain visible beside their evaluated buffer.

ORG-05-001 Export a small table

ORG-05-002 Align the source columns

ORG-05-003 Move a data row down

ORG-05-004 Insert a column

ORG-05-005 Calculate a whole column

ORG-05-006 Change the multiplier

ORG-05-007 Format decimal places

ORG-05-008 Calculate one specific field

ORG-05-009 Sum a range

ORG-05-010 Use an Emacs Lisp formula

ORG-05-011 Refer to a named table

ORG-05-012 Add a caption and a reference

Export a small table

KEYS (NO EVIL) TAB · align / next field; C-c C-c · align; C-c C-e h h · export
DOOM ALIAS C-c l e h h · export HTML

A horizontal separator distinguishes the header from the data.

COMPLETE ORG SOURCE

```
| Item | Count |
|------+-------|
| A | 2 |
| B | 5 |
```

HTML EXPORT

Item	Count
A	2
B	5

Generated by native Org. The frame and typography belong to this guide.

`vertical bar`	Separates cells.
`vertical bar---+---vertical bar`	Horizontal rule; exporter treats the preceding row as a header.

Align the source columns

KEYS (NO EVIL) C-c C-c · org-table-align
DOOM ALIAS C-c l b a · align table

Align the table in the Org buffer without changing its data.

COMPLETE ORG SOURCE

```
| Item | Count |
|------+-------|
| A | 2 |
| B | 5 |
```

ACTION

C-c C-c · org-table-align

```
(org-table-align)
```

Cursor at the beginning of: | A

EDITOR BUFFER

```
| Item | Count |
|------+-------|
| A    |     2 |
| B    |     5 |
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`org-table-align`	Pads cells so the pipe characters line up.
`TAB`	Also aligns while moving between table fields.

Move a data row down

KEYS (NO EVIL) M-DOWN · org-table-move-row-down

Move the A row below B.

COMPLETE ORG SOURCE

```
| Item | Count |
|------+-------|
| A | 2 |
| B | 5 |
```

ACTION

M-DOWN · org-table-move-row-down

```
(org-table-align)
(org-table-move-row-down)
```

Cursor at the beginning of: | A

EDITOR BUFFER

```
| Item | Count |
|------+-------|
| B    |     5 |
| A    |     2 |
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`org-table-move-row-down`	Moves the current table row down one position.

Insert a column

KEYS (NO EVIL) M-S-RIGHT · org-table-insert-column
DOOM ALIAS C-c l b i c · insert column

Insert an empty column at the Count field.

COMPLETE ORG SOURCE

```
| Item | Count |
|------+-------|
| A | 2 |
| B | 5 |
```

ACTION

M-S-RIGHT · org-table-insert-column

```
(org-table-align)
(org-table-insert-column)
```

Cursor at the beginning of: Count

EDITOR BUFFER

```
| Item |   | Count |
|------+---+-------|
| A    |   |     2 |
| B    |   |     5 |
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`org-table-insert-column`	Inserts before the current column.
`M-S-RIGHT`	Standard binding inside a table.

Calculate a whole column

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · formula editor
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

Compute the second column from the first.

COMPLETE ORG SOURCE

```
| x | twice |
|---+-------|
| 2 |       |
| 5 |       |
#+TBLFM: $2=2*$1
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | x

EDITOR BUFFER

```
| x | twice |
|---+-------|
| 2 |     4 |
| 5 |    10 |
#+TBLFM: $2=2*$1
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`$2`	Destination column.
`$1`	Value from the first column in the same row.
`2*$1`	Multiply by two.

Change the multiplier

Compare with ORG-05-005 · Calculate a whole column

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · formula editor
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

Change only the multiplier; the second row now produces 15.

COMPLETE ORG SOURCE

```
| x | twice |
|---+-------|
| 2 |       |
| 5 |       |
#+TBLFM: $2=3*$1
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | x

EDITOR BUFFER

```
| x | twice |
|---+-------|
| 2 |     6 |
| 5 |    15 |
#+TBLFM: $2=3*$1
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`3*$1`	Multiply the current row's first field by three.

Format decimal places

Compare with ORG-05-006 · Change the multiplier

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · formula editor
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

Show the quotient with exactly two decimal places.

COMPLETE ORG SOURCE

```
| x | twice |
|---+-------|
| 2 |       |
| 5 |       |
#+TBLFM: $2=$1/3;%.2f
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | x

EDITOR BUFFER

```
| x | twice |
|---+-------|
| 2 |  0.67 |
| 5 |  1.67 |
#+TBLFM: $2=$1/3;%.2f
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`;%.2f` Calc formula output format: floating-point, two decimal places.

Calculate one specific field

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · formula editor
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

Use row and column coordinates to fill the Total field.

COMPLETE ORG SOURCE

```
| Label | Value |
|-------+-------|
| A     | 2     |
| B     | 5     |
| Total |       |
#+TBLFM: @4$2=@2$2+@3$2
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | Label

EDITOR BUFFER

```
| Label | Value |
|-------+-------|
| A     |     2 |
| B     |     5 |
| Total |     7 |
#+TBLFM: @4$2=@2$2+@3$2
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`@4$2`	Fourth non-rule row, second column; the header counts as row 1.
`@2$2+@3$2`	Add the values at the two explicit coordinates.

Sum a range

Compare with ORG-05-008 · Calculate one specific field

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · formula editor
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

Replace individual additions with a vector sum.

COMPLETE ORG SOURCE

```
| Label | Value |
|-------+-------|
| A     | 2     |
| B     | 5     |
| C     | 8     |
| Total |       |
#+TBLFM: @5$2=vsum(@2$2..@4$2)
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | Label

EDITOR BUFFER

```
| Label | Value |
|-------+-------|
| A     |     2 |
| B     |     5 |
| C     |     8 |
| Total |    15 |
#+TBLFM: @5$2=vsum(@2$2..@4$2)
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`vsum(...)`	Sum the cells in a range.
`..`	Inclusive range between two coordinates.

Use an Emacs Lisp formula

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · formula editor
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

Use Lisp instead of Calc for the same numerical transformation.

COMPLETE ORG SOURCE

```
| x | twice |
|---+-------|
| 2 |       |
| 5 |       |
#+TBLFM: $2='(* 2 $1);N
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | x

EDITOR BUFFER

```
| x | twice |
|---+-------|
| 2 |     4 |
| 5 |    10 |
#+TBLFM: $2='(* 2 $1);N
```

Actual text captured from Emacs; syntax colours are applied by the guide.

'	Selects an Emacs Lisp formula.
;N	Convert referenced fields to numbers before evaluating.

Refer to a named table

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · formula editor
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

Read a value from another table in the same document.

COMPLETE ORG SOURCE

```
#+NAME: inputs
| Item | Count |
|------+-------|
| A    | 2     |
| B    | 5     |

| Summary | Value |
|---------+-------|
| First   |       |
#+TBLFM: @2$2=remote(inputs,@2$2)
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | Summary

EDITOR BUFFER

```
#+NAME: inputs
| Item | Count |
|------+-------|
| A    | 2     |
| B    | 5     |

| Summary | Value |
|---------+-------|
| First   |     2 |
#+TBLFM: @2$2=remote(inputs,@2$2)
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`#+NAME: inputs`	Names the source table.
`remote(inputs,@2$2)`	Reads its row 2, column 2.

Add a caption and a reference

KEYS (NO EVIL) C-c C-e h h · export HTML (org-export-dispatch)
DOOM ALIAS C-c l e h h · export HTML

Give the exported table a caption and a named reference.

COMPLETE ORG SOURCE

```
See [[tab:counts]].

#+NAME: tab:counts
#+CAPTION: Observations by group
| Item | Count |
|------+-------|
| A | 2 |
| B | 5 |
```

HTML EXPORT

See 1.

Table 1: Observations by group

Item	Count
A	2
B	5

Generated by native Org. The frame and typography belong to this guide.

`#+CAPTION:`	Caption for the next element.
`#+NAME:`	Named export target; references use the exported label/number.

ORG-06-000 · 40 EXAMPLES

Spreadsheets and formulas

Cell references, aggregates, conditions and dependencies: start with a calculated column, then change one formula or one recalculation command at a time. Every result is computed by Org. Use the keyboard reference for formula editing and debugging.

ORG-06-001 Multiply two columns

ORG-06-002 Target one cell

ORG-06-003 Target a rectangular range

ORG-06-004 Keep a parameter cell fixed

ORG-06-005 Refer to nearby columns

ORG-06-006 Put a total in the last row

ORG-06-007 Use separators as range boundaries

ORG-06-008 Sum across a row

ORG-06-009 Calculate a row average

ORG-06-010 Calculate a weighted mean

ORG-06-011 Show a share as a percentage

ORG-06-012 Build a running total

ORG-06-013 Compare with the previous row

ORG-06-014 Choose between two values

ORG-06-015 Guard against division by zero

ORG-06-016 Ignore empty cells in a mean

ORG-06-017 Treat empty cells as zero

ORG-06-018 Format fixed decimal places

Spreadsheets and formulas · continued

Cell references, aggregates, conditions and dependencies: start with a calculated column, then change one formula or one recalculation command at a time. Every result is computed by Org. Use the keyboard reference for formula editing and debugging.

ORG-06-019 Use scientific notation

ORG-06-020 Keep exact fractions

ORG-06-021 Use degrees in trigonometry

ORG-06-022 Switch to radians

ORG-06-023 Name a file-level constant

ORG-06-024 Give columns meaningful names

ORG-06-025 Keep a parameter beside the data

ORG-06-026 Read a heading property

ORG-06-027 Chain formulas from left to right

ORG-06-028 See a dependency that needs another pass

ORG-06-029 Iterate until values stabilize

ORG-06-030 Recalculate only the current row

ORG-06-031 Aggregate a different table

ORG-06-032 Look up a row by its key

ORG-06-033 Transform text with Emacs Lisp

ORG-06-034 Use numeric Lisp arguments

ORG-06-035 Add durations

ORG-06-036 Convert a duration to decimal hours

Spreadsheets and formulas · continued

Cell references, aggregates, conditions and dependencies: start with a calculated column, then change one formula or one recalculation command at a time. Every result is computed by Org. Use the keyboard reference for formula editing and debugging.

ORG-06-037 Understand multiple formula lines

ORG-06-038 Apply a different formula line

ORG-06-039 Generate row numbers

ORG-06-040 Make arithmetic grouping explicit

Multiply two columns

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

A column formula fills the data rows below the first separator. Edit the inputs, then recalculate: ordinary Org tables do not update continuously.

COMPLETE ORG SOURCE

```
| Item | Qty | Price | Cost |
|------+-----+-------+------|
| Pen  |   2 |     3 |      |
| Pad  |   4 |     5 |      |
#+TBLFM: $4=$2*$3
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | Item | Qty | Price | Cost |

EDITOR BUFFER

```
| Item | Qty | Price | Cost |
|------+-----+-------+------|
| Pen  |   2 |     3 |    6 |
| Pad  |   4 |     5 |   20 |
#+TBLFM: $4=$2*$3
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`$4`	Destination: column 4.
`$2*$3`	Multiply quantity by price in the current row.
`C-c =`	Edit a column formula while the cursor is in that column.

Target one cell

Compare with ORG-06-001 · Multiply two columns

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

Replace the column destination with an explicit row and column. The second item is left untouched. Row numbers count the header but skip separator lines.

COMPLETE ORG SOURCE

```
| Item | Qty | Price | Cost |
|------+-----+-------+------|
| Pen  |   2 |     3 |      |
| Pad  |   4 |     5 |      |
#+TBLFM: @2$4=$2*$3
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | Item | Qty | Price | Cost |

EDITOR BUFFER

```
| Item | Qty | Price | Cost |
|------+-----+-------+------|
| Pen  |   2 |     3 |    6 |
| Pad  |   4 |     5 |      |
#+TBLFM: @2$4=$2*$3
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`@2$4`	Row 2, column 4: the first data row.
`C-u C-c =`	Edit a field formula at the destination cell.

Target a rectangular range

Compare with ORG-06-002 · Target one cell

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

A range on the left applies the formula to each destination. Here the range has one column and two rows; the final row stays blank.

COMPLETE ORG SOURCE

```
| Item | Qty | Price | Cost |
|------+-----+-------+------|
| Pen  |   2 |     3 |      |
| Pad  |   4 |     5 |      |
| Ink  |   1 |     8 |      |
#+TBLFM: @2$4..@3$4=$2*$3
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | Item | Qty | Price | Cost |

EDITOR BUFFER

```
| Item | Qty | Price | Cost |
|------+-----+-------+------|
| Pen  |   2 |     3 |    6 |
| Pad  |   4 |     5 |   20 |
| Ink  |   1 |     8 |      |
#+TBLFM: @2$4..@3$4=$2*$3
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`@2$4..@3$4`	Inclusive destination range.
`C-c }`	Show row and column coordinates over the table.

Keep a parameter cell fixed

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

Both data rows read the same parameter cell. Absolute references keep the row and column explicit when the formula is applied elsewhere.

COMPLETE ORG SOURCE

```
| Item | Value | Scaled |
| Gain |     3 |        |
|------+-------+--------|
| A    |     2 |        |
| B    |     5 |        |
#+TBLFM: $3=$2*@2$2
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | Item | Value | Scaled |

EDITOR BUFFER

```
| Item | Value | Scaled |
| Gain |     3 |        |
|------+-------+--------|
| A    |     2 |      6 |
| B    |     5 |     15 |
#+TBLFM: $3=$2*@2$2
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`@2$2`	Fixed cell containing the gain.
`First separator`	Column formulas skip the two header rows above it.

Refer to nearby columns

Compare with ORG-06-001 · Multiply two columns

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

Relative column references are measured from the destination. This is the same multiplication as example 001, expressed as two steps and one step to the left.

COMPLETE ORG SOURCE

```
| Item | Qty | Price | Cost |
|------+-----+-------+------|
| Pen  |   2 |     3 |      |
| Pad  |   4 |     5 |      |
#+TBLFM: $4=$-2*$-1
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | Item | Qty | Price | Cost |

EDITOR BUFFER

```
| Item | Qty | Price | Cost |
|------+-----+-------+------|
| Pen  |   2 |     3 |    6 |
| Pad  |   4 |     5 |   20 |
#+TBLFM: $4=$-2*$-1
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`$-2`	Two columns left of the current destination.
`$-1`	One column left of the current destination.

Put a total in the last row

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

The destination follows the final row. The sum stops one row before it, avoiding a self-reference when you add more data rows.

COMPLETE ORG SOURCE

```
| Item  | Amount |
|-------+--------|
| Pen   |      6 |
| Pad   |     20 |
|-------+--------|
| Total |        |
#+TBLFM: @>$2=vsum(@2$2..@-1$2)
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | Item | Amount |

EDITOR BUFFER

```
| Item  | Amount |
|-------+--------|
| Pen   |      6 |
| Pad   |     20 |
|-------+--------|
| Total |     26 |
#+TBLFM: @>$2=vsum(@2$2..@-1$2)
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`@>`	Last row; @< would mean the first row.
`@-1`	Row immediately before the destination.
`vsum(...)`	Sum the values in a range.

Use separators as range boundaries

Compare with ORG-06-006 · Put a total in the last row

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

Horizontal rules can define a data region. Only the values between the first and second separator enter the sum.

COMPLETE ORG SOURCE

```
| Item  | Amount |
|-------+--------|
| Pen   |      6 |
| Pad   |     20 |
|-------+--------|
| Total |        |
#+TBLFM: @>$2=vsum(@I$2..@II$2)
```

EDITOR BUFFER

```
| Item  | Amount |
|-------+--------|
| Pen   |      6 |
| Pad   |     20 |
|-------+--------|
| Total |     26 |
#+TBLFM: @>$2=vsum(@I$2..@II$2)
```

Actual text captured from Emacs; syntax colours are applied by the guide.

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | Item | Amount |

`@I / @II`	First / second horizontal separator.
`Range endpoints`	The separator lines themselves contain no values.

Sum across a row

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

Ranges work horizontally as well as vertically. The current row is implicit when only columns are specified.

COMPLETE ORG SOURCE

```
| Team | Jan | Feb | Mar | Total |
|------+-----+-----+-----+-------|
| A    |   2 |   3 |   4 |       |
| B    |   5 |   1 |   6 |       |
#+TBLFM: $5=vsum($2..$4)
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | Team | Jan | Feb | Mar | Total |

EDITOR BUFFER

```
| Team | Jan | Feb | Mar | Total |
|------+-----+-----+-----+-------|
| A    |   2 |   3 |   4 |     9 |
| B    |   5 |   1 |   6 |    12 |
#+TBLFM: $5=vsum($2..$4)
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`$2..$4`	Columns 2 through 4 of the current row.
`$5`	Store each row total in column 5.

Calculate a row average

Compare with ORG-06-008 · Sum across a row

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

Changing only the aggregate turns the total into a mean. The three numeric readings contribute equally.

COMPLETE ORG SOURCE

```
| Team | Jan | Feb | Mar | Mean |
|------+-----+-----+-----+------|
| A    |   2 |   3 |   4 |      |
| B    |   5 |   1 |   6 |      |
#+TBLFM: $5=vmean($2..$4)
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | Team | Jan | Feb | Mar | Mean |

EDITOR BUFFER

```
| Team | Jan | Feb | Mar | Mean |
|------+-----+-----+-----+------|
| A    |   2 |   3 |   4 |    3 |
| B    |   5 |   1 |   6 |    4 |
#+TBLFM: $5=vmean($2..$4)
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`vmean(...)`	Arithmetic mean of a vector or range.

Calculate a weighted mean

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

Multiply each reading by its weight, add the products and divide by the sum of the weights. Parentheses make the intended grouping explicit.

COMPLETE ORG SOURCE

```
| Value | Weight |
|-------+--------|
|    10 |      1 |
|    20 |      3 |
|-------+--------|
| Mean  |        |
#+TBLFM: @>$2=(@2$1*@2$2+@3$1*@3$2)/vsum(@2$2..@3$2)
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | Value | Weight |

EDITOR BUFFER

```
| Value | Weight |
|-------+--------|
|    10 |      1 |
|    20 |      3 |
|-------+--------|
|  Mean |   17.5 |
#+TBLFM: @>$2=(@2$1*@2$2+@3$1*@3$2)/vsum(@2$2..@3$2)
```

Actual text captured from Emacs; syntax colours are applied by the guide.

Numerator	10 x 1 + 20 x 3 = 70.
Denominator	1 + 3 = 4; weighted mean = 17.5.

Show a share as a percentage

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

Each amount is divided by the same total. Multiply the ratio by 100 and give the output column an explicit percent label.

COMPLETE ORG SOURCE

```
| Item | Amount | Share % |
|------+--------+---------|
| A    |     20 |         |
| B    |     30 |         |
| C    |     50 |         |
#+TBLFM: $3=100*($2/vsum(@2$2..@4$2));%.1f
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | Item | Amount | Share % |

EDITOR BUFFER

```
| Item | Amount | Share % |
|------+--------+---------|
| A    |     20 |    20.0 |
| B    |     30 |    30.0 |
| C    |     50 |    50.0 |
#+TBLFM: $3=100*($2/vsum(@2$2..@4$2));%.1f
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`Fixed total range`	The denominator stays the same for all rows.
`;%.1f`	Format each displayed result with one decimal place.

Build a running total

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

The beginning of the range stays fixed while its end follows the current row. This avoids depending on a previously calculated total.

COMPLETE ORG SOURCE

```
| Day | Amount | Running |
|-----+--------+---------|
| Mon |      2 |         |
| Tue |      3 |         |
| Wed |      4 |         |
#+TBLFM: $3=vsum(@2$2..@0$2)
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | Day | Amount | Running |

EDITOR BUFFER

```
| Day | Amount | Running |
|-----+--------+---------|
| Mon |      2 |       2 |
| Tue |      3 |       5 |
| Wed |      4 |       9 |
#+TBLFM: $3=vsum(@2$2..@0$2)
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`@0`	Current destination row.
`@2$2..@0$2`	From the first amount through the current amount.

Compare with the previous row

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

Start the formula on the second data row, because the first has no earlier reading. The relative reference follows each destination.

COMPLETE ORG SOURCE

```
| Day | Reading | Change |
|-----+---------+--------|
| Mon |      10 |        |
| Tue |      14 |        |
| Wed |      11 |        |
#+TBLFM: @3$3..@4$3=$2-@-1$2
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | Day | Reading | Change |

EDITOR BUFFER

```
| Day | Reading | Change |
|-----+---------+--------|
| Mon |      10 |        |
| Tue |      14 |      4 |
| Wed |      11 |     -3 |
#+TBLFM: @3$3..@4$3=$2-@-1$2
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`@-1$2`	Column 2 of the preceding row.
`Destination range`	Exclude the first reading from the calculation.

Choose between two values

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

Use a condition to assign a numeric band. The threshold is inclusive: a score of exactly 50 receives 1.

COMPLETE ORG SOURCE

```
| Name | Score | Passed |
|------+-------+--------|
| Ada  |    70 |        |
| Bea  |    40 |        |
| Cy   |    50 |        |
#+TBLFM: $3=if($2>=50,1,0)
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | Name | Score | Passed |

EDITOR BUFFER

```
| Name | Score | Passed |
|------+-------+--------|
| Ada  |    70 |      1 |
| Bea  |    40 |      0 |
| Cy   |    50 |      1 |
#+TBLFM: $3=if($2>=50,1,0)
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`if(test,yes,no)`	Evaluate the branch selected by the condition.
`>=`	Greater than or equal to.

Guard against division by zero

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

Choose a defined fallback before dividing. Zero here means the chosen fallback, so use a separate status column if it must be distinguished from a measured zero.

COMPLETE ORG SOURCE

```
| Total | Count | Per item |
|-------+-------+----------|
|    12 |     3 |          |
|     8 |     0 |          |
#+TBLFM: $3=if($2==0,0,$1/$2)
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | Total | Count | Per item |

EDITOR BUFFER

```
| Total | Count | Per item |
|-------+-------+----------|
|    12 |     3 |        4 |
|     8 |     0 |        0 |
#+TBLFM: $3=if($2==0,0,$1/$2)
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`==`	Equality test in a Calc formula.
`if($2==0,0,...)`	Skip the division when the count is zero.

Ignore empty cells in a mean

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

By default, range references drop empty fields. The mean is therefore (10 + 20) / 2, not a mean over three measurements.

COMPLETE ORG SOURCE

```
| A | B | C  | Mean |
|---+---+----+------|
|10 |   | 20 |      |
#+TBLFM: $4=vmean($1..$3)
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | A | B | C | Mean |

EDITOR BUFFER

```
|  A | B |  C | Mean |
|----+---+----+------|
| 10 |   | 20 |   15 |
#+TBLFM: $4=vmean($1..$3)
```

Actual text captured from Emacs; syntax colours are applied by the guide.

Empty range fields	Omitted by default; a blank is not automatically a measured zero.

Treat empty cells as zero

Compare with ORG-06-016 · Ignore empty cells in a mean

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

The formula is unchanged; the EN flags change how its input range is interpreted. Now all three positions contribute to the denominator.

COMPLETE ORG SOURCE

```
| A | B | C  | Mean |
|---+---+----+------|
|10 |   | 20 |      |
#+TBLFM: $4=vmean($1..$3);EN
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | A | B | C | Mean |

EDITOR BUFFER

```
|  A | B |  C | Mean |
|----+---+----+------|
| 10 |   | 20 |   10 |
#+TBLFM: $4=vmean($1..$3);EN
```

Actual text captured from Emacs; syntax colours are applied by the guide.

E	Keep empty cells in range references.
N	Treat nonnumeric fields as zero; combined with E, blanks become zero.

Format fixed decimal places

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

A printf-style format controls the displayed result. Org stores that formatted text in the cell, so later formulas can read an already rounded value.

COMPLETE ORG SOURCE

```
| Value | Third |
|-------+-------|
|     1 |       |
|     2 |       |
#+TBLFM: $2=$1/3;%.2f
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | Value | Third |

EDITOR BUFFER

```
| Value | Third |
|-------+-------|
|     1 |  0.33 |
|     2 |  0.67 |
#+TBLFM: $2=$1/3;%.2f
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`;%.2f`	Exactly two digits after the decimal point.
Rounding	Keep additional digits in intermediate cells when precision matters.

Use scientific notation

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

Only the output format changes. Scientific notation gives a compact view of values across very different scales.

COMPLETE ORG SOURCE

```
| Value   | Copy |
|---------+------|
| 1234000 |      |
| 0.00123 |      |
#+TBLFM: $2=$1;%.2e
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | Value | Copy |

EDITOR BUFFER

```
|   Value |     Copy |
|---------+----------|
| 1234000 | 1.23e+06 |
| 0.00123 | 1.23e-03 |
#+TBLFM: $2=$1;%.2e
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`;%.2e`	Scientific notation with two decimal places in the mantissa.

Keep exact fractions

Compare with ORG-06-018 · Format fixed decimal places

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

Fraction mode keeps rational arithmetic exact. The result remains a fraction instead of a rounded decimal approximation.

COMPLETE ORG SOURCE

```
| Value | Third |
|-------+-------|
|     1 |       |
|     2 |       |
#+TBLFM: $2=$1/3;F
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | Value | Third |

EDITOR BUFFER

```
| Value | Third |
|-------+-------|
|     1 |   1:3 |
|     2 |   2:3 |
#+TBLFM: $2=$1/3;F
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`;F`	Calc fraction mode.
`1:3`	Calc notation for one third.

Use degrees in trigonometry

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

Declare the angle unit explicitly. A numeric angle without a unit would otherwise depend on the calculation mode.

COMPLETE ORG SOURCE

```
| Angle | Sine |
|-------+------|
|    30 |      |
|    90 |      |
#+TBLFM: $2=sin($1);D%.3f
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | Angle | Sine |

EDITOR BUFFER

```
| Angle |  Sine |
|-------+-------|
|    30 | 0.500 |
|    90 | 1.000 |
#+TBLFM: $2=sin($1);D%.3f
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`D`	Interpret trigonometric input in degrees.
`%.3f`	Display three decimal places.

Switch to radians

Compare with ORG-06-021 · Use degrees in trigonometry

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

Change the flag and the input values together. The inputs below are numeric radian values; approximately 1.57079632679 radians is a right angle.

COMPLETE ORG SOURCE

```
 | Angle | Sine |
 |-------+------|
| 0.523598775598 |      |
| 1.57079632679  |      |
#+TBLFM: $2=sin($1);R%.3f
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | Angle | Sine |

EDITOR BUFFER

```
|          Angle |  Sine |
|----------------+-------|
| 0.523598775598 | 0.500 |
|  1.57079632679 | 1.000 |
#+TBLFM: $2=sin($1);R%.3f
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`R`	Interpret trigonometric input in radians.
`Numeric input`	Approximately pi/6 and pi/2 radians.

Name a file-level constant

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

A named constant removes a repeated magic number. Recalculate after changing the constant; the stored result cells are not live views.

COMPLETE ORG SOURCE

```
#+CONSTANTS: gain=3
| Value | Scaled |
|-------+--------|
|     2 |        |
|     5 |        |
#+TBLFM: $2=$1*$gain
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | Value | Scaled |

EDITOR BUFFER

```
#+CONSTANTS: gain=3
| Value | Scaled |
|-------+--------|
|     2 |      6 |
|     5 |     15 |
#+TBLFM: $2=$1*$gain
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`#+CONSTANTS:`	File-level values available to table formulas.
`$gain`	Reference the named constant.

Give columns meaningful names

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

A special ! row names columns. Reserve the first column for control markers and mark data rows with # so whole-table recalculation includes them.

COMPLETE ORG SOURCE

```
| ! | qty | price | cost |
|---+-----+-------+------|
| # |   2 |     3 |      |
| # |   4 |     5 |      |
#+TBLFM: $4=$qty*$price
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | ! | qty | price | cost |

EDITOR BUFFER

```
| ! | qty | price | cost |
|---+-----+-------+------|
| # |   2 |     3 |    6 |
| # |   4 |     5 |   20 |
#+TBLFM: $4=$qty*$price
```

Actual text captured from Emacs; syntax colours are applied by the guide.

!	Column names are defined in this special row.
#	A marked data row; it also supports automatic recalculation on TAB/RET.
$4	Destination column; named columns are used on the right-hand side.

Keep a parameter beside the data

Compare with ORG-06-023 · Name a file-level constant

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

A special $ row holds a named parameter inside the table. As with named columns, the first column is reserved for markers.

COMPLETE ORG SOURCE

```
| $ | gain=3 |        |
|---+--------+--------|
| ! | value  | scaled |
|---+--------+--------|
| # |      2 |        |
| # |      5 |        |
#+TBLFM: $3=$value*$gain
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | $ | gain=3 | |

EDITOR BUFFER

```
| $ | gain=3 |        |
|---+--------+--------|
| ! |  value | scaled |
|---+--------+--------|
| # |      2 |      6 |
| # |      5 |     15 |
#+TBLFM: $3=$value*$gain
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`$ marker`	Defines parameters in fields such as gain=3.
`$gain`	Read the table parameter in a formula.

Read a heading property

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

A table can use a property from its enclosing heading. The PROP_ prefix distinguishes property values from other constants.

COMPLETE ORG SOURCE

```
* Budget
:PROPERTIES:
:RATE: 0.2
:END:

| Net | Gross |
|-----+-------|
| 100 |       |
|  50 |       |
#+TBLFM: $2=$1*(1+$PROP_RATE);%.0f
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | Net | Gross |

EDITOR BUFFER

```
* Budget
:PROPERTIES:
:RATE: 0.2
:END:

| Net | Gross |
|-----+-------|
| 100 |   120 |
|  50 |    60 |
#+TBLFM: $2=$1*(1+$PROP_RATE);%.0f
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`$PROP_RATE`	Value of the RATE property, with inheritance supported.
`C-c C-x p`	Set the property interactively at the heading.

Chain formulas from left to right

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

Separate assignments with two colons. Org calculates column formulas from left to right within each row, so the third column can use the freshly calculated second one.

COMPLETE ORG SOURCE

```
| Input | Double | Plus one |
|-------+--------+----------|
|     2 |      0 |        0 |
|     5 |      0 |        0 |
#+TBLFM: $2=2*$1::$3=$2+1
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | Input | Double | Plus one |

EDITOR BUFFER

```
| Input | Double | Plus one |
|-------+--------+----------|
|     2 |      4 |        5 |
|     5 |     10 |       11 |
#+TBLFM: $2=2*$1::$3=$2+1
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`::`	Separates independent assignments.
`Calculation order`	Column 2 is updated before column 3.

See a dependency that needs another pass

Compare with ORG-06-027 · Chain formulas from left to right

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

Move the dependency to the right and run one pass. Column 2 reads the old zero in column 3; the visibly stale result is intentional.

COMPLETE ORG SOURCE

```
| Input | Plus one | Double |
|-------+----------+--------|
|     2 |        0 |      0 |
|     5 |        0 |      0 |
#+TBLFM: $2=$3+1::$3=2*$1
```

EDITOR BUFFER

```
| Input | Plus one | Double |
|-------+----------+--------|
|     2 |        1 |      4 |
|     5 |        1 |     10 |
#+TBLFM: $2=$3+1::$3=2*$1
```

Actual text captured from Emacs; syntax colours are applied by the guide.

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | Input | Plus one | Double |

`$2=$3+1`	Reads column 3 before its update in this pass.
`Result 1`	Demonstrates the dependency, not the final desired answer.

Iterate until values stabilize

Compare with ORG-06-028 · See a dependency that needs another pass

KEYS (NO EVIL) C-u C-u C-c * · org-table-iterate
DOOM ALIAS C-u C-u C-c I * · iterate table

Keep the same source and change only the command. Iteration repeats recalculation until the table stops changing or reaches its iteration limit; circular formulas need not converge.

COMPLETE ORG SOURCE

```
| Input | Plus one | Double |
|-------+----------+--------|
|     2 |        0 |      0 |
|     5 |        0 |      0 |
#+TBLFM: $2=$3+1::$3=2*$1
```

ACTION

C-u C-u C-c * · org-table-iterate

`(org-table-iterate)`

Cursor at the beginning of: | Input | Plus one | Double |

EDITOR BUFFER

```
| Input | Plus one | Double |
|-------+----------+--------|
|     2 |        5 |      4 |
|     5 |       11 |     10 |
#+TBLFM: $2=$3+1::$3=2*$1
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`C-u C-u C-c *`	Iterate the table rather than run one pass.
`org-table-iterate`	Command for repeatedly recalculating the current table.

Recalculate only the current row

Compare with ORG-06-001 · Multiply two columns

KEYS (NO EVIL) C-c * · org-table-recalculate (current row)
DOOM ALIAS C-c l b r · recalculate current row

Omit the prefix argument to recalculate just the row at point. The second data row remains blank, even though the formula applies to its column.

COMPLETE ORG SOURCE

```
| Item | Qty | Price | Cost |
|------+-----+-------+------|
| Pen  |   2 |     3 |      |
| Pad  |   4 |     5 |      |
#+TBLFM: $4=$2*$3
```

ACTION

C-c * · org-table-recalculate (current row)

`(org-table-recalculate)`

Cursor at the beginning of: | Pen

EDITOR BUFFER

```
| Item | Qty | Price | Cost |
|------+-----+-------+------|
| Pen  |   2 |     3 |    6 |
| Pad  |   4 |     5 |      |
#+TBLFM: $4=$2*$3
```

Actual text captured from Emacs; syntax colours are applied by the guide.

C-c *	Recalculate the current row.
C-u C-c *	Recalculate all applicable rows in the table.

Aggregate a different table

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

Give the input table a name and pass an absolute range to remote. The Summary table reads those cells without copying the data.

COMPLETE ORG SOURCE

```
#+NAME: sales
| Item | Amount |
|------+--------|
| A    |     20 |
| B    |     30 |

| Summary | Value |
|---------+-------|
| Total   |       |
#+TBLFM: @2$2=vsum(remote(sales,@2$2..@3$2))
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | Summary

EDITOR BUFFER

```
#+NAME: sales
| Item | Amount |
|------+--------|
| A    |     20 |
| B    |     30 |

| Summary | Value |
|---------+-------|
| Total   |    50 |
#+TBLFM: @2$2=vsum(remote(sales,@2$2..@3$2))
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`remote(sales,...)`	Read cells from the named table.
Dependencies	Recalculate the source table first if it also contains formulas.

Look up a row by its key

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

An Emacs Lisp formula can match a key in one table and return a value from the corresponding row. Reordering the input rows will not change these matches.

COMPLETE ORG SOURCE

```
#+NAME: prices
| Code | Price |
|------+-------|
| pen  |     3 |
| pad  |     5 |

| Code | Price |
|------+-------|
| pad  |       |
| pen  |       |
#+TBLFM: $2='(org-lookup-first $1 '(remote(prices,@2$1..@3$1))
'(remote(prices,@2$2..@3$2)));E
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | pad | |

EDITOR BUFFER

```
#+NAME: prices
| Code | Price |
|------+-------|
| pen  |     3 |
| pad  |     5 |

| Code | Price |
|------+-------|
| pad  |     5 |
| pen  |     3 |
#+TBLFM: $2='(org-lookup-first $1 '(remote(prices,@2$1..@3$1))
'(remote(prices,@2$2..@3$2)));E
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`org-lookup-first`	Return the first corresponding result for an equal key.
`;E`	Preserve empty fields so the two lookup ranges stay aligned.
`Leading apostrophe`	Select Emacs Lisp rather than Calc.

Transform text with Emacs Lisp

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

Calc is not the only formula engine. In an Emacs Lisp formula, ordinary field references are substituted as strings unless numeric mode is requested.

COMPLETE ORG SOURCE

```
| Name | Upper |
|------+-------|
| ada  |       |
| bea  |       |
#+TBLFM: $2='(upcase $1)
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | Name | Upper |

EDITOR BUFFER

```
| Name | Upper |
|------+-------|
| ada  | ADA   |
| bea  | BEA   |
#+TBLFM: $2='(upcase $1)
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`'(upcase $1)`	Call the Lisp upcase function on the field text.
`No N flag`	Keep the text as a string.

Use numeric Lisp arguments

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

The N flag supplies numeric values to a Lisp formula. Without it, these field references would be strings and + would reject them.

COMPLETE ORG SOURCE

```
| A | B | Sum |
|---+---+-----|
| 2 | 3 |     |
| 5 | 8 |     |
#+TBLFM: $3='(+ $1 $2);N
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | A | B | Sum |

EDITOR BUFFER

```
| A | B | Sum |
|---+---+-----|
| 2 | 3 |   5 |
| 5 | 8 |  13 |
#+TBLFM: $3='(+ $1 $2);N
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`;N`	Read numeric fields as numbers for Lisp arithmetic.
`+`	The Emacs Lisp addition function.

Add durations

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

The duration flag lets Calc read hours and minutes and format the sum as a duration. These cells are elapsed times, not clock times or dates.

COMPLETE ORG SOURCE

```
| Work | Break | Total |
|------+-------+-------|
| 1:30 |  0:45 |       |
| 2:10 |  0:20 |       |
#+TBLFM: $3=$1+$2;U
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | Work | Break | Total |

EDITOR BUFFER

```
| Work | Break | Total |
|------+-------+-------|
| 1:30 |  0:45 | 02:15 |
| 2:10 |  0:20 | 02:30 |
#+TBLFM: $3=$1+$2;U
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`;U`	Interpret durations and display hours:minutes.
`;T`	Use hours:minutes:seconds instead.

Convert a duration to decimal hours

Compare with ORG-06-035 · Add durations

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

Keep the same duration arithmetic and switch the output flag. Lowercase t uses the configurable duration format, whose default is fractional hours.

COMPLETE ORG SOURCE

```
| Work | Break | Hours |
 |------+-------+-------|
 | 1:30 |  0:45 |       |
 | 2:10 |  0:20 |       |
#+TBLFM: $3=$1+$2;t
```

EDITOR BUFFER

```
| Work | Break | Hours |
|------+-------+-------|
| 1:30 |  0:45 |  2.25 |
| 2:10 |  0:20 |  2.50 |
#+TBLFM: $3=$1+$2;t
```

Actual text captured from Emacs; syntax colours are applied by the guide.

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | Work | Break | Hours |

`;t`	Format using org-table-duration-custom-format.
`Default format`	Fractional hours with two decimal places in this tested environment.

Understand multiple formula lines

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

Keep two alternative formula lines under the table. A normal table recalculation uses the first line; it does not apply both alternatives in sequence.

COMPLETE ORG SOURCE

```
| Value | Result |
|-------+--------|
|     2 |        |
|     5 |        |
#+TBLFM: $2=2*$1
#+TBLFM: $2=3*$1
```

EDITOR BUFFER

```
| Value | Result |
|-------+--------|
|     2 |      4 |
|     5 |     10 |
#+TBLFM: $2=2*$1
#+TBLFM: $2=3*$1
```

Actual text captured from Emacs; syntax colours are applied by the guide.

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | Value | Result |

`First #+TBLFM line`	Default formula set used by table recalculation.
`:: on one line`	Use this when both assignments belong to the same formula set.

Apply a different formula line

Compare with ORG-06-037 · Understand multiple formula lines

KEYS (NO EVIL) C-c C-c on the second #+TBLFM line · org-ctrl-c-ctrl-c
DOOM ALIAS C-c l b f · inspect formula assignments

Move point to the second formula line and use C-c C-c. Org applies that selected line for this calculation; the first line remains the default for later table recalculation.

COMPLETE ORG SOURCE

```
| Value | Result |
|-------+--------|
|     2 |        |
|     5 |        |
#+TBLFM: $2=2*$1
#+TBLFM: $2=3*$1
```

ACTION

C-c C-c on the second #+TBLFM line · org-ctrl-c-ctrl-c

`(org-ctrl-c-ctrl-c)`

Cursor at the beginning of: `#+TBLFM: $2=3*$1`

EDITOR BUFFER

```
| Value | Result |
|-------+--------|
|     2 |      6 |
|     5 |     15 |
#+TBLFM: $2=2*$1
#+TBLFM: $2=3*$1
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`C-c C-c on #+TBLFM`	Apply the formula set on that particular line.
`Cursor position`	Distinguishes this action from C-c C-c inside a table cell.

Generate row numbers

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

The @# token is the numeric row index of the destination. Subtract one because the header occupies the first row.

COMPLETE ORG SOURCE

```
| No | Item |
|----+------|
|    | Pen  |
|    | Pad  |
|    | Ink  |
#+TBLFM: $1=@#-1
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | No | Item |

EDITOR BUFFER

```
| No | Item |
|----+------|
|  1 | Pen  |
|  2 | Pad  |
|  3 | Ink  |
#+TBLFM: $1=@#-1
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`@#`	Numeric row index of the current destination.
`$#`	The corresponding token for the column index.

Make arithmetic grouping explicit

KEYS (NO EVIL) C-u C-c * · recalculate table; C-c ' · edit formulas
DOOM ALIAS C-u C-c l b r · recalculate whole table; C-c l b f · formulas

Calc gives multiplication special precedence after division. Parenthesize the intended quotient before multiplying; the three result columns make the difference visible.

COMPLETE ORG SOURCE

```
| a | b | c | a/b*c | (a/b)*c | a/(b*c) |
|---+---+---+-------+---------+---------|
|12 | 2 | 3 |       |         |         |
#+TBLFM: $4=$1/$2*$3::$5=($1/$2)*$3::$6=$1/($2*$3)
```

ACTION

C-u C-c * · org-table-recalculate (whole table)

```
(org-table-recalculate t)
```

Cursor at the beginning of: | a | b | c | a/b*c | (a/b)*c | a/(b*c) |

EDITOR BUFFER

```
|  a | b | c | a/b*c | (a/b)*c | a/(b*c) |
|----+---+---+-------+---------+---------|
| 12 | 2 | 3 |     2 |      18 |       2 |
#+TBLFM: $4=$1/$2*$3::$5=($1/$2)*$3::$6=$1/($2*$3)
```

Actual text captured from Emacs; syntax colours are applied by the guide.

Calc parsing	a/b*c is parsed as a/(b*c).
(a/b)*c	Use parentheses when the intended result is 18.

ORG-07-000 · 12 EXAMPLES

Tasks and planning

Separate task state, checklist progress, scheduled work and deadlines. Agenda examples use a fixed date.

ORG-07-001 Mark a heading as a task

ORG-07-002 Complete a task

ORG-07-003 Define a workflow

ORG-07-004 Set a priority

ORG-07-005 Start a checklist

ORG-07-006 Check one item

ORG-07-007 Count completed items

ORG-07-008 Show percentage progress

ORG-07-009 See one day in the agenda

ORG-07-010 Move the scheduled task

ORG-07-011 Show all unfinished tasks

ORG-07-012 Add a repeating date

Mark a heading as a task

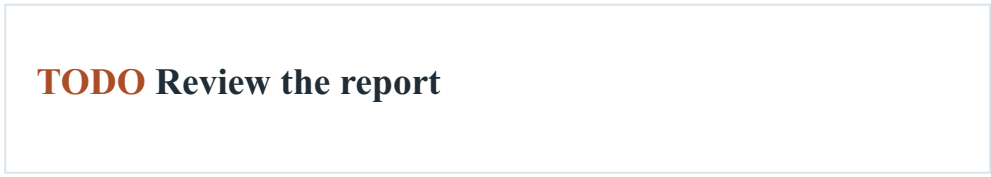
KEYS (NO EVIL) C-c C-t · org-todo; C-c C-e h h · export
DOOM ALIAS C-c l t · change TODO state

TODO is a recognized keyword at the start of a heading.

COMPLETE ORG SOURCE

```
* TODO Review the report
```

HTML EXPORT



Generated by native Org. The frame and typography belong to this guide.

TODO	Default unfinished task state.

Complete a task

Compare with ORG-07-001 · Mark a heading as a task

KEYS (NO EVIL) C-c C-t · choose DONE
DOOM ALIAS C-c l t · change TODO state

Change the task state at the heading. This build disables completion timestamps to keep the example focused.

COMPLETE ORG SOURCE

```
* TODO Review the report
```

ACTION

C-c C-t · choose DONE

```
(org-todo "DONE")
```

Cursor at the beginning of: * TODO

EDITOR BUFFER

```
* DONE Review the report
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`org-todo`	Changes the current task state.
`DONE`	Default completed state.

Define a workflow

KEYS (NO EVIL) C-c C-t · choose NEXT
DOOM ALIAS C-c l t · change TODO state

Define additional states locally, then select NEXT.

COMPLETE ORG SOURCE

```
#+TODO: TODO NEXT WAIT | DONE
* TODO Review the report
```

ACTION

C-c C-t · choose NEXT

```
(org-todo "NEXT")
```

Cursor at the beginning of: * TODO

EDITOR BUFFER

```
#+TODO: TODO NEXT WAIT | DONE
* NEXT Review the report
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`#+TODO:`	File-local keyword sequence.
`TODO NEXT WAIT`	Unfinished states.
`vertical bar`	Separates unfinished from completed states.

Set a priority

KEYS (NO EVIL) C-c , · org-priority; S-UP / S-DOWN · adjust priority
DOOM ALIAS C-c l p p · set priority

Add a priority cookie to the heading.

COMPLETE ORG SOURCE

```
* TODO [#A] Review the report
```

HTML EXPORT

TODO Review the report

Generated by native Org. The frame and typography belong to this guide.

`[#A]`	Highest default priority; default priorities run from A to C.

Start a checklist

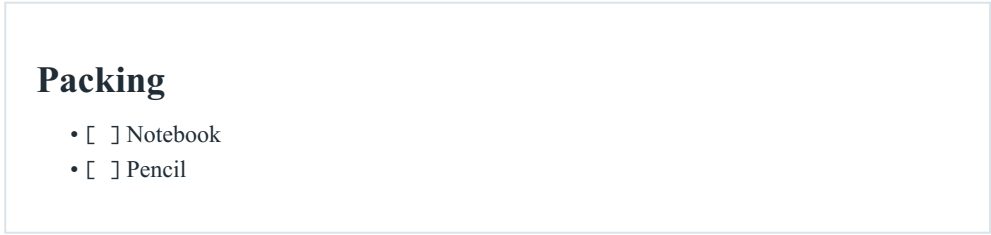
KEYS (NO EVIL) M-S-RET · insert checkbox item; C-c C-c · toggle checkbox
DOOM ALIAS C-c l x · toggle checkbox

Checkboxes belong to list items and are independent of heading TODO states.

COMPLETE ORG SOURCE

```
* Packing
- [ ] Notebook
- [ ] Pencil
```

HTML EXPORT



Generated by native Org. The frame and typography belong to this guide.

`[ ]`	Unchecked checkbox; [X] is checked; [-] is partial.

Check one item

Compare with ORG-07-005 · Start a checklist

KEYS (NO EVIL) C-c C-c · toggle checkbox
DOOM ALIAS C-c l x · toggle checkbox

Toggle the checkbox at Notebook.

COMPLETE ORG SOURCE

```
* Packing
- [ ] Notebook
- [ ] Pencil
```

ACTION

C-c C-c · toggle checkbox

`(org-toggle-checkbox)`

Cursor at the beginning of: - [ ] Notebook

EDITOR BUFFER

```
* Packing
- [X] Notebook
- [ ] Pencil
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`org-toggle-checkbox`	Changes checkbox state at point.

Count completed items

Compare with ORG-07-006 · Check one item

KEYS (NO EVIL) C-c C-c on the cookie
DOOM ALIAS C-c l # · update statistics cookies

Refresh the fraction in the heading from the checklist below it.

COMPLETE ORG SOURCE

```
* Packing [0/2]
- [X] Notebook
- [ ] Pencil
```

ACTION

C-c C-c on the cookie

```
(org-update-statistics-cookies t)
```

Cursor at the beginning of: * Packing

EDITOR BUFFER

```
* Packing [1/2]
- [X] Notebook
- [ ] Pencil
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`[0/2]`	Statistics cookie showing completed / total items.
`org-update-statistics-cookies`	Recomputes task and checkbox statistics.

Show percentage progress

Compare with ORG-07-007 · Count completed items

KEYS (NO EVIL) C-c C-c on the cookie
DOOM ALIAS C-c l # · update statistics cookies

Use a percentage cookie for the same checklist.

COMPLETE ORG SOURCE

```
* Packing [0%]
 - [X] Notebook
 - [ ] Pencil
```

ACTION

C-c C-c on the cookie

```
(org-update-statistics-cookies t)
```

Cursor at the beginning of: * Packing

EDITOR BUFFER

```
* Packing [50%]
- [X] Notebook
- [ ] Pencil
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`[0%]`	Percentage form of the statistics cookie.

See one day in the agenda

KEYS (NO EVIL) M-x org-agenda RET a · agenda; d · day view; j · jump to date
DOOM ALIAS C-c n a a · open agenda; then d and j

Collect scheduled work, deadline warnings and an appointment into one day view. The reference day is 12 October 2026.

COMPLETE ORG SOURCE

```
* TODO Analyse readings
SCHEDULED: <2026-10-12 Mon>
* TODO Send report
DEADLINE: <2026-10-14 Wed>
* Meeting
<2026-10-12 Mon 14:00>
```

ACTION

M-x org-agenda-list · one day

```
(org-agenda-list nil "2026-10-12" 1)
```

AGENDA BUFFER

```
Day-agenda (W42):
Monday     12 October 2026 W42
 14:00...... Meeting
 Scheduled:  TODO Analyse readings
 In   2 d.:  TODO Send report
```

Actual text captured from Emacs; syntax colours are applied by the guide.

SCHEDULED:	Date on which work should start.
DEADLINE:	Due date; warnings may appear before it.
<... 14:00>	Active timestamp with appointment time.

Move the scheduled task

Compare with ORG-07-009 · See one day in the agenda

KEYS (NO EVIL) C-c C-s · reschedule task; M-x org-agenda RET a · agenda
DOOM ALIAS C-c l d s · schedule; C-c n a a · open agenda

Move the scheduled date one day forward. It no longer appears as scheduled work on Monday.

COMPLETE ORG SOURCE

```
* TODO Analyse readings
SCHEDULED: <2026-10-13 Tue>
* TODO Send report
DEADLINE: <2026-10-14 Wed>
* Meeting
<2026-10-12 Mon 14:00>
```

ACTION
Agenda for 12 October 2026

```
(org-agenda-list nil "2026-10-12" 1)
```

AGENDA BUFFER

```
Day-agenda (W42):
Monday     12 October 2026 W42
 14:00...... Meeting
 In   2 d.:  TODO Send report
```

Actual text captured from Emacs; syntax colours are applied by the guide.

2026-10-13 Tue	The new start date; the displayed agenda day is unchanged.

Show all unfinished tasks

KEYS (NO EVIL) M-x org-agenda RET t · global TODO list (org-todo-list)
DOOM ALIAS C-c n t · global TODO list

A global TODO list collects unfinished headings even if they have no appointment today.

COMPLETE ORG SOURCE

```
* TODO Analyse readings
SCHEDULED: <2026-10-12 Mon>
* TODO Send report
DEADLINE: <2026-10-14 Wed>
* Meeting
<2026-10-12 Mon 14:00>
```

ACTION

M-x org-todo-list

```
(org-todo-list)
```

AGENDA BUFFER

```
Global list of TODO items of type: ALL
Press ‘N r’ (e.g. ‘0 r’) to search again: (0)[ALL] (1)DONE (2)TODO
 TODO Analyse readings
 TODO Send report
```

Actual text captured from Emacs; syntax colours are applied by the guide.

org-todo-list	Builds the global TODO view from org-agenda-files.

Add a repeating date

KEYS (NO EVIL) C-c . · insert timestamp, including +1w; C-c C-s · schedule
DOOM ALIAS C-c l d t · timestamp; C-c l d s · schedule

A repeater records how to advance the date when the task is completed. The preview shows the source because repeater behavior is an editor operation.

COMPLETE ORG SOURCE

```
* TODO Check backups
SCHEDULED: <2026-10-12 Mon +1w>
```

ACTION
Source inspection · no command

EDITOR BUFFER

```
* TODO Check backups
SCHEDULED: <2026-10-12 Mon +1w>
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`+1w`	Advance from the previous scheduled date by one week on completion.
`++1w / .+1w`	Respectively catch up into the future / repeat relative to completion.

ORG-08-000 · 8 EXAMPLES

Tags and properties

Attach machine-readable context to headings. Inspect metadata and simple filtered views.

ORG-08-001 Tag a heading

ORG-08-002 Add a property drawer

ORG-08-003 Read a property

ORG-08-004 Inherit a parent property

ORG-08-005 Override an inherited value

ORG-08-006 Set a property with a command

ORG-08-007 Find headings with a tag

ORG-08-008 Filter by a property value

Tag a heading

KEYS (NO EVIL) C-c C-q · org-set-tags-command
DOOM ALIAS C-c l q · set tags

Tags classify headings without changing their titles.

COMPLETE ORG SOURCE

```
* TODO Review report :writing:projectA:
```

EDITOR BUFFER

```
* TODO Review report :writing:projectA:
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`:writing:projectA:`	Colon-separated tags at the end of the heading.

Add a property drawer

KEYS (NO EVIL) C-c C-x p · org-set-property
DOOM ALIAS C-c l o · set property

A drawer stores named values below a heading.

COMPLETE ORG SOURCE

```
* Study
:PROPERTIES:
:OWNER: Alex
:STATUS: draft
:END:
Notes.
```

EDITOR BUFFER

```
* Study
:PROPERTIES:
:OWNER: Alex
:STATUS: draft
:END:
Notes.
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`:PROPERTIES: / :END:`	Drawer delimiters.
`:OWNER: Alex`	Property name and value.

Read a property

KEYS (NO EVIL) M-: · eval-expression; enter the displayed Lisp form, then RET

Read OWNER programmatically at the current heading.

COMPLETE ORG SOURCE

```
* Study
:PROPERTIES:
:OWNER: Alex
:END:
Notes.
```

ACTION

(org-entry-get nil "OWNER")

```
(org-entry-get nil "OWNER")
```

Cursor at the beginning of: * Study

RETURNED VALUE

```
"Alex"
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`org-entry-get`	Gets a named property from the entry at point.
`nil`	Use the current point as the location.

Inherit a parent property

Compare with ORG-08-003 · Read a property

KEYS (NO EVIL) M-: · eval-expression; enter the displayed Lisp form, then RET

Read OWNER at the child heading, explicitly enabling inheritance for this lookup.

COMPLETE ORG SOURCE

```
* Study
:PROPERTIES:
:OWNER: Alex
:END:
** Analysis
Notes.
```

ACTION

(org-entry-get nil "OWNER" t)

```
(org-entry-get nil "OWNER" t)
```

Cursor at the beginning of: ** Analysis

RETURNED VALUE

```
"Alex"
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`Third argument t`	Search ancestors if the current entry has no value.

Override an inherited value

Compare with ORG-08-004 · Inherit a parent property

KEYS (NO EVIL) M-: · eval-expression; enter the displayed Lisp form, then RET

A property on the child takes precedence over the parent value.

COMPLETE ORG SOURCE

```
* Study
:PROPERTIES:
:OWNER: Alex
:END:
** Analysis
:PROPERTIES:
:OWNER: Sam
:END:
Notes.
```

ACTION

(org-entry-get nil "OWNER" t)

```
(org-entry-get nil "OWNER" t)
```

Cursor at the beginning of: ** Analysis

RETURNED VALUE

```
"Sam"
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`Local OWNER`	Overrides the value found in ancestors.

Set a property with a command

KEYS (NO EVIL) C-c C-x p · org-set-property; enter OWNER, then Alex
DOOM ALIAS C-c l o · set property

Create the drawer and property through Org instead of typing the syntax.

COMPLETE ORG SOURCE

```
* Study
Notes.
```

ACTION

C-c C-x p · OWNER · Alex

```
(org-set-property "OWNER" "Alex")
```

Cursor at the beginning of: * Study

EDITOR BUFFER

```
* Study
:PROPERTIES:
:OWNER:    Alex
:END:
Notes.
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`org-set-property`	Writes a property on the current entry.

Find headings with a tag

KEYS (NO EVIL) M-: · evaluate shown form; C-c / m · interactive sparse-tree match
DOOM ALIAS C-c l s s m · interactive sparse-tree match

Map a function over headings matching the review tag. The result is a list of heading titles.

COMPLETE ORG SOURCE

```
* TODO Check data :review:
* TODO Write text :writing:
* TODO Check text :review:
```

ACTION

org-map-entries · tag match

```
(org-map-entries (lambda () (org-get-heading t t t t)) "review" 'file)
```

RETURNED VALUE

```
("Check data" "Check text")
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`org-map-entries`	Calls a function on each matching entry.
`"review"`	Tag matcher.
`'file`	Restrict the search to this file.

Filter by a property value

KEYS (NO EVIL) M-: · evaluate shown form; C-c / m · interactive sparse-tree match
DOOM ALIAS C-c l s s m · interactive sparse-tree match

Use a property matcher to select Study A.

COMPLETE ORG SOURCE

```
* Study A
:PROPERTIES:
:OWNER: Alex
:END:
* Study B
:PROPERTIES:
:OWNER: Sam
:END:
```

ACTION

org-map-entries · property match

```
(org-map-entries (lambda () (org-get-heading t t t t)) "OWNER=\"Alex\"" 'file)
```

RETURNED VALUE

```
("Study A")
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`OWNER="Alex"`	Matches the exact property value Alex.

ORG-09-000 · 6 EXAMPLES

Images and blocks

Keep content portable; make format-specific presentation choices explicit.

ORG-09-001 Insert an image

ORG-09-002 Constrain the HTML image width

ORG-09-003 Add an image caption

ORG-09-004 Quote a paragraph

ORG-09-005 Preserve a literal example

ORG-09-006 Create a named special block

Insert an image

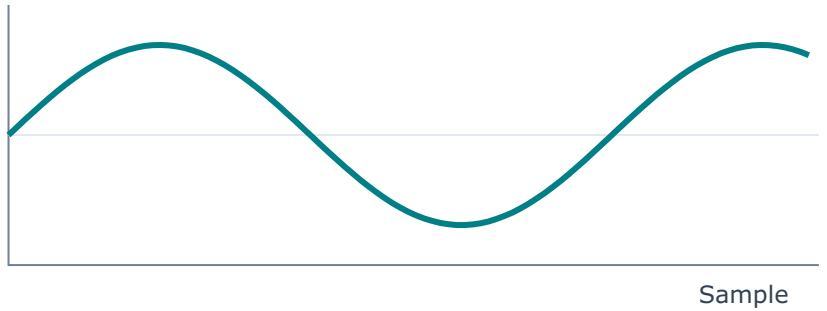
KEYS (NO EVIL) C-c C-l · insert image link; C-c C-x C-v · toggle inline images; C-c C-e h h · export
DOOM ALIAS C-c l e h h · export HTML

An image-only link is exported as an inline image. This asset is a small, reproducibly generated SVG.

COMPLETE ORG SOURCE

```
[[file:assets/curve.svg]]
```

HTML EXPORT



Generated by native Org. The frame and typography belong to this guide.

`Image file link`	Org recognizes supported image extensions.

Constrain the HTML image width

Compare with ORG-09-001 · Insert an image

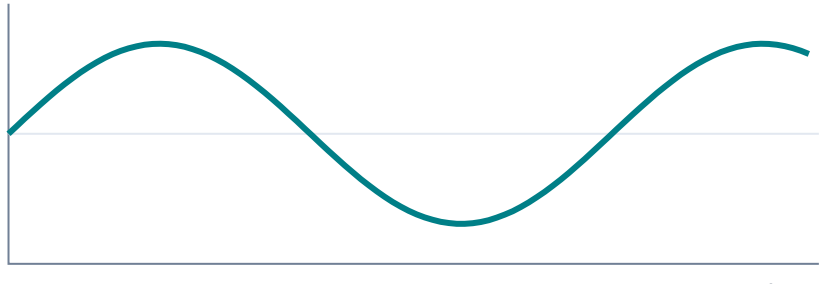
KEYS (NO EVIL) C-c C-l · insert image link; C-c C-x C-v · toggle inline images; C-c C-e h h · export
DOOM ALIAS C-c l e h h · export HTML

Set the image width in the HTML backend.

COMPLETE ORG SOURCE

```
#+ATTR_HTML: :width 240
[[file:assets/curve.svg]]
```

HTML EXPORT



Generated by native Org. The frame and typography belong to this guide.

`#+ATTR_HTML:`	Attributes for the next exported element.
`:width 240`	Requested width in CSS pixels; this does not set LaTeX width.

Add an image caption

Compare with ORG-09-002 · Constrain the HTML image width

KEYS (NO EVIL) C-c C-l · insert image link; C-c C-x C-v · toggle inline images; C-c C-e h h · export
DOOM ALIAS C-c l e h h · export HTML

Attach a caption to the same image.

COMPLETE ORG SOURCE

```
#+CAPTION: A tiny signal
#+ATTR_HTML: :width 240
[[file:assets/curve.svg]]
```

HTML EXPORT

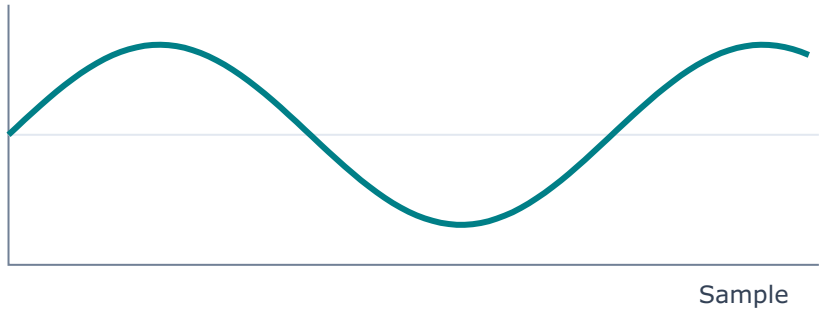


Figure 1: A tiny signal

Generated by native Org. The frame and typography belong to this guide.

`#+CAPTION:`	Text exported with the figure.

Quote a paragraph

KEYS (NO EVIL) C-c C-e h h · export HTML (org-export-dispatch)
DOOM ALIAS C-c l e h h · export HTML

A quote block gives the paragraph quotation semantics.

COMPLETE ORG SOURCE

```
#+BEGIN_QUOTE
A clear record makes a result easier to check.
#+END_QUOTE
```

HTML EXPORT

> A clear record makes a result easier to check.

Generated by native Org. The frame and typography belong to this guide.

quote	Exports as blockquote in HTML and a quotation environment in LaTeX.

Preserve a literal example

KEYS (NO EVIL) C-c C-e h h · export HTML (org-export-dispatch)
DOOM ALIAS C-c l e h h · export HTML

An example block displays Org-looking content without interpreting it.

COMPLETE ORG SOURCE

```
#+BEGIN_EXAMPLE
* This is literal text
#+TITLE: Also literal
#+END_EXAMPLE
```

HTML EXPORT

#+BEGIN_EXAMPLE

This is literal text

#+END_EXAMPLE

Generated by native Org. The frame and typography belong to this guide.

`example block`	Preserves literal lines in a fixed-width block.

Create a named special block

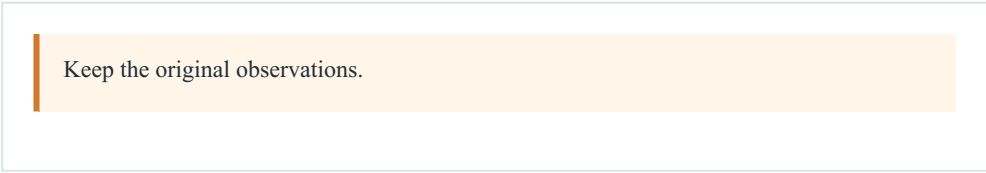
KEYS (NO EVIL) C-c C-e h h · export HTML (org-export-dispatch)
DOOM ALIAS C-c l e h h · export HTML

HTML exports an arbitrary special block as a named container. Styling that class is a separate step; LaTeX would need a matching environment.

COMPLETE ORG SOURCE

```
#+begin_note
Keep the original observations.
#+end_note
```

HTML EXPORT



Generated by native Org. The frame and typography belong to this guide.

`note`	Becomes a div with class note in HTML.
`Backend support`	An arbitrary block name is not automatically a predefined LaTeX environment.

ORG-10-000 · 14 EXAMPLES

Export and layout

Vary one export setting at a time. HTML fragments and native LaTeX/PDF previews are clearly labelled.

ORG-10-001 Hide numbers and the contents list

ORG-10-002 Number the headings

ORG-10-003 Add a shallow contents list

ORG-10-004 Include child headings in the contents

ORG-10-005 Number only the main headings

ORG-10-006 Move the contents list

ORG-10-007 Exclude a section

ORG-10-008 Select sections explicitly

ORG-10-009 Hide TODO keywords in export

ORG-10-010 Remove tasks from export

ORG-10-011 Require braces for subscripts

ORG-10-012 Change the PDF margins

ORG-10-013 Use larger PDF type

ORG-10-014 Insert an explicit PDF page break

Hide numbers and the contents list

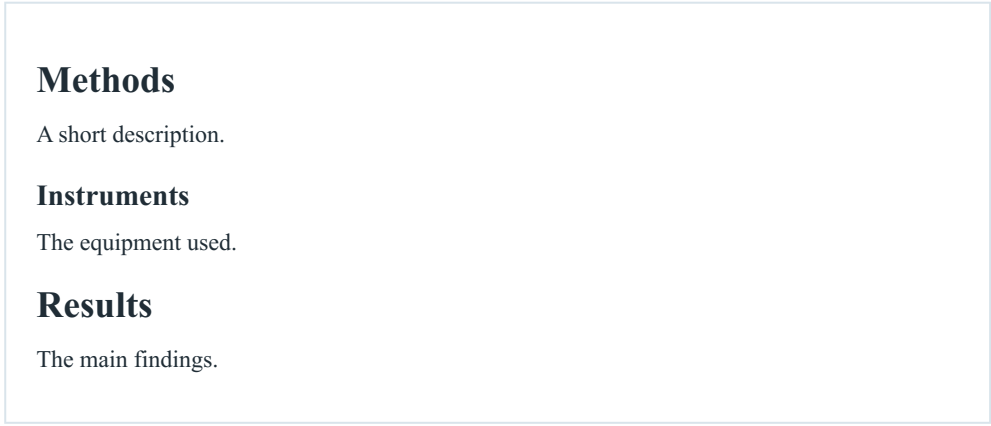
KEYS (NO EVIL) C-c C-e h h · export HTML (org-export-dispatch)
DOOM ALIAS C-c l e h h · export HTML

Start with the simplest outline presentation.

COMPLETE ORG SOURCE

```
#+OPTIONS: num:nil toc:nil
* Methods
A short description.
** Instruments
The equipment used.
* Results
The main findings.
```

HTML EXPORT



Methods

A short description.

Instruments

The equipment used.

Results

The main findings.

Generated by native Org. The frame and typography belong to this guide.

`#+OPTIONS:`	num:nil disables heading numbers; toc:nil disables the table of contents.

Number the headings

Compare with ORG-10-001 · Hide numbers and the contents list

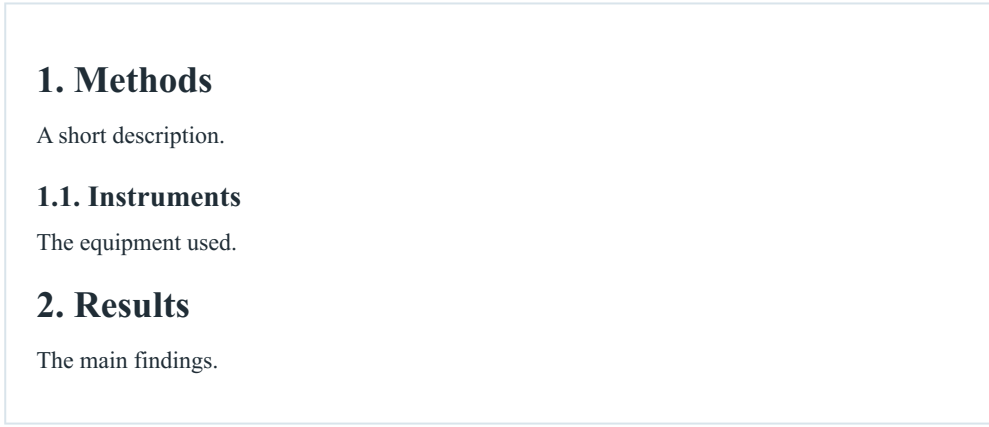
KEYS (NO EVIL) C-c C-e h h · export HTML (org-export-dispatch)
DOOM ALIAS C-c l e h h · export HTML

Turn on section numbers without adding a contents list.

COMPLETE ORG SOURCE

```
#+OPTIONS: num:t toc:nil
* Methods
A short description.
** Instruments
The equipment used.
* Results
The main findings.
```

HTML EXPORT



1. Methods

A short description.

1.1. Instruments

The equipment used.

2. Results

The main findings.

Generated by native Org. The frame and typography belong to this guide.

`#+OPTIONS:`	num:t enables heading numbering.

Add a shallow contents list

Compare with ORG-10-002 · Number the headings

KEYS (NO EVIL) C-c C-e h h · export HTML (org-export-dispatch)
DOOM ALIAS C-c l e h h · export HTML

List only level-one headings in the contents.

COMPLETE ORG SOURCE

```
#+OPTIONS: num:t toc:1
* Methods
A short description.
** Instruments
The equipment used.
* Results
The main findings.
```

HTML EXPORT

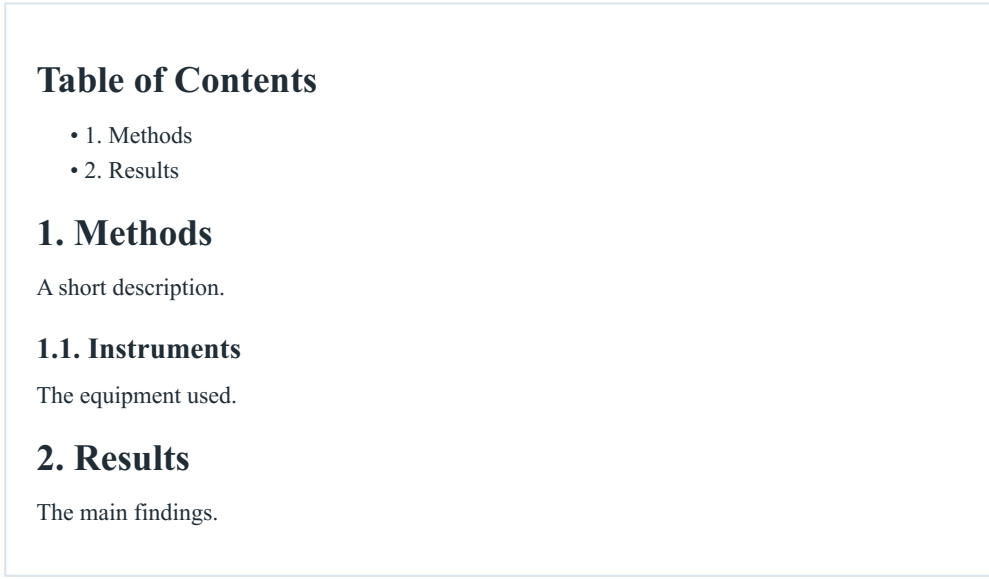


Table of Contents

- 1. Methods
- 2. Results

1. Methods

A short description.

1.1. Instruments

The equipment used.

2. Results

The main findings.

Generated by native Org. The frame and typography belong to this guide.

`#+OPTIONS:`	toc:1 limits the contents list to one level.

Include child headings in the contents

Compare with ORG-10-003 · Add a shallow contents list

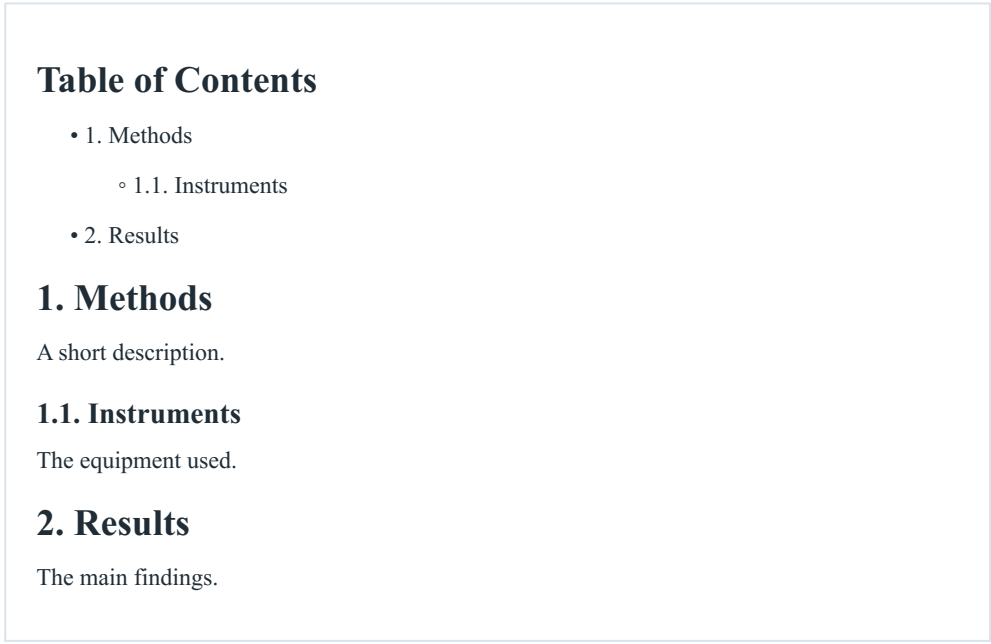
KEYS (NO EVIL) C-c C-e h h · export HTML (org-export-dispatch)
DOOM ALIAS C-c l e h h · export HTML

Include Instruments in the contents list.

COMPLETE ORG SOURCE

```
#+OPTIONS: num:t toc:2
* Methods
A short description.
** Instruments
The equipment used.
* Results
The main findings.
```

HTML EXPORT



Generated by native Org. The frame and typography belong to this guide.

`#+OPTIONS:` toc:2 includes two outline levels.

Number only the main headings

Compare with ORG-10-004 · Include child headings in the contents

KEYS (NO EVIL) C-c C-e h h · export HTML (org-export-dispatch)
DOOM ALIAS C-c l e h h · export HTML

Keep the deep contents list but number only the first heading level.

COMPLETE ORG SOURCE

```
#+OPTIONS: num:1 toc:2
* Methods
A short description.
** Instruments
The equipment used.
* Results
The main findings.
```

HTML EXPORT

Table of Contents

- 1. Methods
  - Instruments
- 2. Results

1. Methods

A short description.

Instruments

The equipment used.

2. Results

The main findings.

Generated by native Org. The frame and typography belong to this guide.

`#+OPTIONS:`	num:1 limits numbering to level one.

Move the contents list

Compare with ORG-10-005 · Number only the main headings

KEYS (NO EVIL) C-c C-e h h · export HTML (org-export-dispatch)
DOOM ALIAS C-c l e h h · export HTML

Disable the default contents location and insert one after the introduction.

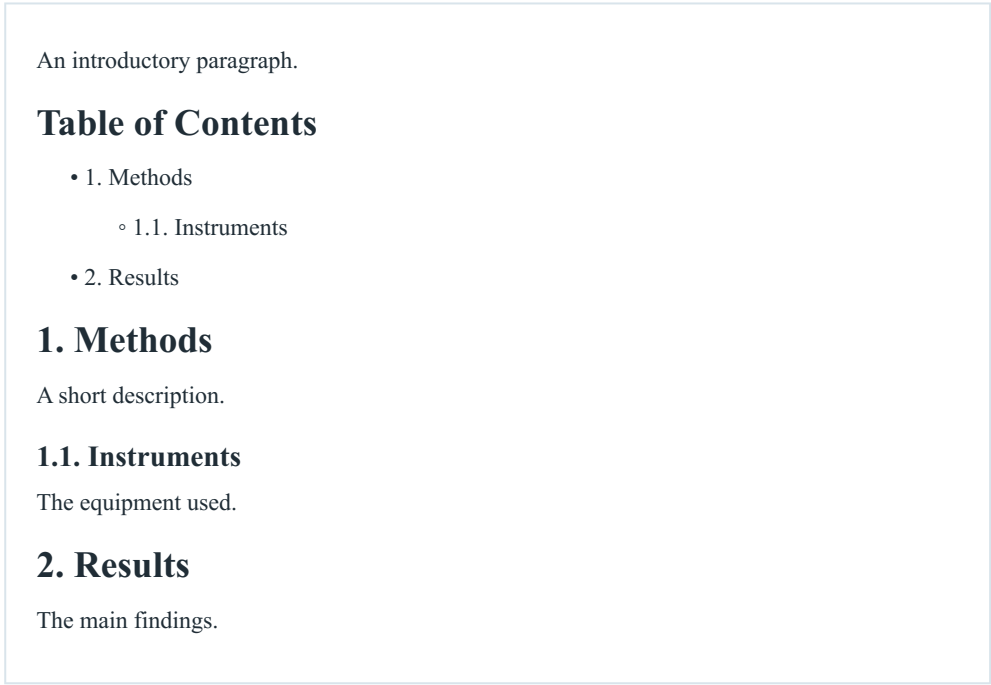
COMPLETE ORG SOURCE

```
#+OPTIONS: num:t toc:nil
An introductory paragraph.

#+TOC: headlines 2

* Methods
A short description.
** Instruments
The equipment used.
* Results
The main findings.
```

HTML EXPORT



An introductory paragraph.

Table of Contents

- 1. Methods
  - 1.1. Instruments
- 2. Results

1. Methods

A short description.

1.1. Instruments

The equipment used.

2. Results

The main findings.

Generated by native Org. The frame and typography belong to this guide.

`#+TOC: headlines 2` Insert a two-level heading contents list at this position.

Exclude a section

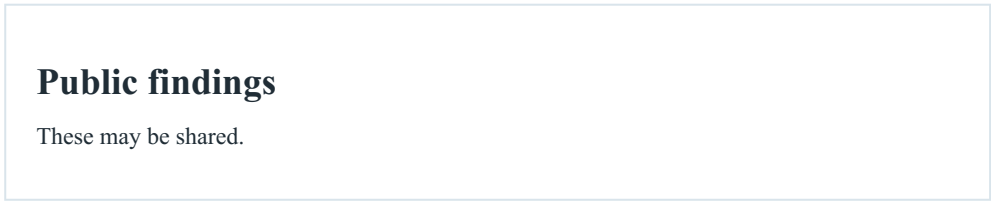
KEYS (NO EVIL) C-c C-e h h · export HTML (org-export-dispatch)
DOOM ALIAS C-c l e h h · export HTML

The noexport tag excludes the entire subtree from the exported document.

COMPLETE ORG SOURCE

```
* Public findings
These may be shared.
* Internal notes :noexport:
Keep these out of the export.
```

HTML EXPORT



Generated by native Org. The frame and typography belong to this guide.

`:noexport:`	Default exclusion tag; it does not remove the source text.

Select sections explicitly

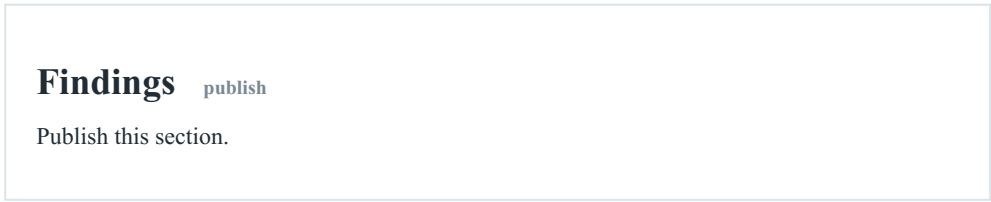
KEYS (NO EVIL) C-c C-e h h · export HTML (org-export-dispatch)
DOOM ALIAS C-c l e h h · export HTML

Export only tagged sections and their subtrees.

COMPLETE ORG SOURCE

```
#+SELECT_TAGS: publish
* Findings :publish:
Publish this section.
* Working notes
Do not publish this section.
```

HTML EXPORT



Generated by native Org. The frame and typography belong to this guide.

`#+SELECT_TAGS: publish` Use publish as the selection tag.

Hide TODO keywords in export

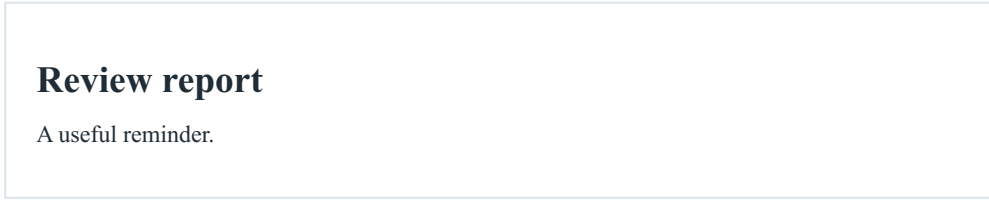
KEYS (NO EVIL) C-c C-e h h · export HTML (org-export-dispatch)
DOOM ALIAS C-c l e h h · export HTML

Keep the task heading but remove its TODO keyword from the exported text.

COMPLETE ORG SOURCE

```
#+OPTIONS: todo:nil
* TODO Review report
A useful reminder.
```

HTML EXPORT



Generated by native Org. The frame and typography belong to this guide.

`todo:nil`	Hides the keyword; tasks:nil would exclude task entries altogether.

Remove tasks from export

Compare with ORG-10-009 · Hide TODO keywords in export

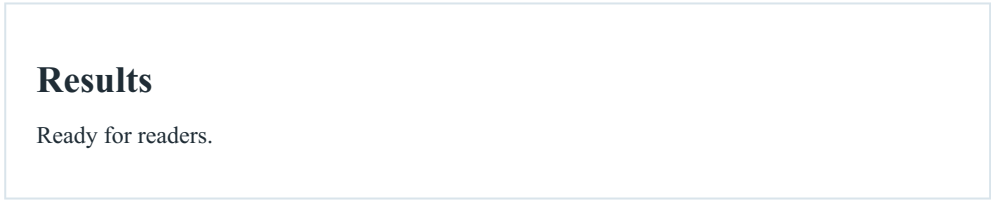
KEYS (NO EVIL) C-c C-e h h · export HTML (org-export-dispatch)
DOOM ALIAS C-c l e h h · export HTML

Exclude task entries, preserving ordinary headings.

COMPLETE ORG SOURCE

```
#+OPTIONS: tasks:nil
 * TODO Review report
 A useful reminder.
* Results
Ready for readers.
```

HTML EXPORT



Generated by native Org. The frame and typography belong to this guide.

`tasks:nil`	Removes task headings and their contents from the export.

Require braces for subscripts

KEYS (NO EVIL) C-c C-e h h · export HTML (org-export-dispatch)
DOOM ALIAS C-c l e h h · export HTML

Require braces for subscript and superscript markup so common identifiers remain intact.

COMPLETE ORG SOURCE

```
#+OPTIONS: ^:{}
Keep sample_A literal; write CO_{2} with a subscript.
```

HTML EXPORT

Keep sample_A literal; write CO_2 with a subscript.

Generated by native Org. The frame and typography belong to this guide.

`^:{}`	Interpret sub/superscripts only when the argument is braced.

Change the PDF margins

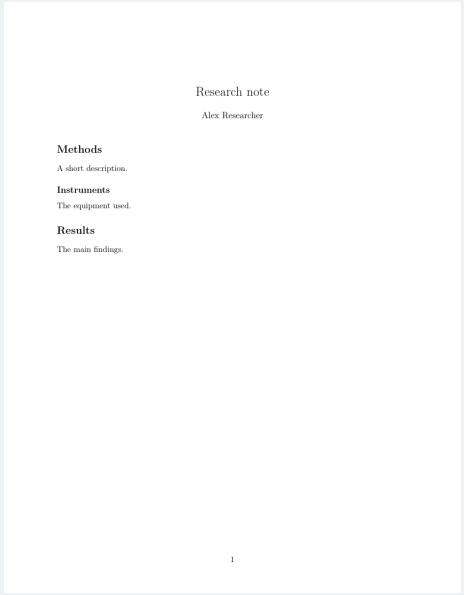
KEYS (NO EVIL) C-c C-e l p · export PDF (org-export-dispatch)
DOOM ALIAS C-c l e l p · export PDF

Export through the native Org LaTeX backend and compile with pdfLaTeX. This preview is a real PDF page.

COMPLETE ORG SOURCE

```
#+LATEX_CLASS: article
#+LATEX_HEADER: \usepackage[margin=25mm]{geometry}
#+TITLE: Research note
#+AUTHOR: Alex Researcher
#+DATE:
* Methods
A short description.
** Instruments
The equipment used.
* Results
The main findings.
```

NATIVE PDF EXPORT



Page 1

Actual native Org → LaTeX → pdfLaTeX output; preview scaled to fit.

`#+LATEX_CLASS: article`	Choose the article document class.
`geometry / margin=25mm`	Set all four page margins to 25 mm.

Use larger PDF type

Compare with ORG-10-012 · Change the PDF margins

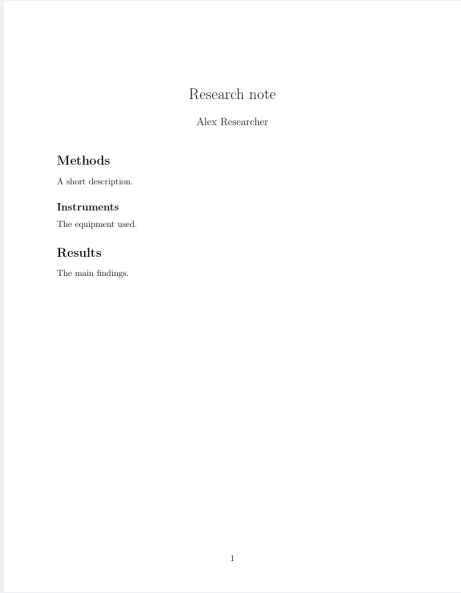
KEYS (NO EVIL) C-c C-e l p · export PDF (org-export-dispatch)
DOOM ALIAS C-c l e l p · export PDF

Change the base font size while keeping margins and content unchanged.

COMPLETE ORG SOURCE

```
#+LATEX_CLASS: article
#+LATEX_CLASS_OPTIONS: [12pt]
#+LATEX_HEADER: \usepackage[margin=25mm]{geometry}
#+TITLE: Research note
#+AUTHOR: Alex Researcher
#+DATE:
* Methods
A short description.
** Instruments
The equipment used.
* Results
The main findings.
```

NATIVE PDF EXPORT


Research note

Alex Researcher

Methods

A short description.

Instruments

The equipment used.

Results

The main findings.

1

Page 1

Actual native Org → LaTeX → pdfLaTeX output; preview scaled to fit.

`#+LATEX_CLASS_OPTIONS: [12pt]` Pass the 12pt option to the LaTeX document class.

Insert an explicit PDF page break

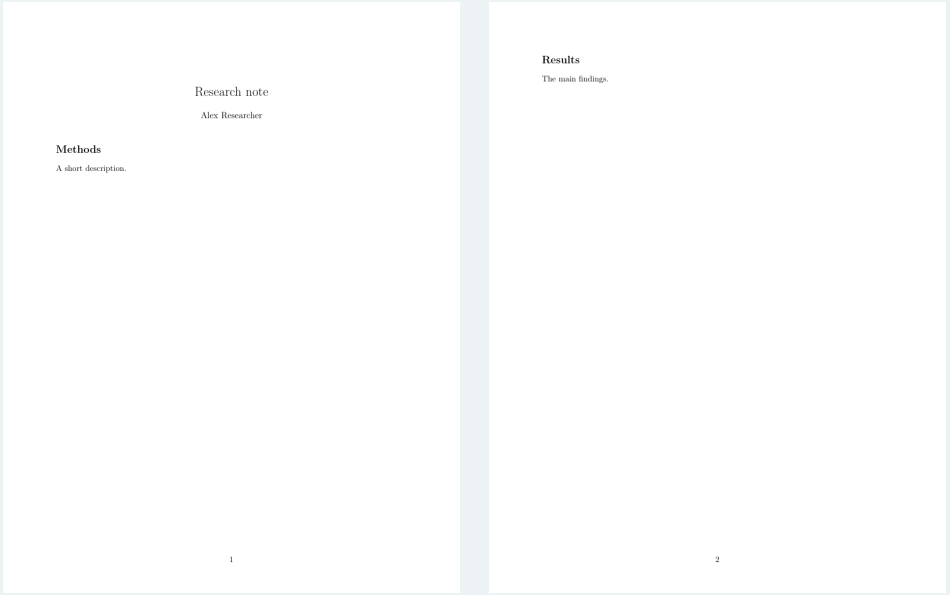
KEYS (NO EVIL) C-c C-e l p · export PDF (org-export-dispatch)
DOOM ALIAS C-c l e l p · export PDF

Place Results on a new PDF page. The guide previews every generated page.

COMPLETE ORG SOURCE

```
#+LATEX_CLASS: article
#+LATEX_HEADER: \usepackage[margin=25mm]{geometry}
#+TITLE: Research note
#+AUTHOR: Alex Researcher
#+DATE:
* Methods
A short description.
#+LATEX: \newpage
* Results
The main findings.
```

NATIVE PDF EXPORT



Page 1 Page 2

Actual native Org → LaTeX → pdfLaTeX output; preview scaled to fit.

`#+LATEX:`	Emit a backend-specific LaTeX line.
`\newpage`	End the current page.

ORG-11-000 · 16 EXAMPLES

Source blocks and Babel

Distinguish execution, stored results and export. Header arguments control each of these separately.

ORG-11-001 Display a source block

ORG-11-002 Run a Bash block

ORG-11-003 Return multiple output lines

ORG-11-004 Return an Emacs Lisp value

ORG-11-005 Return a table

ORG-11-006 Supply an input variable

ORG-11-007 Change only the input

ORG-11-008 Read a named table as input

ORG-11-009 Run Python with a value result

ORG-11-010 Export only the code

ORG-11-011 Export only the result

ORG-11-012 Export code and result

ORG-11-013 Append a new result

ORG-11-014 Set defaults for a whole file

ORG-11-015 Create a plot from a named table

ORG-11-016 Change the plot style

Display a source block

KEYS (NO EVIL) C-c C-, s · insert source block; C-c ' · edit block; C-c C-e h h · export
DOOM ALIAS C-c l e h h · export HTML

Export code without running it.

COMPLETE ORG SOURCE

```
#+BEGIN_SRC bash :eval never
printf "hello\n"
#+END_SRC
```

HTML EXPORT

```
printf "hello\n"
```

Generated by native Org. The frame and typography belong to this guide.

`:eval never`	Prevents Babel evaluation of this block.
`bash`	Language identifier for execution and highlighting.

Run a Bash block

KEYS (NO EVIL) C-c C-c on a source block · execute; C-c C-v b · execute buffer
DOOM ALIAS C-c l ' · edit source block

Execute the block and insert captured standard output below it.

COMPLETE ORG SOURCE

```
#+BEGIN_SRC bash :results output
printf "hello\n"
#+END_SRC
```

EVALUATED ORG BUFFER

```
#+BEGIN_SRC bash :results output
printf "hello\n"
#+END_SRC

#+RESULTS:
: hello
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`:results output`	Capture stdout rather than a language-specific return value.
`#+RESULTS:`	Marker inserted by Babel above the stored result.

Return multiple output lines

Compare with ORG-11-002 · Run a Bash block

KEYS (NO EVIL) C-c C-c on a source block · execute; C-c C-v b · execute buffer
DOOM ALIAS C-c l ' · edit source block

The output result preserves both lines.

COMPLETE ORG SOURCE

```
#+BEGIN_SRC bash :results output
printf "alpha\nbeta\n"
#+END_SRC
```

EVALUATED ORG BUFFER

```
#+BEGIN_SRC bash :results output
printf "alpha\nbeta\n"
#+END_SRC

#+RESULTS:
: alpha
: beta
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`\n`	A newline escape interpreted by printf.

Return an Emacs Lisp value

KEYS (NO EVIL) C-c C-c on a source block · execute; C-c C-v b · execute buffer
DOOM ALIAS C-c l ' · edit source block

Capture the value of the final Lisp expression.

COMPLETE ORG SOURCE

```
#+BEGIN_SRC emacs-lisp :results value
(+ 2 3)
#+END_SRC
```

EVALUATED ORG BUFFER

```
#+BEGIN_SRC emacs-lisp :results value
(+ 2 3)
#+END_SRC

#+RESULTS:
: 5
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`:results value`	Use the language backend's return value.
`(+ 2 3)`	Lisp addition; produces 5.

Return a table

KEYS (NO EVIL) C-c C-c on a source block · execute; C-c C-v b · execute buffer
DOOM ALIAS C-c l ' · edit source block

A list of rows can become an Org table.

COMPLETE ORG SOURCE

```
#+BEGIN_SRC emacs-lisp :results value table
'(("A" 2) ("B" 5))
#+END_SRC
```

EVALUATED ORG BUFFER

```
#+BEGIN_SRC emacs-lisp :results value table
'(("A" 2) ("B" 5))
#+END_SRC

#+RESULTS:
| A | 2 |
| B | 5 |
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`:results value table`	Format the returned rows as a table.
`'(...)`	A literal Lisp list; inner lists are rows.

Supply an input variable

KEYS (NO EVIL) C-c C-c on a source block · execute; C-c C-v b · execute buffer
DOOM ALIAS C-c l ' · edit source block

Pass a named input through a header argument.

COMPLETE ORG SOURCE

```
#+BEGIN_SRC emacs-lisp :var n=4 :results value
(* n n)
#+END_SRC
```

EVALUATED ORG BUFFER

```
#+BEGIN_SRC emacs-lisp :var n=4 :results value
(* n n)
#+END_SRC

#+RESULTS:
: 16
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`:var n=4`	Bind n to the number 4.
`(* n n)`	Square the input.

Change only the input

Compare with ORG-11-006 · Supply an input variable

KEYS (NO EVIL) C-c C-c on a source block · execute; C-c C-v b · execute buffer
DOOM ALIAS C-c l ' · edit source block

Keep the code unchanged and change the parameter to seven.

COMPLETE ORG SOURCE

```
#+BEGIN_SRC emacs-lisp :var n=7 :results value
(* n n)
#+END_SRC
```

EVALUATED ORG BUFFER

```
#+BEGIN_SRC emacs-lisp :var n=7 :results value
(* n n)
#+END_SRC

#+RESULTS:
: 49
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`:var n=7`

The changed input produces 49.

Read a named table as input

KEYS (NO EVIL) C-c C-c on a source block · execute; C-c C-v b · execute buffer
DOOM ALIAS C-c l ' · edit source block

Use the table as a list of rows and sum the first field of each row.

COMPLETE ORG SOURCE

```
#+NAME: measurements
| 2 |
| 5 |
| 8 |

#+BEGIN_SRC emacs-lisp :var data=measurements :results value
(apply #'+ (mapcar #'car data))
#+END_SRC
```

EVALUATED ORG BUFFER

```
#+NAME: measurements
| 2 |
| 5 |
| 8 |

#+BEGIN_SRC emacs-lisp :var data=measurements :results value
(apply #'+ (mapcar #'car data))
#+END_SRC

#+RESULTS:
: 15
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`:var data=measurements`	Resolve the named table and bind its data.
`mapcar / car`	Extract the first item from each row.
`apply`	Call addition with all extracted values.

Run Python with a value result

KEYS (NO EVIL) C-c C-c on a source block · execute; C-c C-v b · execute buffer
DOOM ALIAS C-c l ' · edit source block

For a non-session Python value block, use an explicit return statement.

COMPLETE ORG SOURCE

```
#+BEGIN_SRC python :results value
return sum([2, 5, 8])
#+END_SRC
```

EVALUATED ORG BUFFER

```
#+BEGIN_SRC python :results value
return sum([2, 5, 8])
#+END_SRC

#+RESULTS:
: 15
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`python`	Requires a Python interpreter and ob-python.
`return`	Returns the value from Babel's generated function wrapper.

Export only the code

KEYS (NO EVIL) C-c C-v b · execute buffer; C-c C-e h h · export HTML
DOOM ALIAS C-c l e h h · export HTML

Execute first, then export only the source. The result remains stored in the working Org buffer.

COMPLETE ORG SOURCE

```
#+BEGIN_SRC emacs-lisp :results value :exports code
(+ 2 3)
#+END_SRC
```

HTML AFTER EVALUATION

```
(+ 2 3)
```

Generated by native Org. The frame and typography belong to this guide.

`:exports code`	Include source, omit the result in export.

Export only the result

Compare with ORG-11-010 · Export only the code

KEYS (NO EVIL) C-c C-v b · execute buffer; C-c C-e h h · export HTML
DOOM ALIAS C-c l e h h · export HTML

Use the same computation but hide the source in the rendered document.

COMPLETE ORG SOURCE

```
#+BEGIN_SRC emacs-lisp :results value :exports results
(+ 2 3)
#+END_SRC
```

HTML AFTER EVALUATION

```
5
```

Generated by native Org. The frame and typography belong to this guide.

`:exports results`	Include the stored result, omit source.

Export code and result

Compare with ORG-11-011 · Export only the result

KEYS (NO EVIL) C-c C-v b · execute buffer; C-c C-e h h · export HTML
DOOM ALIAS C-c l e h h · export HTML

Show both sides of the computation.

COMPLETE ORG SOURCE

```
#+BEGIN_SRC emacs-lisp :results value :exports both
(+ 2 3)
#+END_SRC
```

HTML AFTER EVALUATION

```
(+ 2 3)
```

```
5
```

Generated by native Org. The frame and typography belong to this guide.

`:exports both`	Include source and result.

Append a new result

KEYS (NO EVIL) C-c C-c on a source block · execute; C-c C-v b · execute buffer
DOOM ALIAS C-c l ' · edit source block

Keep the existing result and append the new one. Re-running this example starts from the same source each time.

COMPLETE ORG SOURCE

```
#+BEGIN_SRC emacs-lisp :results value append
(+ 2 3)
#+END_SRC

#+RESULTS:
: 2
```

EVALUATED ORG BUFFER

```
#+BEGIN_SRC emacs-lisp :results value append
(+ 2 3)
#+END_SRC

#+RESULTS:
: 2
: 5
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`:results append`	Append to the current result instead of replacing it.

Set defaults for a whole file

KEYS (NO EVIL) C-c C-v b · execute buffer; C-c C-e h h · export HTML
DOOM ALIAS C-c l e h h · export HTML

Move shared header arguments to a language-specific property.

COMPLETE ORG SOURCE

```
#+PROPERTY: header-args:emacs-lisp :results value :exports both

#+BEGIN_SRC emacs-lisp
(+ 2 3)
#+END_SRC
```

HTML AFTER EVALUATION

```
(+ 2 3)
```

```
5
```

Generated by native Org. The frame and typography belong to this guide.

`#+PROPERTY: header-args:emacs-lisp`	Default arguments for Emacs Lisp blocks in this file.
`Local block arguments`	More specific block arguments override these defaults.

Create a plot from a named table

KEYS (NO EVIL) C-c C-v b · execute buffer; C-c C-e h h · export HTML
DOOM ALIAS C-c l e h h · export HTML

Pass the table to gnuplot and insert a link to the generated SVG.

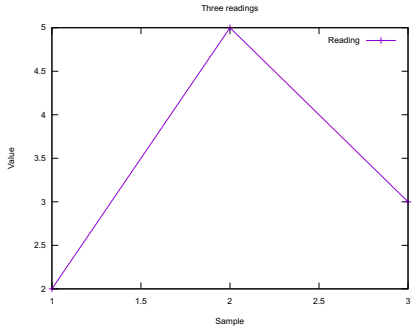
COMPLETE ORG SOURCE

```
#+NAME: readings
| 1 | 2 |
| 2 | 5 |
| 3 | 3 |

#+BEGIN_SRC gnuplot :var data=readings :file plot.svg :results file
graphics :exports results
set title "Three readings"
set xlabel "Sample"
set ylabel "Value"
plot data using 1:2 with linespoints title "Reading"
#+END_SRC
```

HTML AFTER EVALUATION

1	2
2	5
3	3



Generated by native Org. The frame and typography belong to this guide.

`:var data=readings`	Pass the named table to the gnuplot backend.
`:file plot.svg`	Output file, relative to this isolated example directory.
`:results file graphics`	Insert a file result recognized as graphics.

Change the plot style

Compare with ORG-11-015 · Create a plot from a named table

KEYS (NO EVIL) C-c C-v b · execute buffer; C-c C-e h h · export HTML
DOOM ALIAS C-c l e h h · export HTML

Change only linespoints to boxes. The Org data flow and output settings stay the same.

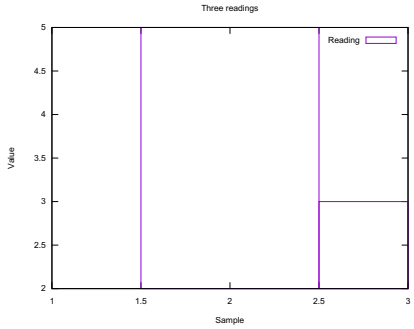
COMPLETE ORG SOURCE

```
#+NAME: readings
| 1 | 2 |
| 2 | 5 |
| 3 | 3 |

#+BEGIN_SRC gnuplot :var data=readings :file plot.svg :results file
graphics :exports results
set title "Three readings"
set xlabel "Sample"
set ylabel "Value"
plot data using 1:2 with boxes title "Reading"
#+END_SRC
```

HTML AFTER EVALUATION

1	2
2	5
3	3



Generated by native Org. The frame and typography belong to this guide.

`with boxes` — Draw a bar at each x value instead of connecting points.

ORG-12-000 · 8 EXAMPLES

Text macros

Reusable text, arguments, metadata and computed replacements. Macro expansion is part of document export; a Babel call is a separate execution mechanism.

ORG-12-001 Expand a macro without arguments

ORG-12-002 Pass one macro argument

ORG-12-003 Pass two macro arguments

ORG-12-004 Keep a comma inside an argument

ORG-12-005 Return markup from a macro

ORG-12-006 Compute a macro replacement

ORG-12-007 Reuse document metadata

ORG-12-008 Load shared macro definitions

Expand a macro without arguments

Compare with ORG-13-001 · Define a text macro

KEYS (NO EVIL) C-c C-e h h · export HTML (macros expand during export)
DOOM ALIAS C-c l e h h · export HTML

Define reusable wording once, then place it in prose. The source keeps the macro call; the exported sentence contains the replacement.

COMPLETE ORG SOURCE

```
#+MACRO: project Open Lab
The {{{project}}} report is ready.
```

HTML EXPORT

The Open Lab report is ready.

Generated by native Org. The frame and typography belong to this guide.

`#+MACRO:`	Define the name and replacement text.
`{{{project}}}`	Call a text macro without parameters.

Pass one macro argument

Compare with ORG-12-001 · Expand a macro without arguments

KEYS (NO EVIL) C-c C-e h h · export HTML (macros expand during export)
DOOM ALIAS C-c l e h h · export HTML

A placeholder inserts the supplied text. This example changes the reusable label without changing the surrounding sentence.

COMPLETE ORG SOURCE

```
#+MACRO: status Status: $1
{{{status(ready)}}}.
```

HTML EXPORT

Status: ready.

Generated by native Org. The frame and typography belong to this guide.

`$1`	First argument, passed as text.
`{{{status(ready)}}}`	One argument in parentheses.

Pass two macro arguments

Compare with ORG-12-002 · Pass one macro argument

KEYS (NO EVIL) C-c C-e h h · export HTML (macros expand during export)
DOOM ALIAS C-c l e h h · export HTML

Use separate placeholders for a value and its unit. The same definition produces both inline measurements.

COMPLETE ORG SOURCE

```
#+MACRO: quantity $1 $2
Length: {{{quantity(12,cm)}}}.
Time: {{{quantity(5,s)}}}.
```

HTML EXPORT

Length: 12 cm. Time: 5 s.

Generated by native Org. The frame and typography belong to this guide.

`$1 / $2`	First / second argument.
`,`	Separates macro arguments.

Keep a comma inside an argument

Compare with ORG-12-003 · Pass two macro arguments

KEYS (NO EVIL) C-c C-e h h · export HTML (macros expand during export)
DOOM ALIAS C-c l e h h · export HTML

Escape a comma that belongs to the text. Without that escape, Org would interpret it as an argument separator.

COMPLETE ORG SOURCE

```
#+MACRO: location $1 — $2
{{{location(Aachen\, Germany,home)}}}
```

HTML EXPORT

Aachen, Germany — home

Generated by native Org. The frame and typography belong to this guide.

`\,`	A literal comma inside one argument.
Expansion	The first argument contains the complete location.

Return markup from a macro

KEYS (NO EVIL) C-c C-e h h · export HTML (macros expand during export)
DOOM ALIAS C-c l e h h · export HTML

A macro can produce Org markup that is then rendered by the exporter. Both labels become emphasized text.

COMPLETE ORG SOURCE

```
#+MACRO: label *$1*
{{{label(Ready)}}} for review.

{{{label(Approved)}}} for publication.
```

HTML EXPORT

Ready for review.

Approved for publication.

Generated by native Org. The frame and typography belong to this guide.

`*$1*`	Bold markup surrounds the argument.
`Export order`	Macro expansion happens before the resulting Org markup is rendered.

Compute a macro replacement

KEYS (NO EVIL) C-c C-e h h · export HTML (macros expand during export)
DOOM ALIAS C-c l e h h · export HTML

An eval macro runs an Emacs Lisp expression while exporting. Its argument arrives as a string, so this expression converts it to a number first.

COMPLETE ORG SOURCE

```
#+MACRO: twice (eval (* 2 (string-to-number $1)))
Twice seven is {{{twice(7)}}}.
```

HTML EXPORT

Twice seven is 14.

Generated by native Org. The frame and typography belong to this guide.

`(eval ...)`	Compute the replacement during export.
`string-to-number`	Convert the textual argument before arithmetic.
`Trusted source`	Exporting this document evaluates the macro expression.

Reuse document metadata

KEYS (NO EVIL) C-c C-e h h · export HTML (macros expand during export)
DOOM ALIAS C-c l e h h · export HTML

Built-in macros can read document keywords. Keep the title and author in metadata and reuse them in a sentence.

COMPLETE ORG SOURCE

```
#+TITLE: Small report
#+AUTHOR: Alex Researcher
This is {{{title}}}, prepared by {{{author}}}.
```

HTML EXPORT

This is Small report, prepared by Alex Researcher.

Generated by native Org. The frame and typography belong to this guide.

`{{{title}}}`	Read the TITLE keyword.
`{{{author}}}`	Read the AUTHOR keyword.

Load shared macro definitions

Compare with ORG-12-003 · Pass two macro arguments

KEYS (NO EVIL) C-c C-e h h · export HTML (macros expand during export)
DOOM ALIAS C-c l e h h · export HTML

A setup file can hold shared definitions. The companion file is bundled with this example and shown in the companion-inputs appendix.

COMPLETE ORG SOURCE

```
#+SETUPFILE: assets/macros.org
{{{project}}}: {{{quantity(12,cm)}}}.
```

Companion input: assets/macros.org

HTML EXPORT

Open Lab: 12 cm.

Generated by native Org. The frame and typography belong to this guide.

`#+SETUPFILE:`	Load document configuration, including macro definitions.
`assets/macros.org`	Bundled source; changing it affects every document that loads it.

ORG-13-000 · 10 EXAMPLES

Reuse and tangling

Reuse text, include files, expand named code and generate standalone scripts. These mechanisms solve different problems.

ORG-13-001 Define a text macro

ORG-13-002 Pass a macro argument

ORG-13-003 Include another Org file

ORG-13-004 Include a file as source code

ORG-13-005 Tangle one Bash script

ORG-13-006 Add an executable shebang

ORG-13-007 Combine blocks into one file

ORG-13-008 Expand a named code fragment

ORG-13-009 Expand Noweb only when tangling

ORG-13-010 Tangle to two destinations

Define a text macro

KEYS (NO EVIL) C-c C-e h h · export HTML (org-export-dispatch)
DOOM ALIAS C-c l e h h · export HTML

An export macro replaces a marker with text.

COMPLETE ORG SOURCE

```
#+MACRO: project Signal study
Welcome to {{{project}}}.
```

HTML EXPORT

Welcome to Signal study.

Generated by native Org. The frame and typography belong to this guide.

`#+MACRO: name text`	Defines an export-time text replacement.
`{{{project}}}`	Calls the macro.

Pass a macro argument

Compare with ORG-13-001 · Define a text macro

KEYS (NO EVIL) C-c C-e h h · export HTML (org-export-dispatch)
DOOM ALIAS C-c l e h h · export HTML

Use an argument to generate consistent labels.

COMPLETE ORG SOURCE

```
#+MACRO: sample Sample $1
Compare {{{sample(A)}}} with {{{sample(B)}}}.
```

HTML EXPORT

Compare Sample A with Sample B.

Generated by native Org. The frame and typography belong to this guide.

`$1`	First macro argument.
`{{{sample(A)}}}`	Calls sample with argument A.

Include another Org file

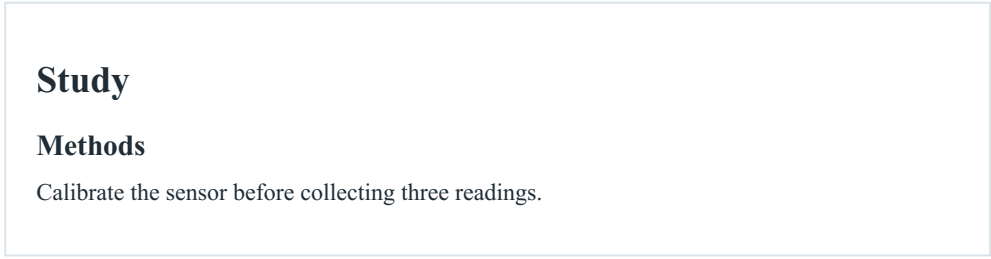
KEYS (NO EVIL) C-c C-e h h · export HTML (org-export-dispatch)
DOOM ALIAS C-c l e h h · export HTML

Include reusable text from a companion file. The asset is supplied in the ZIP.

COMPLETE ORG SOURCE

```
* Study
#+INCLUDE: "assets/methods.org"
```

HTML EXPORT



Study

Methods

Calibrate the sensor before collecting three readings.

Generated by native Org. The frame and typography belong to this guide.

`#+INCLUDE:`	Expands another file during export.
`Quoted path`	Relative to this Org document.

Include a file as source code

KEYS (NO EVIL) C-c C-e h h · export HTML (org-export-dispatch)
DOOM ALIAS C-c l e h h · export HTML

Display a file as code without evaluating it.

COMPLETE ORG SOURCE

```
#+INCLUDE: "assets/analyse.py" src python
```

HTML EXPORT

```
readings = [2, 5, 8]
print(sum(readings) / len(readings))
```

Generated by native Org. The frame and typography belong to this guide.

`src python`	Wrap included file content as a Python source block.

Tangle one Bash script

KEYS (NO EVIL) C-c C-v t · org-babel-tangle; C-c ' · edit block
DOOM ALIAS C-c l ' · edit source block

Extract the code into hello.sh. Tangling writes source files; it does not run the extracted script.

COMPLETE ORG SOURCE

```
#+BEGIN_SRC bash :tangle hello.sh
printf "hello\n"
#+END_SRC
```

TANGLED FILE(S)

```
FILE: hello.sh
printf "hello\n"
```

Actual generated file contents. Tangling does not execute these scripts.

`:tangle hello.sh`	Destination filename.
`C-c C-v t`	Run org-babel-tangle.

Add an executable shebang

Compare with ORG-13-005 · Tangle one Bash script

KEYS (NO EVIL) C-c C-v t · org-babel-tangle; C-c ' · edit block
DOOM ALIAS C-c l ' · edit source block

Write a shebang before the extracted code. Org also makes a shebang-bearing tangled file executable.

COMPLETE ORG SOURCE

```
#+BEGIN_SRC bash :tangle hello.sh :shebang "#!/usr/bin/env bash"
printf "hello\n"
#+END_SRC
```

TANGLED FILE(S)

```
FILE: hello.sh
#!/usr/bin/env bash
printf "hello\n"
```

Actual generated file contents. Tangling does not execute these scripts.

:shebang	First line identifying the interpreter.

Combine blocks into one file

KEYS (NO EVIL) C-c C-v t · org-babel-tangle; C-c ' · edit block
DOOM ALIAS C-c l ' · edit source block

Blocks targeting the same file are concatenated in document order.

COMPLETE ORG SOURCE

```
#+BEGIN_SRC bash :tangle run.sh
printf "first\n"
#+END_SRC

#+BEGIN_SRC bash :tangle run.sh
printf "second\n"
#+END_SRC
```

TANGLED FILE(S)

```
FILE: run.sh
printf "first\n"

printf "second\n"
```

Actual generated file contents. Tangling does not execute these scripts.

`Repeated :tangle target`	Both block bodies contribute to run.sh.

Expand a named code fragment

KEYS (NO EVIL) C-c C-v t · org-babel-tangle; C-c ' · edit block
DOOM ALIAS C-c l ' · edit source block

Insert the named fragment into the tangled script.

COMPLETE ORG SOURCE

```
#+NAME: greeting
#+BEGIN_SRC bash :tangle no
printf "hello\n"
#+END_SRC

#+BEGIN_SRC bash :tangle run.sh :noweb yes
<<greeting>>
printf "done\n"
#+END_SRC
```

TANGLED FILE(S)

```
FILE: run.sh
printf "hello\n"
printf "done\n"
```

Actual generated file contents. Tangling does not execute these scripts.

`#+NAME: greeting`	Names a reusable block.
`<<greeting>>`	Noweb reference to that block.
`:noweb yes`	Expand references during evaluation and tangling.
`:tangle no`	Do not write this fragment to a separate file.

Expand Noweb only when tangling

Compare with ORG-13-008 · Expand a named code fragment

KEYS (NO EVIL) C-c C-v t · org-babel-tangle; C-c ' · edit block
DOOM ALIAS C-c l ' · edit source block

Generate the same script while restricting expansion to tangling.

COMPLETE ORG SOURCE

```
#+NAME: greeting
#+BEGIN_SRC bash :tangle no
printf "hello\n"
#+END_SRC

#+BEGIN_SRC bash :tangle run.sh :noweb tangle
<<greeting>>
printf "done\n"
#+END_SRC
```

TANGLED FILE(S)

```
FILE: run.sh
printf "hello\n"
printf "done\n"
```

Actual generated file contents. Tangling does not execute these scripts.

`:noweb tangle` — Expand for tangling only; leave the source reference visible during export.

Tangle to two destinations

KEYS (NO EVIL) C-c C-v t · org-babel-tangle; C-c ' · edit block
DOOM ALIAS C-c l ' · edit source block

One Org document can produce scripts in multiple languages and files.

COMPLETE ORG SOURCE

```
#+BEGIN_SRC bash :tangle analyse.sh
printf "analysis\n"
#+END_SRC

#+BEGIN_SRC python :tangle analyse.py
print(sum([2, 5, 8]))
#+END_SRC
```

TANGLED FILE(S)

```
FILE: analyse.py
print(sum([2, 5, 8]))

FILE: analyse.sh
printf "analysis\n"
```

Actual generated file contents. Tangling does not execute these scripts.

`Distinct :tangle targets`

Each target collects only the blocks assigned to it.

ORG-14-000 · 18 EXAMPLES

Functions and inline calls

Call named Babel blocks without arguments, override defaults, pass data and compute values inside prose. Each action evaluates the indicated call or block in isolation.

ORG-14-001 Call a block without parameters

ORG-14-002 Call a function with a default input

ORG-14-003 Override one function parameter

ORG-14-004 Pass two named parameters

ORG-14-005 Keep an omitted parameter at its default

ORG-14-006 Pass a string to a function

ORG-14-007 Pass a named table to a function

ORG-14-008 Collect a function return value

ORG-14-009 Override how the function collects results

ORG-14-010 Format a call result as a table

ORG-14-011 Call a function inline without parameters

ORG-14-012 Pass a parameter in an inline call

ORG-14-013 Export prose containing an inline result

ORG-14-014 Evaluate code directly inside a sentence

ORG-14-015 Call a function from another Org file

ORG-14-016 Feed one function result into another

ORG-14-017 Export an inline result as plain text

ORG-14-018 Ingest a reusable function library

Call a block without parameters

KEYS (NO EVIL) C-c C-c at the indicated call/block · evaluate; C-c ' · edit source
DOOM ALIAS C-c l ' · edit source; C-c l e · export menu

A named block behaves like a callable unit. Evaluate the CALL line to insert its result beneath that line; the block definition is left intact.

COMPLETE ORG SOURCE

```
#+NAME: greeting
#+BEGIN_SRC emacs-lisp :results value
"Hello from Org"
#+END_SRC

#+CALL: greeting()
```

ACTION

C-c C-c · org-babel-lob-execute-maybe

`(org-babel-lob-execute-maybe)`

Cursor at the beginning of: #+CALL: greeting()

EDITOR BUFFER

```
#+NAME: greeting
#+BEGIN_SRC emacs-lisp :results value
"Hello from Org"
#+END_SRC

#+CALL: greeting()

#+RESULTS:
: Hello from Org
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`#+NAME: greeting`	Give the code block a unique name.
`greeting()`	The empty argument list is still written with parentheses.

Call a function with a default input

KEYS (NO EVIL) C-c C-c at the indicated call/block · evaluate; C-c ' · edit source
DOOM ALIAS C-c l ' · edit source; C-c l e · export menu

The function declares a default input. Calling it without arguments uses that value.

COMPLETE ORG SOURCE

```
#+NAME: double
#+BEGIN_SRC emacs-lisp :var n=3 :results value
(* 2 n)
#+END_SRC

#+CALL: double()
```

ACTION

C-c C-c · org-babel-lob-execute-maybe

`(org-babel-lob-execute-maybe)`

Cursor at the beginning of: #+CALL: double()

EDITOR BUFFER

```
#+NAME: double
#+BEGIN_SRC emacs-lisp :var n=3 :results value
(* 2 n)
#+END_SRC

#+CALL: double()

#+RESULTS:
: 6
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`:var n=3`	Default argument declared on the block.
`double()`	Use the default n=3.

Override one function parameter

Compare with ORG-14-002 · Call a function with a default input

KEYS (NO EVIL) C-c C-c at the indicated call/block · evaluate; C-c ' · edit source
DOOM ALIAS C-c l ' · edit source; C-c l e · export menu

Override the default at the call site. The function body and default remain unchanged.

COMPLETE ORG SOURCE

```
#+NAME: double
#+BEGIN_SRC emacs-lisp :var n=3 :results value
(* 2 n)
#+END_SRC

#+CALL: double(n=7)
```

ACTION

C-c C-c · org-babel-lob-execute-maybe

`(org-babel-lob-execute-maybe)`

Cursor at the beginning of: #+CALL: double(n=7)

EDITOR BUFFER

```
#+NAME: double
#+BEGIN_SRC emacs-lisp :var n=3 :results value
(* 2 n)
#+END_SRC

#+CALL: double(n=7)

#+RESULTS:
: 14
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`n=7`	Named argument passed by this call.
`Argument syntax`	Use n=7 inside parentheses, not :var n=7.

Pass two named parameters

KEYS (NO EVIL) C-c C-c at the indicated call/block · evaluate; C-c ' · edit source
DOOM ALIAS C-c l ' · edit source; C-c l e · export menu

Both parameters are explicit at the call site. Their names connect the inputs to the variables used by the block.

COMPLETE ORG SOURCE

```
#+NAME: scale
#+BEGIN_SRC emacs-lisp :var x=3 factor=2 :results value
(* x factor)
#+END_SRC

#+CALL: scale(x=7,factor=4)
```

ACTION

C-c C-c · org-babel-lob-execute-maybe

`(org-babel-lob-execute-maybe)`

Cursor at the beginning of: #+CALL: scale(x=7,factor=4)

EDITOR BUFFER

```
#+NAME: scale
#+BEGIN_SRC emacs-lisp :var x=3 factor=2 :results value
(* x factor)
#+END_SRC

#+CALL: scale(x=7,factor=4)

#+RESULTS:
: 28
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`:var x=3 factor=2`	Declare two default inputs.
`x=7,factor=4`	Override both arguments; commas separate them.

Keep an omitted parameter at its default

Compare with ORG-14-004 · Pass two named parameters

KEYS (NO EVIL) C-c C-c at the indicated call/block · evaluate; C-c ' · edit source
DOOM ALIAS C-c l ' · edit source; C-c l e · export menu

Override only x. The omitted factor keeps its default value of 2.

COMPLETE ORG SOURCE

```
#+NAME: scale
#+BEGIN_SRC emacs-lisp :var x=3 factor=2 :results value
(* x factor)
#+END_SRC

#+CALL: scale(x=7)
```

ACTION

C-c C-c · org-babel-lob-execute-maybe

`(org-babel-lob-execute-maybe)`

Cursor at the beginning of: #+CALL: scale(x=7)

EDITOR BUFFER

```
#+NAME: scale
#+BEGIN_SRC emacs-lisp :var x=3 factor=2 :results value
(* x factor)
#+END_SRC

#+CALL: scale(x=7)

#+RESULTS:
: 14
```

Actual text captured from Emacs; syntax colours are applied by the guide.

Partial override	Explicit arguments replace defaults individually.

Pass a string to a function

KEYS (NO EVIL) C-c C-c at the indicated call/block · evaluate; C-c ' · edit source
DOOM ALIAS C-c l ' · edit source; C-c l e · export menu

Quote a string parameter at the call site. Numeric parameters do not need those quotes.

COMPLETE ORG SOURCE

```
#+NAME: greet
#+BEGIN_SRC emacs-lisp :var name="reader" :results value
(concat "Hello, " name "!")
#+END_SRC

#+CALL: greet(name="Ada")
```

ACTION

C-c C-c · org-babel-lob-execute-maybe

`(org-babel-lob-execute-maybe)`

Cursor at the beginning of: #+CALL: greet(name="Ada")

EDITOR BUFFER

```
#+NAME: greet
#+BEGIN_SRC emacs-lisp :var name="reader" :results value
(concat "Hello, " name "!")
#+END_SRC

#+CALL: greet(name="Ada")

#+RESULTS:
: Hello, Ada!
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`name="Ada"`	A string argument.
`concat`	Combine the greeting and name.

Pass a named table to a function

KEYS (NO EVIL) C-c C-c at the indicated call/block · evaluate; C-c ' · edit source
DOOM ALIAS C-c l ' · edit source; C-c l e · export menu

The call supplies a table as a list of rows. This example adds the first cell of each row.

COMPLETE ORG SOURCE

```
#+NAME: readings
| 2 |
| 5 |
| 8 |

#+NAME: total
#+BEGIN_SRC emacs-lisp :var data=readings :results value
(apply #'+ (mapcar #'car data))
#+END_SRC

#+CALL: total(data=readings)
```

ACTION

C-c C-c · org-babel-lob-execute-maybe

`(org-babel-lob-execute-maybe)`

Cursor at the beginning of: #+CALL:

EDITOR BUFFER

```
#+NAME: readings
| 2 |
| 5 |
| 8 |

#+NAME: total
#+BEGIN_SRC emacs-lisp :var data=readings :results value
(apply #'+ (mapcar #'car data))
#+END_SRC

#+CALL: total(data=readings)

#+RESULTS:
: 15
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`data=readings`	Resolve the named table and pass its values.
`mapcar / car`	Select the first cell from each row.

Collect a function return value

KEYS (NO EVIL) C-c C-c at the indicated call/block · evaluate; C-c ' · edit source
DOOM ALIAS C-c l ' · edit source; C-c l e · export menu

The block prints one string and returns another. Value collection keeps the returned value.

COMPLETE ORG SOURCE

```
#+NAME: demo
#+BEGIN_SRC emacs-lisp :results value
(princ "printed")
"returned"
#+END_SRC

#+CALL: demo()
```

ACTION

C-c C-c · org-babel-lob-execute-maybe

`(org-babel-lob-execute-maybe)`

Cursor at the beginning of: #+CALL: demo()

EDITOR BUFFER

```
#+NAME: demo
#+BEGIN_SRC emacs-lisp :results value
(princ "printed")
"returned"
#+END_SRC

#+CALL: demo()

#+RESULTS:
: returned
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`:results value`	Collect the final Lisp value.
`princ`	Produces printed output, which is not the chosen result here.

Override how the function collects results

Compare with ORG-14-008 · Collect a function return value

KEYS (NO EVIL) C-c C-c at the indicated call/block · evaluate; C-c ' · edit source
DOOM ALIAS C-c l ' · edit source; C-c l e · export menu

Move a collection override inside the brackets before the argument list. The result now comes from printed output.

COMPLETE ORG SOURCE

```
#+NAME: demo
#+BEGIN_SRC emacs-lisp :results value
(princ "printed")
"returned"
#+END_SRC

#+CALL: demo[:results output]()
```

ACTION

C-c C-c · org-babel-lob-execute-maybe

`(org-babel-lob-execute-maybe)`

Cursor at the beginning of: #+CALL: demo[:results output]()

EDITOR BUFFER

```
#+NAME: demo
#+BEGIN_SRC emacs-lisp :results value
(princ "printed")
"returned"
#+END_SRC

#+CALL: demo[:results output]()

#+RESULTS:
: printed
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`[:results output]`	Inside headers alter execution of the called block.
`()`	The function still takes no arguments.

Format a call result as a table

KEYS (NO EVIL) C-c C-c at the indicated call/block · evaluate; C-c ' · edit source
DOOM ALIAS C-c l ' · edit source; C-c l e · export menu

End headers describe how the returned value is inserted at the call site. This list of rows becomes an Org table.

COMPLETE ORG SOURCE

```
#+NAME: items
#+BEGIN_SRC emacs-lisp :results value
'(("Ada" 3) ("Bea" 5))
#+END_SRC

#+CALL: items() :results value table
```

ACTION

C-c C-c · org-babel-lob-execute-maybe

`(org-babel-lob-execute-maybe)`

Cursor at the beginning of: #+CALL: items() :results value table

EDITOR BUFFER

```
#+NAME: items
#+BEGIN_SRC emacs-lisp :results value
'(("Ada" 3) ("Bea" 5))
#+END_SRC

#+CALL: items() :results value table

#+RESULTS:
| Ada | 3 |
| Bea | 5 |
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`After ()`	End headers apply to the result at the call site.
`:results value table`	Insert the returned rows as an Org table.

Call a function inline without parameters

KEYS (NO EVIL) C-c C-c at the indicated call/block · evaluate; C-c ' · edit source
DOOM ALIAS C-c l ' · edit source; C-c l e · export menu

An inline call evaluates the same kind of named block inside a sentence. The inserted results macro records the value in the Org buffer.

COMPLETE ORG SOURCE

```
#+NAME: greeting
#+BEGIN_SRC emacs-lisp :results value :exports none
"Hello from Org"
#+END_SRC

Message: call_greeting().
```

ACTION

C-c C-c · org-babel-lob-execute-maybe

```
(org-babel-lob-execute-maybe)
```

Cursor at the beginning of: call_greeting

EDITOR BUFFER

```
#+NAME: greeting
#+BEGIN_SRC emacs-lisp :results value :exports none
"Hello from Org"
#+END_SRC

Message: call_greeting() {{{results(=Hello from Org=)}}}.
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`call_greeting()`	Inline call syntax: no #+CALL keyword.
`{{{results(...)}}}`	The stored inline result, produced by Babel.

Pass a parameter in an inline call

Compare with ORG-14-011 · Call a function inline without parameters

KEYS (NO EVIL) C-c C-c at the indicated call/block · evaluate; C-c ' · edit source
DOOM ALIAS C-c l ' · edit source; C-c l e · export menu

Pass a parameter from prose. The function is evaluated at the inline call position.

COMPLETE ORG SOURCE

```
#+NAME: double
#+BEGIN_SRC emacs-lisp :var n=3 :results value :exports none
(* 2 n)
#+END_SRC

The result is call_double(n=7).
```

EDITOR BUFFER

```
#+NAME: double
#+BEGIN_SRC emacs-lisp :var n=3 :results value :exports none
(* 2 n)
#+END_SRC

The result is call_double(n=7) {{{results(=14=)}}}.
```

Actual text captured from Emacs; syntax colours are applied by the guide.

ACTION

C-c C-c · org-babel-lob-execute-maybe

`(org-babel-lob-execute-maybe)`

Cursor at the beginning of: call_double

`call_double(n=7)`	Inline call with a named argument.
`:exports none`	Hide the function definition in document export.

Export prose containing an inline result

Compare with ORG-14-012 · Pass a parameter in an inline call

KEYS (NO EVIL) C-c C-c at the indicated call/block · evaluate; C-c ' · edit source
DOOM ALIAS C-c l ' · edit source; C-c l e · export menu

Evaluate the inline call, then export the document. The stored result appears in the sentence; the hidden function definition does not.

COMPLETE ORG SOURCE

```
#+NAME: double
#+BEGIN_SRC emacs-lisp :var n=3 :results value :exports none
(* 2 n)
#+END_SRC

The result is call_double(n=7).
```

HTML AFTER COMMAND

The result is 14.

Generated by native Org. The frame and typography belong to this guide.

ACTION

C-c C-c · org-babel-lob-execute-maybe

`(org-babel-lob-execute-maybe)`

Cursor at the beginning of: call_double

Evaluation first	C-c C-c on the inline call stores the result.
Export second	C-c C-e h h publishes the sentence.

Evaluate code directly inside a sentence

KEYS (NO EVIL) C-c C-c at the indicated call/block · evaluate; C-c ' · edit source
DOOM ALIAS C-c l ' · edit source; C-c l e · export menu

Inline source syntax contains code itself. Inline call syntax instead refers to a separately named block.

COMPLETE ORG SOURCE

```
The mean is src_emacs-lisp{(/ (+ 2 5 8) 3.0)}.
```

ACTION

C-c C-c · org-babel-execute-src-block

`(org-babel-execute-src-block)`

Cursor at the beginning of: src_emacs-lisp

HTML AFTER COMMAND

The mean is `5.0`.

Generated by native Org. The frame and typography belong to this guide.

`src_emacs-lisp{...}`	An inline source block with a Lisp expression.
`call_name(...)`	Use this when calling a named block instead.

Call a function from another Org file

KEYS (NO EVIL) C-c C-c at the indicated call/block · evaluate; C-c ' · edit source
DOOM ALIAS C-c l ' · edit source; C-c l e · export menu

The file path and block name identify the function. The companion file is included in the ZIP and in the HTML companion-inputs section.

COMPLETE ORG SOURCE

```
#+CALL: assets/functions.org:scale(x=7,factor=4)
```

Companion input: assets/functions.org

ACTION

C-c C-c · org-babel-lob-execute-maybe

`(org-babel-lob-execute-maybe)`

Cursor at the beginning of: #+CALL:

EDITOR BUFFER

```
#+CALL: assets/functions.org:scale(x=7,factor=4)

#+RESULTS:
: 28
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`file.org:name(...)`	Call a named block in another file.
`assets/functions.org`	Contains the scale block with x and factor parameters.

Feed one function result into another

KEYS (NO EVIL) C-c C-c at the indicated call/block · evaluate; C-c ' · edit source
DOOM ALIAS C-c l ' · edit source; C-c l e · export menu

A header argument can call a named block to obtain its input. This evaluates the dependency before the consumer block.

COMPLETE ORG SOURCE

```
#+NAME: double
#+BEGIN_SRC emacs-lisp :var n=3 :results value
(* 2 n)
#+END_SRC

#+BEGIN_SRC emacs-lisp :var x=double(n=7) :results value
(+ x 1)
#+END_SRC
```

ACTION

C-c C-c · org-babel-execute-src-block

`(org-babel-execute-src-block)`

Cursor at the beginning of: #+BEGIN_SRC emacs-lisp :var x=

EDITOR BUFFER

```
#+NAME: double
#+BEGIN_SRC emacs-lisp :var n=3 :results value
(* 2 n)
#+END_SRC

#+BEGIN_SRC emacs-lisp :var x=double(n=7) :results value
(+ x 1)
#+END_SRC

#+RESULTS:
: 15
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`:var x=double(n=7)`	Run double with n=7; bind its result to x.
`Consumer result`	14 + 1 = 15.

Export an inline result as plain text

Compare with ORG-14-013 · Export prose containing an inline result

KEYS (NO EVIL) C-c C-c at inline call · evaluate; C-c C-e h h · export
DOOM ALIAS C-c l e h h · export HTML

End headers on an inline call can change its result format. Raw output removes the default verbatim styling from this number.

COMPLETE ORG SOURCE

```
#+NAME: double
#+BEGIN_SRC emacs-lisp :var n=3 :results value :exports none
(* 2 n)
#+END_SRC

The result is call_double(n=7)[:results raw].
```

ACTION

C-c C-c · org-babel-lob-execute-maybe

`(org-babel-lob-execute-maybe)`

Cursor at the beginning of: call_double

HTML AFTER COMMAND

The result is 14.

Generated by native Org. The frame and typography belong to this guide.

`[:results raw] after ()`	Inline end headers format the returned result.
`Plain text`	The number uses the surrounding paragraph style.

Ingest a reusable function library

Compare with ORG-14-015 · Call a function from another Org file

KEYS (NO EVIL) C-c C-v i · ingest assets/functions.org; C-c C-c on CALL · evaluate
DOOM ALIAS C-c l ' · edit source blocks

Ingest the companion file into the Library of Babel, then call its block by name. This registration belongs to the Emacs session; another session must ingest the file again.

COMPLETE ORG SOURCE

```
# First ingest assets/functions.org with C-c C-v i.
#+CALL: scale(x=7,factor=4)
```

Companion input: assets/functions.org

ACTION

C-c C-v i · org-babel-lob-ingest; then C-c C-c on CALL

```
(let ((org-babel-library-of-babel nil))
  (org-babel-lob-ingest "assets/functions.org")
  (org-babel-lob-execute-maybe))
```

Cursor at the beginning of: #+CALL:

EDITOR BUFFER

```
# First ingest assets/functions.org with C-c C-v i.
#+CALL: scale(x=7,factor=4)

#+RESULTS:
: 28
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`org-babel-lob-ingest`	Read named code blocks into the session's Library of Babel.
`Local let binding`	Isolate this demonstration from other examples in the build.

ORG-15-000 · 14 EXAMPLES

Result formats and files

Choose how to collect a result, represent it and insert it. A return value, printed output, result format and export policy answer different questions.

ORG-15-001 Return a plain scalar

ORG-15-002 Preserve a string literally

ORG-15-003 Render the same string as Org markup

ORG-15-004 Return a table

ORG-15-005 Return a bullet list

ORG-15-006 Wrap output in a results drawer

ORG-15-007 Return HTML markup

ORG-15-008 Return LaTeX markup

ORG-15-009 Write a text file as the result

ORG-15-010 Export a downloadable file link

ORG-15-011 Replace an earlier result

ORG-15-012 Append to earlier results

ORG-15-013 Prepend to earlier results

ORG-15-014 Evaluate without inserting a result

Return a plain scalar

KEYS (NO EVIL) C-c C-c at the indicated call/block · evaluate; C-c ' · edit source
DOOM ALIAS C-c l ' · edit source; C-c l e · export menu

Start with one returned value. Babel records it as a fixed-width result line.

COMPLETE ORG SOURCE

```
#+BEGIN_SRC emacs-lisp :results value
(+ 2 5)
#+END_SRC
```

EVALUATED ORG BUFFER

```
#+BEGIN_SRC emacs-lisp :results value
(+ 2 5)
#+END_SRC

#+RESULTS:
: 7
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`:results value`	Collect the returned value.
`: 7`	Org fixed-width text representation.

Preserve a string literally

KEYS (NO EVIL) C-c C-c at the indicated call/block · evaluate; C-c ' · edit source
DOOM ALIAS C-c l ' · edit source; C-c l e · export menu

Verbatim collection keeps the asterisks as text. They do not become bold markup in this exported result.

COMPLETE ORG SOURCE

```
#+BEGIN_SRC emacs-lisp :results value verbatim :exports results
"*Ready*"
#+END_SRC
```

HTML AFTER EVALUATION

```
"*Ready*"
```

Generated by native Org. The frame and typography belong to this guide.

`verbatim`	Keep the result literal.
`:exports results`	Publish the result without the source block.

Render the same string as Org markup

Compare with ORG-15-002 · Preserve a string literally

KEYS (NO EVIL) C-c C-c at the indicated call/block · evaluate; C-c ' · edit source
DOOM ALIAS C-c l ' · edit source; C-c l e · export menu

Changing verbatim to raw makes the result ordinary Org content. The same asterisks now mark bold text.

COMPLETE ORG SOURCE

```
#+BEGIN_SRC emacs-lisp :results value raw :exports results
 "*Ready*"
#+END_SRC
```

HTML AFTER EVALUATION

Ready

Generated by native Org. The frame and typography belong to this guide.

`raw`	Insert the returned string as Org markup.

Return a table

KEYS (NO EVIL) C-c C-c at the indicated call/block · evaluate; C-c ' · edit source
DOOM ALIAS C-c l ' · edit source; C-c l e · export menu

A nested list gives Org rows and columns. The stored table remains editable like any other Org table.

COMPLETE ORG SOURCE

```
#+BEGIN_SRC emacs-lisp :results value table
'(("Ada" 3) ("Bea" 5))
#+END_SRC
```

EVALUATED ORG BUFFER

```
#+BEGIN_SRC emacs-lisp :results value table
'(("Ada" 3) ("Bea" 5))
#+END_SRC

#+RESULTS:
| Ada | 3 |
| Bea | 5 |
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`Outer list`	Table rows.
`Inner list`	Cells in one row.

Return a bullet list

Compare with ORG-15-004 · Return a table

KEYS (NO EVIL) C-c C-c at the indicated call/block · evaluate; C-c ' · edit source
DOOM ALIAS C-c l ' · edit source; C-c l e · export menu

A flat list can be inserted as ordinary Org list items.

COMPLETE ORG SOURCE

```
#+BEGIN_SRC emacs-lisp :results value list
'("Collect" "Check" "Publish")
#+END_SRC
```

EVALUATED ORG BUFFER

```
#+BEGIN_SRC emacs-lisp :results value list
'("Collect" "Check" "Publish")
#+END_SRC

#+RESULTS:
- Collect
- Check
- Publish
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`:results list`	One list item for each returned element.

Wrap output in a results drawer

KEYS (NO EVIL) C-c C-c at the indicated call/block · evaluate; C-c ' · edit source
DOOM ALIAS C-c l ' · edit source; C-c l e · export menu

A drawer makes multiline output collapsible. The result remains visible here as actual buffer text.

COMPLETE ORG SOURCE

```
#+BEGIN_SRC emacs-lisp :results output drawer
(princ "first line\nsecond line")
#+END_SRC
```

EVALUATED ORG BUFFER

```
#+BEGIN_SRC emacs-lisp :results output drawer
(princ "first line\nsecond line")
#+END_SRC

#+RESULTS:
:results:
first line
second line
:end:
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`drawer`	Wrap inserted text in a RESULTS drawer.
`TAB at drawer`	Collapse or expand it in the editor.

Return HTML markup

KEYS (NO EVIL) C-c C-c at the indicated call/block · evaluate; C-c ' · edit source
DOOM ALIAS C-c l ' · edit source; C-c l e · export menu

The html format inserts an HTML export block. This preview comes from the native HTML exporter.

COMPLETE ORG SOURCE

```
#+BEGIN_SRC emacs-lisp :results value html :exports results
"<p><strong>Ready</strong> for review.</p>"
#+END_SRC
```

HTML AFTER EVALUATION

Ready for review.

Generated by native Org. The frame and typography belong to this guide.

`html`	Wrap the result in #+BEGIN_EXPORT html.
`Backend-specific`	This markup targets HTML output.

Return LaTeX markup

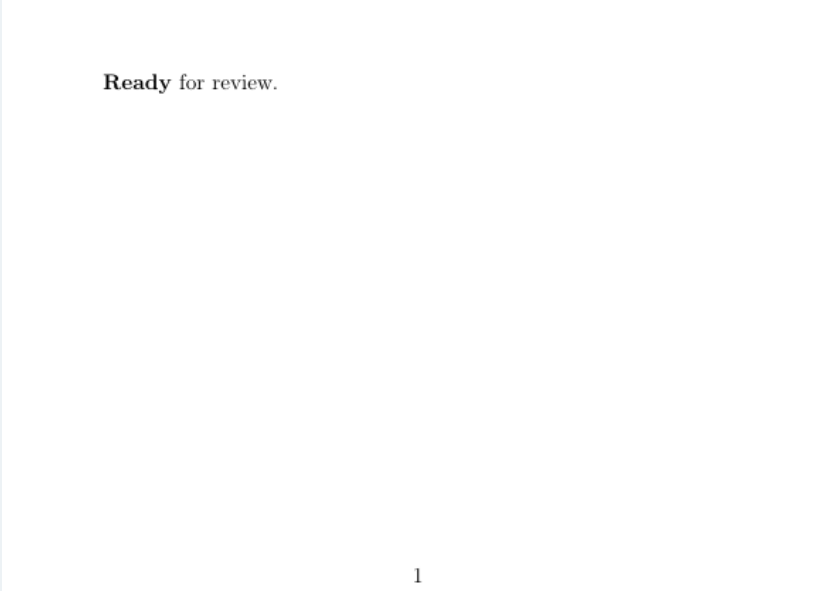
KEYS (NO EVIL) C-c C-c · evaluate block; C-c C-e l p · export PDF
DOOM ALIAS C-c l ' · edit source; C-c l e · export menu

Use latex for content intended for the LaTeX exporter. The preview shows the compiled PDF.

COMPLETE ORG SOURCE

```
#+LATEX_HEADER: \usepackage[a6paper,landscape,margin=12mm]{geometry}
#+BEGIN_SRC emacs-lisp :results value latex :exports results
"\\textbf{Ready} for review."
#+END_SRC
```

NATIVE PDF AFTER EVALUATION



Page 1

Actual native Org → LaTeX → pdfLaTeX output; preview scaled to fit.

`latex`	Wrap the result in #+BEGIN_EXPORT latex.
`Native PDF`	The build exports LaTeX and runs pdfLaTeX.

Write a text file as the result

KEYS (NO EVIL) C-c C-c at the indicated call/block · evaluate; C-c ' · edit source
DOOM ALIAS C-c l ' · edit source; C-c l e · export menu

With file results, Babel writes the returned text to the named file and inserts a link. The build checks the file contents as well.

COMPLETE ORG SOURCE

```
#+BEGIN_SRC emacs-lisp :results value file :file message.txt :file-desc "Download message"
"Ready for review."
#+END_SRC
```

EVALUATED ORG BUFFER

```
#+BEGIN_SRC emacs-lisp :results value file :file message.txt :file-desc "Download message"
"Ready for review."
#+END_SRC

#+RESULTS:
[[file:message.txt][Download message]]
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`:file message.txt`	Output path relative to this example.
`:file-desc`	Readable text for the inserted link.

Export a downloadable file link

Compare with ORG-15-009 · Write a text file as the result

KEYS (NO EVIL) C-c C-c at the indicated call/block · evaluate; C-c ' · edit source
DOOM ALIAS C-c l ' · edit source; C-c l e · export menu

Keep the same file result and export it. In the self-contained HTML guide, the download embeds the generated file.

COMPLETE ORG SOURCE

```
#+BEGIN_SRC emacs-lisp :results value file :file message.txt :file-desc "Download
message" :exports results
 "Ready for review."
#+END_SRC
```

HTML AFTER EVALUATION

Download message

Generated by native Org. The frame and typography belong to this guide.

`:exports results`	The exported content is the download link.
`Generated file`	Contains the exact text returned by the block.

Replace an earlier result

KEYS (NO EVIL) C-c C-c at the indicated call/block · evaluate; C-c ' · edit source
DOOM ALIAS C-c l ' · edit source; C-c l e · export menu

Replace is the default insertion policy. The new result takes the place of the stored old one.

COMPLETE ORG SOURCE

```
#+BEGIN_SRC emacs-lisp :results value replace
"new"
#+END_SRC

#+RESULTS:
: old
```

EVALUATED ORG BUFFER

```
#+BEGIN_SRC emacs-lisp :results value replace
"new"
#+END_SRC

#+RESULTS:
: new
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`replace`	Keep only the latest inserted result.

Append to earlier results

Compare with ORG-15-011 · Replace an earlier result

KEYS (NO EVIL) C-c C-c at the indicated call/block · evaluate; C-c ' · edit source
DOOM ALIAS C-c l ' · edit source; C-c l e · export menu

Change only the insertion policy. The old result is retained and the new result is added below it.

COMPLETE ORG SOURCE

```
#+BEGIN_SRC emacs-lisp :results value append
 "new"
 #+END_SRC

 #+RESULTS:
 : old
```

EVALUATED ORG BUFFER

```
#+BEGIN_SRC emacs-lisp :results value append
"new"
#+END_SRC

#+RESULTS:
: old
: new
```

Actual text captured from Emacs; syntax colours are applied by the guide.

append	Keep history with the newest result at the bottom.

Prepend to earlier results

Compare with ORG-15-012 · Append to earlier results

KEYS (NO EVIL) C-c C-c at the indicated call/block · evaluate; C-c ' · edit source
DOOM ALIAS C-c l ' · edit source; C-c l e · export menu

Prepend preserves the old result but places the newest result above it.

COMPLETE ORG SOURCE

```
#+BEGIN_SRC emacs-lisp :results value prepend
 "new"
 #+END_SRC

 #+RESULTS:
 : old
```

EVALUATED ORG BUFFER

```
#+BEGIN_SRC emacs-lisp :results value prepend
"new"
#+END_SRC

#+RESULTS:
: new
: old
```

Actual text captured from Emacs; syntax colours are applied by the guide.

prepend	Keep history with the newest result at the top.

Evaluate without inserting a result

Compare with ORG-15-001 · Return a plain scalar

KEYS (NO EVIL) C-c C-c at the indicated call/block · evaluate; C-c ' · edit source
DOOM ALIAS C-c l ' · edit source; C-c l e · export menu

Silent execution leaves the Org buffer unchanged. Emacs reports the result in its echo area; this panel deliberately shows the unchanged buffer.

COMPLETE ORG SOURCE

```
#+BEGIN_SRC emacs-lisp :results value silent
(+ 2 5)
#+END_SRC
```

EVALUATED ORG BUFFER

```
#+BEGIN_SRC emacs-lisp :results value silent
(+ 2 5)
#+END_SRC
```

Actual text captured from Emacs; syntax colours are applied by the guide.

`silent`	Return the value without inserting #+RESULTS.
`Editor view`	No result line is added below the block.

ORG-16-000 · 12 EXAMPLES

Citations and publications

Publication keys, in-text citations and printed bibliographies. The executed examples use Org-cite with its built-in basic processor and a bundled bibliography. The citation setup reference explains Citar, CSL and BibLaTeX.

ORG-16-001 Cite a publication and print its reference

ORG-16-002 Cite two publications together

ORG-16-003 Make the author part of the sentence

ORG-16-004 Show author and year separately

ORG-16-005 Add a prefix and page locator

ORG-16-006 Add context around a citation group

ORG-16-007 Switch to numeric references

ORG-16-008 Include an uncited publication

ORG-16-009 Put a citation in a footnote

ORG-16-010 Export author–year references to PDF

ORG-16-011 Export numeric references to PDF

ORG-16-012 Inspect the publication metadata

Cite a publication and print its reference

KEYS (NO EVIL) C-c C-x @ · insert citation; C-c C-e h h · export HTML
DOOM ALIAS C-c l @ · insert citation; C-c l e · export menu

The citation key resolves to a record in the bundled bibliography. The citation and the printed reference are separate elements.

COMPLETE ORG SOURCE

```
#+BIBLIOGRAPHY: assets/references.bib
#+CITE_EXPORT: basic author-year

Literate programming [cite:@knuth1984] joins explanation and code.

#+PRINT_BIBLIOGRAPHY:
```

Companion input: assets/references.bib

HTML EXPORT

Literate programming (Knuth, Donald E., 1984) joins explanation and code.

Knuth, Donald E. (1984). *Literate Programming*, The Computer Journal.

Generated by native Org. The frame and typography belong to this guide.

`@knuth1984`	Key in assets/references.bib.
`#+BIBLIOGRAPHY:`	Bibliographic input file.
`#+PRINT_BIBLIOGRAPHY:`	Print the cited references here.

Cite two publications together

Compare with ORG-16-001 · Cite a publication and print its reference

KEYS (NO EVIL) C-c C-x @ · insert citation; C-c C-e h h · export HTML
DOOM ALIAS C-c l @ · insert citation; C-c l e · export menu

Semicolons separate keys inside one citation group. Both cited publications appear in the reference list.

COMPLETE ORG SOURCE

```
#+BIBLIOGRAPHY: assets/references.bib
#+CITE_EXPORT: basic author-year

Related approaches [cite:@knuth1984;@schulte2012] support this report.

#+PRINT_BIBLIOGRAPHY:
```

Companion input: assets/references.bib

HTML EXPORT

Related approaches (Knuth, Donald E., 1984, Schulte, Eric and Davison, Dan and Dye, Thomas and Dominik, Carsten, 2012) support this report.

Knuth, Donald E. (1984). *Literate Programming*, The Computer Journal.

Schulte, Eric and Davison, Dan and Dye, Thomas and Dominik, Carsten (2012). *A Multi-Language Computing Environment for Literate Programming and Reproducible Research*, Journal of Statistical Software.

Generated by native Org. The frame and typography belong to this guide.

;	Separate reference keys within one citation.

Make the author part of the sentence

Compare with ORG-16-001 · Cite a publication and print its reference

KEYS (NO EVIL) C-c C-x @ · insert citation; C-c C-e h h · export HTML
DOOM ALIAS C-c l @ · insert citation; C-c l e · export menu

The text style makes the citation fit a narrative sentence.

COMPLETE ORG SOURCE

```
#+BIBLIOGRAPHY: assets/references.bib
#+CITE_EXPORT: basic author-year

[cite/text:@knuth1984] introduced literate programming.

#+PRINT_BIBLIOGRAPHY:
```

Companion input: assets/references.bib

HTML EXPORT

Knuth, Donald E. (1984) introduced literate programming.

Knuth, Donald E. (1984). *Literate Programming*, The Computer Journal.

Generated by native Org. The frame and typography belong to this guide.

`/text`	Narrative citation style.
`Same key`	Formatting changes while the bibliographic record stays the same.

Show author and year separately

Compare with ORG-16-003 · Make the author part of the sentence

KEYS (NO EVIL) C-c C-x @ · insert citation; C-c C-e h h · export HTML
DOOM ALIAS C-c l @ · insert citation; C-c l e · export menu

Author-only and noauthor styles expose different parts of the same reference.

COMPLETE ORG SOURCE

```
#+BIBLIOGRAPHY: assets/references.bib
#+CITE_EXPORT: basic author-year

Author: [cite/author:@knuth1984]

Year: [cite/noauthor:@knuth1984].

#+PRINT_BIBLIOGRAPHY:
```

Companion input: assets/references.bib

HTML EXPORT

Author: Knuth, Donald E.

Year: (1984).

Knuth, Donald E. (1984). *Literate Programming*, The Computer Journal.

Generated by native Org. The frame and typography belong to this guide.

/author	Author-only citation.
/noauthor	Omit the author from the citation, leaving the year here.

Add a prefix and page locator

Compare with ORG-16-001 · Cite a publication and print its reference

KEYS (NO EVIL) C-c C-x @ · insert citation; C-c C-e h h · export HTML
DOOM ALIAS C-c l @ · insert citation; C-c l e · export menu

Text before and after a key supplies context for that reference. The page locator points into the cited publication.

COMPLETE ORG SOURCE

```
#+BIBLIOGRAPHY: assets/references.bib
#+CITE_EXPORT: basic author-year

Background [cite:see @knuth1984 p. 97].

#+PRINT_BIBLIOGRAPHY:
```

Companion input: assets/references.bib

HTML EXPORT

Background (see Knuth, Donald E., 1984 p. 97).

Knuth, Donald E. (1984). *Literate Programming*, The Computer Journal.

Generated by native Org. The frame and typography belong to this guide.

see	Prefix attached to this reference.
p. 97	Suffix giving a page locator.

Add context around a citation group

Compare with ORG-16-002 · Cite two publications together

KEYS (NO EVIL) C-c C-x @ · insert citation; C-c C-e h h · export HTML
DOOM ALIAS C-c l @ · insert citation; C-c l e · export menu

The text before the first semicolon and after the last semicolon belongs to the whole citation.

COMPLETE ORG SOURCE

```
#+BIBLIOGRAPHY: assets/references.bib
#+CITE_EXPORT: basic author-year

Background [cite:compare ;@knuth1984;@schulte2012; for two approaches].

#+PRINT_BIBLIOGRAPHY:
```

Companion input: assets/references.bib

HTML EXPORT

Background (compare Knuth, Donald E., 1984, Schulte, Eric and Davison, Dan and Dye, Thomas and Dominik, Carsten, 2012 for two approaches).

Knuth, Donald E. (1984). *Literate Programming*, The Computer Journal.

Schulte, Eric and Davison, Dan and Dye, Thomas and Dominik, Carsten (2012). *A Multi-Language Computing Environment for Literate Programming and Reproducible Research*, Journal of Statistical Software.

Generated by native Org. The frame and typography belong to this guide.

`Group prefix / suffix`	Placed outside the individual reference entries.
`Semicolons`	Separate affixes and reference items.

Switch to numeric references

Compare with ORG-16-002 · Cite two publications together

KEYS (NO EVIL) C-c C-x @ · insert citation; C-c C-e h h · export HTML
DOOM ALIAS C-c l @ · insert citation; C-c l e · export menu

Change the processor options while keeping the same two publications. Numeric citation markers correspond to numbered bibliography entries.

COMPLETE ORG SOURCE

```
#+BIBLIOGRAPHY: assets/references.bib
#+CITE_EXPORT: basic numeric numeric

Related approaches [cite:@knuth1984;@schulte2012] support this report.

#+PRINT_BIBLIOGRAPHY:
```

Companion input: assets/references.bib

HTML EXPORT

Related approaches (1, 2) support this report.

[1] Knuth, Donald E., *Literate Programming*, The Computer Journal, 1984.

[2] Schulte, Eric and Davison, Dan and Dye, Thomas and Dominik, Carsten, *A Multi-Language Computing Environment for Literate Programming and Reproducible Research*, Journal of Statistical Software, 2012.

Generated by native Org. The frame and typography belong to this guide.

`basic numeric numeric`	Numeric bibliography labels and default numeric citation style.
`basic`	Built-in processor; no CSL or BibTeX installation needed.

Include an uncited publication

KEYS (NO EVIL) C-c C-x @ · insert citation; C-c C-e h h · export HTML
DOOM ALIAS C-c l @ · insert citation; C-c l e · export menu

Nocite includes a reference in the bibliography without printing a normal citation at that position.

COMPLETE ORG SOURCE

```
#+BIBLIOGRAPHY: assets/references.bib
#+CITE_EXPORT: basic author-year

The Org workflow is described in [cite:@schulte2012].

[cite/nocite:@knuth1984]

#+PRINT_BIBLIOGRAPHY:
```

Companion input: assets/references.bib

HTML EXPORT

The Org workflow is described in (Schulte, Eric and Davison, Dan and Dye, Thomas and Dominik, Carsten, 2012).

Knuth, Donald E. (1984). *Literate Programming*, The Computer Journal.

Schulte, Eric and Davison, Dan and Dye, Thomas and Dominik, Carsten (2012). *A Multi-Language Computing Environment for Literate Programming and Reproducible Research*, Journal of Statistical Software.

Generated by native Org. The frame and typography belong to this guide.

`/nocite`	Include the keyed record without a visible citation marker.
`Reference list`	Contains both the cited and the additional publication.

Put a citation in a footnote

Compare with ORG-16-001 · Cite a publication and print its reference

KEYS (NO EVIL) C-c C-x @ · insert citation; C-c C-e h h · export HTML
DOOM ALIAS C-c l @ · insert citation; C-c l e · export menu

The note style moves the citation into a footnote. The bibliography still lists the publication.

COMPLETE ORG SOURCE

```
#+BIBLIOGRAPHY: assets/references.bib
#+CITE_EXPORT: basic author-year

This report follows a literate approach[cite/note:@knuth1984].

#+PRINT_BIBLIOGRAPHY:
```

Companion input: assets/references.bib

HTML EXPORT

This report follows a literate approach.[1]

Knuth, Donald E. (1984). *Literate Programming*, The Computer Journal.

Footnotes:

1

Knuth, Donald E. (1984)

Generated by native Org. The frame and typography belong to this guide.

`/note`	Footnote citation with the basic processor.
`Bibliography`	Independent of the citation position.

Export author–year references to PDF

Compare with ORG-16-001 · Cite a publication and print its reference

KEYS (NO EVIL) C-c C-x @ · insert citation; C-c C-e l p · export PDF
DOOM ALIAS C-c l @ · insert citation; C-c l e · export menu

The same basic processor also supports LaTeX output. Add a heading explicitly, because basic does not supply the bibliography heading.

COMPLETE ORG SOURCE

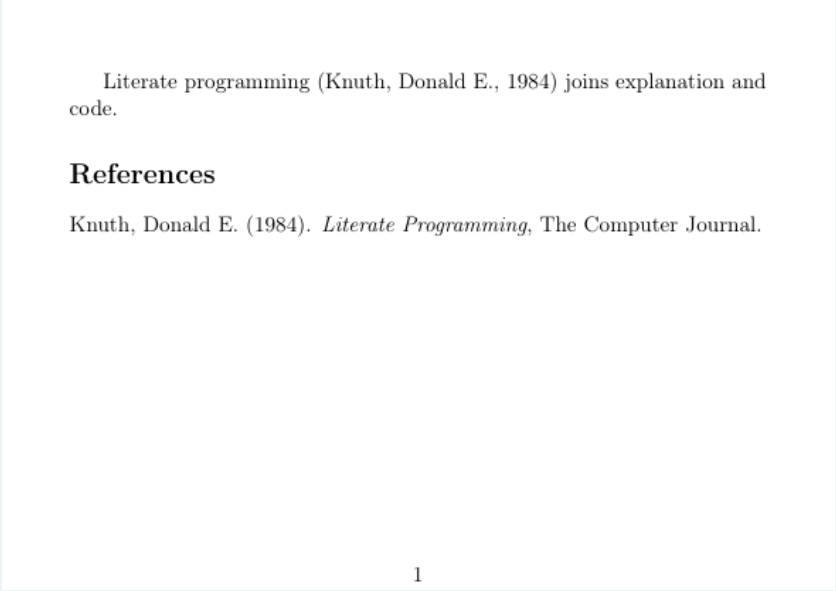
```
#+LATEX_HEADER: \usepackage[a6paper,landscape,margin=12mm]{geometry}
#+BIBLIOGRAPHY: assets/references.bib
#+CITE_EXPORT: basic author-year

Literate programming [cite:@knuth1984] joins explanation and code.

* References
#+PRINT_BIBLIOGRAPHY:
```

Companion input: assets/references.bib

NATIVE PDF EXPORT


Literate programming (Knuth, Donald E., 1984) joins explanation and code.

References

Knuth, Donald E. (1984). *Literate Programming*, The Computer Journal.

1

Page 1

Actual native Org → LaTeX → pdfLaTeX output; preview scaled to fit.

`basic`	Produces the reference text directly for LaTeX.
`No biber step`	This example compiles with pdfLaTeX alone.

Export numeric references to PDF

Compare with ORG-16-010 · Export author–year references to PDF

KEYS (NO EVIL) C-c C-x @ · insert citation; C-c C-e l p · export PDF
DOOM ALIAS C-c l @ · insert citation; C-c l e · export menu

Only the citation configuration changes from the preceding PDF example.

COMPLETE ORG SOURCE

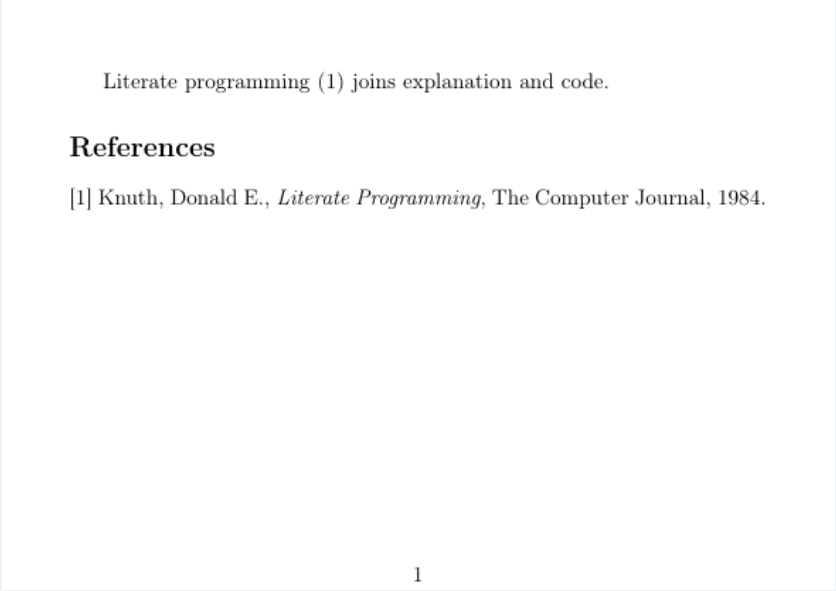
```
#+LATEX_HEADER: \usepackage[a6paper,landscape,margin=12mm]{geometry}
#+BIBLIOGRAPHY: assets/references.bib
#+CITE_EXPORT: basic numeric numeric

Literate programming [cite:@knuth1984] joins explanation and code.

* References
#+PRINT_BIBLIOGRAPHY:
```

Companion input: assets/references.bib

NATIVE PDF EXPORT


Literate programming (1) joins explanation and code.

References

[1] Knuth, Donald E., *Literate Programming*, The Computer Journal, 1984.

1

Page 1

Actual native Org → LaTeX → pdfLaTeX output; preview scaled to fit.

`Numeric mode` Changes citation markers and bibliography labels together.

Inspect the publication metadata

KEYS (NO EVIL) C-x C-f · open assets/references.bib; C-c C-e h h · export
DOOM ALIAS C-c l @ · select a publication in the Org document

The companion file contains two real journal articles. Stable citation keys connect the Org text to these records; a DOI identifies the publication itself.

COMPLETE ORG SOURCE

```
#+INCLUDE: "assets/references.bib" src bibtex
```

Companion input: assets/references.bib

HTML EXPORT

```
@article{knuth1984,
  author = {Knuth, Donald E.},
  title = {Literate Programming},
  journal = {The Computer Journal},
  year = {1984}, volume = {27}, number = {2},
  pages = {97--111},
  doi = {10.1093/comjnl/27.2.97}
}

@article{schulte2012,
  author = {Schulte, Eric and Davison, Dan and Dye, Thomas and Dominik, Carsten},
  title = {A Multi-Language Computing Environment for Literate Programming and
Reproducible Research},
  journal = {Journal of Statistical Software},
  year = {2012}, volume = {46}, number = {3},
  pages = {1--24},
  doi = {10.18637/jss.v046.i03}
}
```

Generated by native Org. The frame and typography belong to this guide.

Citation key	Local identifier, such as knuth1984; keep it stable.
DOI	Persistent publication identifier; different from an Org CUSTOM_ID.
Metadata sources	Publisher pages for the two article DOIs.

ORG-17-000 · 4 EXAMPLES

A small reproducible report

Combine familiar pieces into a report. These integration examples deliberately change more than one detail.

ORG-17-001 Build the report with visible analysis

ORG-17-002 Publish a result-focused edition

ORG-17-003 Change the data and rebuild

ORG-17-004 Export the report to native PDF

Build the report with visible analysis

KEYS (NO EVIL) C-c C-v b · execute buffer; C-c C-e h h · export HTML
DOOM ALIAS C-c l e h h · export HTML

Keep the question, input data, executable analysis and interpretation together.

COMPLETE ORG SOURCE

```
#+TITLE: Three readings
#+OPTIONS: toc:nil num:nil
* Question
How large is the average reading?
* Data
#+NAME: observations
| 2 |
| 5 |
| 8 |
* Analysis
#+BEGIN_SRC emacs-lisp :var data=observations :results value :exports both
(/ (apply #'+ (mapcar #'car data)) (float (length data)))
#+END_SRC
* Interpretation
The arithmetic mean summarizes these three illustrative readings.
```

HTML AFTER EVALUATION

Question

How large is the average reading?

Data

2
5
8

Analysis

```
(/ (apply #'+ (mapcar #'car data)) (float (length data)))
```

```
5.0
```

Interpretation

The arithmetic mean summarizes these three illustrative readings.

Generated by native Org. The frame and typography belong to this guide.

`:exports both`	Make the computation visible to readers.
`float`	Force floating-point division.
`length`	Count observations.

Publish a result-focused edition

Compare with ORG-17-001 · Build the report with visible analysis

KEYS (NO EVIL) C-c C-v b · execute buffer; C-c C-e h h · export HTML
DOOM ALIAS C-c l e h h · export HTML

Change only the export policy. The calculation stays in the Org source.

COMPLETE ORG SOURCE

```
#+TITLE: Three readings
#+OPTIONS: toc:nil num:nil
* Question
How large is the average reading?
* Data
#+NAME: observations
| 2 |
| 5 |
| 8 |
* Analysis
#+BEGIN_SRC emacs-lisp :var data=observations :results value :exports results
(/ (apply #'+ (mapcar #'car data)) (float (length data)))
#+END_SRC
* Interpretation
The arithmetic mean summarizes these three illustrative readings.
```

HTML AFTER EVALUATION

Question

How large is the average reading?

Data

2
5
8

Analysis

```
5.0
```

Interpretation

The arithmetic mean summarizes these three illustrative readings.

Generated by native Org. The frame and typography belong to this guide.

`:exports results` Hide the implementation in the published edition.

Change the data and rebuild

KEYS (NO EVIL) C-c C-v b · execute buffer; C-c C-e h h · export HTML
DOOM ALIAS C-c l e h h · export HTML

Change the last observation from 8 to 11 and re-run the calculation. The mean changes to 6.0.

COMPLETE ORG SOURCE

```
#+TITLE: Three readings
#+OPTIONS: toc:nil num:nil
* Question
How large is the average reading?
* Data
#+NAME: observations
| 2 |
| 5 |
| 11 |
* Analysis
#+BEGIN_SRC emacs-lisp :var data=observations :results value :exports both
(/ (apply #'+ (mapcar #'car data)) (float (length data)))
#+END_SRC
* Interpretation
The arithmetic mean summarizes these three illustrative readings.
```

HTML AFTER EVALUATION

Question

How large is the average reading?

Data

2
5
11

Analysis

```
(/ (apply #'+ (mapcar #'car data)) (float (length data)))
```

```
6.0
```

Interpretation

The arithmetic mean summarizes these three illustrative readings.

Generated by native Org. The frame and typography belong to this guide.

Named data table

A single changed input flows into the calculation.

Export the report to native PDF

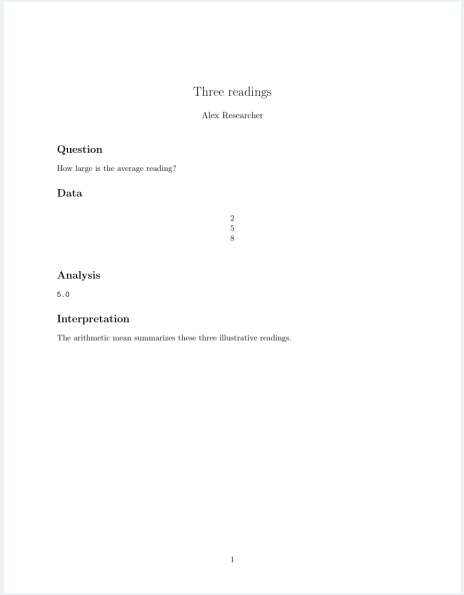
KEYS (NO EVIL) C-c C-v b · execute buffer; C-c C-e l p · export PDF
DOOM ALIAS C-c l e l p · export PDF

Evaluate the same Org report and compile its LaTeX export. The preview shows the actual PDF page.

COMPLETE ORG SOURCE

```
#+TITLE: Three readings
#+AUTHOR: Alex Researcher
#+DATE:
#+LATEX_HEADER: \usepackage[margin=25mm]{geometry}
#+OPTIONS: toc:nil num:nil
* Question
How large is the average reading?
* Data
#+NAME: observations
| 2 |
| 5 |
| 8 |
* Analysis
#+BEGIN_SRC emacs-lisp :var data=observations :results value :exports results
(/ (apply #'+ (mapcar #'car data)) (float (length data)))
#+END_SRC
* Interpretation
The arithmetic mean summarizes these three illustrative readings.
```

NATIVE PDF AFTER EVALUATION



Three readings

Alex Researcher

Question

How large is the average reading?

Data

2
5
8

Analysis

5.0

Interpretation

The arithmetic mean summarizes these three illustrative readings.

1

Page 1

Actual native Org → LaTeX → pdfLaTeX output; preview scaled to fit.

HTML and LaTeX backends	Different renderers consume the same structured source.
Evaluation before export	The build stores fresh results before creating either view.

WORKFLOW REFERENCE

Macros, calls and inline code

Use the smallest mechanism that matches the job. Text macros shape exported prose; Babel executes code. A macro name and a Babel block name live in different mechanisms, even if you happen to give them the same spelling. The examples use unique names within each standalone file.

Mechanism	Syntax	When it runs
Text macro	{{{name}}} or {{{name(a,b)}}}	During document export.
Computed macro	#+MACRO: name (eval ...)	During export; arguments are strings.
Named block call	#+CALL: name(x=7)	C-c C-c on the CALL line; export evaluation depends on :eval.
Inline function call	call_name(x=7)	C-c C-c at the inline call; result stored in a results macro.
Inline source	src_emacs-lisp{(+ 2 5)}	Evaluates the expression directly in the sentence.
Code inclusion	<<name>> with :noweb yes	Expands source text; see the REU chapter.

WORKFLOW REFERENCE

Four choices behind a result

Result collection, representation, insertion and export are independent choices. These controls let one computation serve an editable notebook and a clean publication. The build evaluates the examples explicitly and uses :eval never-export to keep export from running them again.

Question	Header argument	Examples
What is captured?	:results value or output	Return value versus printed output.
How is it represented?	:results table, list, verbatim, raw, html, latex, drawer	Structured values or backend-specific markup.
Where does it go?	:results replace, append, prepend, silent	Update, retain history or leave the buffer unchanged.
What is published?	:exports code, results, both, none	See ORG-11-010 through ORG-11-012 and the FUN inline cases.
Is a file produced?	:results file :file message.txt	Write the result and insert a file link.
Is evaluation permitted?	:eval never-export	Allow explicit evaluation; suppress execution during export.

Within a CALL, inside headers precede the argument list and control the named block. End headers follow the argument list and control insertion of its returned result. Inline calls put the end headers in brackets.

```
#+CALL: demo[:results output]() :results verbatim
Inline: call_double(n=7)[:results raw].
```

WORKFLOW REFERENCE

Publication workflow in Doom

Enable :lang org, :tools biblio and :completion vertico in Doom's module selection, then run doom sync after changing modules. Citar provides selection and actions for bibliographic records. Org-cite stores the citation syntax; the export processor formats the final citations and bibliography. This setup recipe is not a simulated Citar UI or an executed example.

```
;; In your Doom config.el; replace the path with your bibliography.
(after! citar
  (setq citar-bibliography '("~/bib/references.bib")))
(after! org
  (setq org-cite-global-bibliography '("~/bib/references.bib")
        org-cite-insert-processor 'citar
        org-cite-follow-processor 'citar
        org-cite-activate-processor 'citar))
```

Keys / command	Action	Context
C-c C-x @ / C-c l @	org-cite-insert	Insert a citation or edit the one at point.
C-c C-o	org-open-at-point	Follow the citation through the configured processor.
M-x citar-open-entry	Open bibliographic metadata	Requires Citar; choose a reference.
M-x citar-open	Open a resource for a publication	Available PDFs, notes and links depend on your Citar setup.
C-h k, then keys	Inspect active binding	Useful if local Doom customizations change a shortcut.

WORKFLOW REFERENCE

Choose the citation export processor

The executed CIT examples use basic so HTML and native PDF need no extra citation engine. Its intentionally simple formatting includes literal author names and a limited set of bibliographic fields. A publisher-specific style needs CSL or a LaTeX bibliography processor. Citar is a selection frontend, not the export processor.

Processor	Output	Additional requirements
basic	HTML, LaTeX/PDF and other backends	Built into Org; used by all executed citation examples.
csl	HTML, LaTeX/PDF and other backends	citeproc-el, a CSL style file and locale data.
biblatex	LaTeX/PDF	BibLaTeX package and a matching biber or BibTeX build step.
natbib / bibtex	LaTeX/PDF	Matching LaTeX package/style and BibTeX build step.

These are alternative setup recipes, not additional executed cases. Use one CITE_EXPORT setting in a document. Keep your chosen CSL style and bibliography with the project for reproducible output.

```
# Cross-format CSL output, after installing citeproc-el:
#+CITE_EXPORT: csl styles/apa.csl

# LaTeX-only alternative, with biber in the compilation pipeline:
#+CITE_EXPORT: biblatex authoryear,backend=biber

# Shared citation input and bibliography placement:
#+BIBLIOGRAPHY: references.bib
[cite:@knuth1984]
#+PRINT_BIBLIOGRAPHY:
```

COMPANION INPUTS

Shared macros and callable blocks

These files are bundled in assets/ and copied beside each standalone example. They are also embedded in the canonical Org source. Publication records are shown in ORG-16-012.

assets/macros.org

```
#+MACRO: project Open Lab
#+MACRO: quantity $1 $2
```

assets/functions.org

```
#+NAME: scale
#+BEGIN_SRC emacs-lisp :var x=3 factor=2 :results value
(* x factor)
#+END_SRC
```

Source, results and stable references

The Org source and example index are attached to the PDF. The HTML and PDF show the same captured results, with complete source beside each example.

Captured environment: Emacs 31.1 and Org 9.8.7. Native PDF previews show actual exported pages; editor examples show captured buffer text.

Checks: All 208 examples passed their declared expectations. Native PDF examples were compiled twice.

Stable references: An ID such as ORG-01-001 identifies the tour, chapter and example. Chapter numbers have two digits; example numbers have three and start at 001 in each chapter. The 000 entry identifies the chapter. IDs are independent of PDF pages.

Reading the results: Editor results show text rather than your Emacs theme or window chrome. Folded text is shown with ellipses. HTML and LaTeX styling can differ. Capture templates, clock reports, publishing projects and external packages are useful further topics.

QUICK LOOKUP

Command index

Command / action	Example	Purpose
M-RIGHT · org-do-demote	ORG-03-001	Demote one heading
M-S-RIGHT · org-demote-subtree	ORG-03-002	Demote a whole subtree
M-S-LEFT · org-promote-subtree	ORG-03-003	Promote a whole subtree
M-DOWN · org-move-subtree-down	ORG-03-004	Move a subtree down
C-c ^, then a · org-sort-entries	ORG-03-005	Sort sibling headings
C-x n s · org-narrow-to-subtree	ORG-03-006	Focus on one subtree
C-x n s, then C-x n w	ORG-03-007	Widen the buffer again
TAB · org-cycle	ORG-03-008	Fold a subtree
C-c C-c · org-table-align	ORG-05-002	Align the source columns
M-DOWN · org-table-move-row-down	ORG-05-003	Move a data row down
M-S-RIGHT · org-table-insert-column	ORG-05-004	Insert a column
C-u C-c * · org-table-recalculate (whole table)	ORG-05-005	Calculate a whole column
C-u C-c * · org-table-recalculate (whole table)	ORG-05-006	Change the multiplier
C-u C-c * · org-table-recalculate (whole table)	ORG-05-007	Format decimal places
C-u C-c * · org-table-recalculate (whole table)	ORG-05-008	Calculate one specific field
C-u C-c * · org-table-recalculate (whole table)	ORG-05-009	Sum a range

Command index · continued

Command / action	Example	Purpose
C-u C-c * · org-table-recalculate (whole table)	ORG-05-010	Use an Emacs Lisp formula
C-u C-c * · org-table-recalculate (whole table)	ORG-05-011	Refer to a named table
C-u C-c * · org-table-recalculate (whole table)	ORG-06-001	Multiply two columns
C-u C-c * · org-table-recalculate (whole table)	ORG-06-002	Target one cell
C-u C-c * · org-table-recalculate (whole table)	ORG-06-003	Target a rectangular range
C-u C-c * · org-table-recalculate (whole table)	ORG-06-004	Keep a parameter cell fixed
C-u C-c * · org-table-recalculate (whole table)	ORG-06-005	Refer to nearby columns
C-u C-c * · org-table-recalculate (whole table)	ORG-06-006	Put a total in the last row
C-u C-c * · org-table-recalculate (whole table)	ORG-06-007	Use separators as range boundaries
C-u C-c * · org-table-recalculate (whole table)	ORG-06-008	Sum across a row
C-u C-c * · org-table-recalculate (whole table)	ORG-06-009	Calculate a row average
C-u C-c * · org-table-recalculate (whole table)	ORG-06-010	Calculate a weighted mean
C-u C-c * · org-table-recalculate (whole table)	ORG-06-011	Show a share as a percentage
C-u C-c * · org-table-recalculate (whole table)	ORG-06-012	Build a running total
C-u C-c * · org-table-recalculate (whole table)	ORG-06-013	Compare with the previous row
C-u C-c * · org-table-recalculate (whole table)	ORG-06-014	Choose between two values

Command index · continued

Command / action	Example	Purpose
C-u C-c * · org-table-recalculate (whole table)	ORG-06-015	Guard against division by zero
C-u C-c * · org-table-recalculate (whole table)	ORG-06-016	Ignore empty cells in a mean
C-u C-c * · org-table-recalculate (whole table)	ORG-06-017	Treat empty cells as zero
C-u C-c * · org-table-recalculate (whole table)	ORG-06-018	Format fixed decimal places
C-u C-c * · org-table-recalculate (whole table)	ORG-06-019	Use scientific notation
C-u C-c * · org-table-recalculate (whole table)	ORG-06-020	Keep exact fractions
C-u C-c * · org-table-recalculate (whole table)	ORG-06-021	Use degrees in trigonometry
C-u C-c * · org-table-recalculate (whole table)	ORG-06-022	Switch to radians
C-u C-c * · org-table-recalculate (whole table)	ORG-06-023	Name a file-level constant
C-u C-c * · org-table-recalculate (whole table)	ORG-06-024	Give columns meaningful names
C-u C-c * · org-table-recalculate (whole table)	ORG-06-025	Keep a parameter beside the data
C-u C-c * · org-table-recalculate (whole table)	ORG-06-026	Read a heading property
C-u C-c * · org-table-recalculate (whole table)	ORG-06-027	Chain formulas from left to right
C-u C-c * · org-table-recalculate (whole table)	ORG-06-028	See a dependency that needs another pass
C-u C-u C-c * · org-table-iterate	ORG-06-029	Iterate until values stabilize
C-c * · org-table-recalculate (current row)	ORG-06-030	Recalculate only the current row

Command index · continued

Command / action	Example	Purpose
C-u C-c * · org-table-recalculate (whole table)	ORG-06-031	Aggregate a different table
C-u C-c * · org-table-recalculate (whole table)	ORG-06-032	Look up a row by its key
C-u C-c * · org-table-recalculate (whole table)	ORG-06-033	Transform text with Emacs Lisp
C-u C-c * · org-table-recalculate (whole table)	ORG-06-034	Use numeric Lisp arguments
C-u C-c * · org-table-recalculate (whole table)	ORG-06-035	Add durations
C-u C-c * · org-table-recalculate (whole table)	ORG-06-036	Convert a duration to decimal hours
C-u C-c * · org-table-recalculate (whole table)	ORG-06-037	Understand multiple formula lines
C-c C-c on the second #+TBLFM line · org-ctrl-c-ctrl-c	ORG-06-038	Apply a different formula line
C-u C-c * · org-table-recalculate (whole table)	ORG-06-039	Generate row numbers
C-u C-c * · org-table-recalculate (whole table)	ORG-06-040	Make arithmetic grouping explicit
C-c C-t · choose DONE	ORG-07-002	Complete a task
C-c C-t · choose NEXT	ORG-07-003	Define a workflow
C-c C-c · toggle checkbox	ORG-07-006	Check one item
C-c C-c on the cookie	ORG-07-007	Count completed items
C-c C-c on the cookie	ORG-07-008	Show percentage progress

M-x org-agenda-list · one day	ORG-07-009	See one day in the agenda

Command index · continued

Command / action	Example	Purpose
Agenda for 12 October 2026	ORG-07-010	Move the scheduled task
M-x org-todo-list	ORG-07-011	Show all unfinished tasks
Source inspection · no command	ORG-07-012	Add a repeating date
(org-entry-get nil "OWNER")	ORG-08-003	Read a property
(org-entry-get nil "OWNER" t)	ORG-08-004	Inherit a parent property
(org-entry-get nil "OWNER" t)	ORG-08-005	Override an inherited value
C-c C-x p · OWNER · Alex	ORG-08-006	Set a property with a command
org-map-entries · tag match	ORG-08-007	Find headings with a tag
org-map-entries · property match	ORG-08-008	Filter by a property value
C-c C-c · org-babel-lob-execute-maybe	ORG-14-001	Call a block without parameters
C-c C-c · org-babel-lob-execute-maybe	ORG-14-002	Call a function with a default input
C-c C-c · org-babel-lob-execute-maybe	ORG-14-003	Override one function parameter
C-c C-c · org-babel-lob-execute-maybe	ORG-14-004	Pass two named parameters
C-c C-c · org-babel-lob-execute-maybe	ORG-14-005	Keep an omitted parameter at its default
C-c C-c · org-babel-lob-execute-maybe	ORG-14-006	Pass a string to a function
C-c C-c · org-babel-lob-execute-maybe	ORG-14-007	Pass a named table to a function

Command index · continued

Command / action	Example	Purpose
C-c C-c · org-babel-lob-execute-maybe	ORG-14-008	Collect a function return value
C-c C-c · org-babel-lob-execute-maybe	ORG-14-009	Override how the function collects results
C-c C-c · org-babel-lob-execute-maybe	ORG-14-010	Format a call result as a table
C-c C-c · org-babel-lob-execute-maybe	ORG-14-011	Call a function inline without parameters
C-c C-c · org-babel-lob-execute-maybe	ORG-14-012	Pass a parameter in an inline call
C-c C-c · org-babel-lob-execute-maybe	ORG-14-013	Export prose containing an inline result
C-c C-c · org-babel-execute-src-block	ORG-14-014	Evaluate code directly inside a sentence
C-c C-c · org-babel-lob-execute-maybe	ORG-14-015	Call a function from another Org file
C-c C-c · org-babel-execute-src-block	ORG-14-016	Feed one function result into another
C-c C-c · org-babel-lob-execute-maybe	ORG-14-017	Export an inline result as plain text
C-c C-v i · org-babel-lob-ingest; then C-c C-c on CALL	ORG-14-018	Ingest a reusable function library

FURTHER READING

Official documentation

Original demonstrations executed with the installed Org version. The manual is the reference for syntax; inspect your local version with M-x org-version.

Topic	Official manual chapter
First document	https://orgmode.org/manual/Document-Structure.html
Text and lists	https://orgmode.org/manual/Markup-for-Rich-Contents.html
Outline and navigation	https://orgmode.org/manual/Structure-Editing.html
Links and references	https://orgmode.org/manual/Hyperlinks.html
Tables and calculations	https://orgmode.org/manual/Tables.html
Spreadsheets and formulas	https://orgmode.org/manual/The-Spreadsheet.html
Tasks and planning	https://orgmode.org/manual/TODO-Items.html
Tags and properties	https://orgmode.org/manual/Properties-and-Columns.html
Images and blocks	https://orgmode.org/manual/Images.html
Export and layout	https://orgmode.org/manual/Export-Settings.html

Official documentation · continued

Original demonstrations executed with the installed Org version. The manual is the reference for syntax; inspect your local version with M-x org-version.

Topic	Official manual chapter
Source blocks and Babel	https://orgmode.org/manual/Working-with-Source-Code.html
Text macros	https://orgmode.org/manual/Macro-Replacement.html
Reuse and tangling	https://orgmode.org/manual/Extracting-Source-Code.html
Functions and inline calls	https://orgmode.org/manual/Evaluating-Code-Blocks.html
Result formats and files	https://orgmode.org/manual/Results-of-Evaluation.html
Citations and publications	https://orgmode.org/manual/Citation-handling.html
A small reproducible report	https://orgmode.org/manual/Working-with-Source-Code.html

FURTHER READING

Keyboard sources and verification

Standard bindings were checked in Emacs 29.3 / Org 9.6.15. Doom aliases were checked against its module source on 10 October 2026. A full Doom session and your personal configuration were not run by this build. Use C-h k to check the active binding locally.

Reference	Source
Doom non-Evil global bindings	Doom modules · config/default/+emacs-bindings.el
Doom Org localleader bindings	Doom modules · lang/org/config.el · +org-init-keybinds-h
Disabling the Evil module	Doom modules · editor/evil · How do I remove evil?
Table editing	Org manual · Built-in Table Editor
Formula editor and debugger	Org manual · Editing and debugging formulas
Recalculation and iteration	Org manual · Updating the table
Babel commands	Org manual · Key Bindings and Useful Functions
Export menu	Org manual · The Export Dispatcher

Doom prefix defaults assume its default configuration module. Changes to module selection, local bindings or operating-system shortcuts can change the effective keys. The M-x command names remain the reference entry points.

FURTHER READING

Publication and citation sources

Reference	Source
Knuth (1984), Literate Programming	doi:10.1093/comjnl/27.2.97 · The Computer Journal 27(2), 97–111.
Schulte, Davison, Dye and Dominik (2012)	A Multi-Language Computing Environment for Literate Programming and Reproducible Research · Journal of Statistical Software 46(3), 1–24. doi:10.18637/jss.v046.i03
Citar frontend and Doom integration	Official Citar repository and configuration guide
Export processors and bibliography styles	Org manual · Citation export processors
Bibliography placement	Org manual · Bibliography printing

The bundled .bib file contains publication metadata, not full article text. Local citation keys are chosen for the demonstrations. Exported typography and available citation styles depend on the selected processor.

Example and command index

- ORG-01-001 · Start with a paragraph
- ORG-01-002 · Separate two paragraphs
- ORG-01-003 · Add a heading
- ORG-01-004 · Nest a heading
- ORG-01-005 · Add a document title
- ORG-01-006 · Add an author
- ORG-01-007 · Choose an explicit date
- ORG-01-008 · Add a short abstract
- ORG-02-001 · Make a word bold
- ORG-02-002 · Use italic emphasis
- ORG-02-003 · Show inline code
- ORG-02-004 · Show literal text
- ORG-02-005 · Strike out an old statement
- ORG-02-006 · Create a bullet list
- ORG-02-007 · Nest a list item
- ORG-02-008 · Number the items
- ORG-02-009 · Use a description list
- ORG-02-010 · Preserve line breaks in verse
- ORG-03-001 · Demote one heading · M-RIGHT · org-do-demote
- ORG-03-002 · Demote a whole subtree · M-S-RIGHT · org-demote-subtree
- ORG-03-003 · Promote a whole subtree · M-S-LEFT · org-promote-subtree
- ORG-03-004 · Move a subtree down · M-DOWN · org-move-subtree-down
- ORG-03-005 · Sort sibling headings · C-c ^, then a · org-sort-entries
- ORG-03-006 · Focus on one subtree · C-x n s · org-narrow-to-subtree
- ORG-03-007 · Widen the buffer again · C-x n s, then C-x n w
- ORG-03-008 · Fold a subtree · TAB · org-cycle
- ORG-04-001 · Insert a web link
- ORG-04-002 · Give a link readable text
- ORG-04-003 · Link to another heading
- ORG-04-004 · Use a stable custom target
- ORG-04-005 · Link to an inline target
- ORG-04-006 · Add a footnote
- ORG-04-007 · Write an inline footnote
- ORG-04-008 · Link to a local file
- ORG-05-001 · Export a small table
- ORG-05-002 · Align the source columns · C-c C-c · org-table-align
- ORG-05-003 · Move a data row down · M-DOWN · org-table-move-row-down
- ORG-05-004 · Insert a column · M-S-RIGHT · org-table-insert-column
- ORG-05-005 · Calculate a whole column · C-u C-c * · org-table-recalculate (whole table)
- ORG-05-006 · Change the multiplier · C-u C-c * · org-table-recalculate (whole table)
- ORG-05-007 · Format decimal places · C-u C-c * · org-table-recalculate (whole table)
- ORG-05-008 · Calculate one specific field · C-u C-c * · org-table-recalculate (whole table)
- ORG-05-009 · Sum a range · C-u C-c * · org-table-recalculate (whole table)
- ORG-05-010 · Use an Emacs Lisp formula · C-u C-c * · org-table-recalculate (whole table)
- ORG-05-011 · Refer to a named table · C-u C-c * · org-table-recalculate (whole table)
- ORG-05-012 · Add a caption and a reference
- ORG-06-001 · Multiply two columns · C-u C-c * · org-table-recalculate (whole table)
- ORG-06-002 · Target one cell · C-u C-c * · org-table-recalculate (whole table)
- ORG-06-003 · Target a rectangular range · C-u C-c * · org-table-recalculate (whole table)
- ORG-06-004 · Keep a parameter cell fixed · C-u C-c * · org-table-recalculate (whole table)
- ORG-06-005 · Refer to nearby columns · C-u C-c * · org-table-recalculate (whole table)
- ORG-06-006 · Put a total in the last row · C-u C-c * · org-table-recalculate (whole table)

- ORG-06-007 · Use separators as range boundaries · C-u C-c * · org-table-recalculate (whole table)
- ORG-06-008 · Sum across a row · C-u C-c * · org-table-recalculate (whole table)
- ORG-06-009 · Calculate a row average · C-u C-c * · org-table-recalculate (whole table)
- ORG-06-010 · Calculate a weighted mean · C-u C-c * · org-table-recalculate (whole table)
- ORG-06-011 · Show a share as a percentage · C-u C-c * · org-table-recalculate (whole table)
- ORG-06-012 · Build a running total · C-u C-c * · org-table-recalculate (whole table)
- ORG-06-013 · Compare with the previous row · C-u C-c * · org-table-recalculate (whole table)
- ORG-06-014 · Choose between two values · C-u C-c * · org-table-recalculate (whole table)
- ORG-06-015 · Guard against division by zero · C-u C-c * · org-table-recalculate (whole table)
- ORG-06-016 · Ignore empty cells in a mean · C-u C-c * · org-table-recalculate (whole table)
- ORG-06-017 · Treat empty cells as zero · C-u C-c * · org-table-recalculate (whole table)
- ORG-06-018 · Format fixed decimal places · C-u C-c * · org-table-recalculate (whole table)
- ORG-06-019 · Use scientific notation · C-u C-c * · org-table-recalculate (whole table)
- ORG-06-020 · Keep exact fractions · C-u C-c * · org-table-recalculate (whole table)
- ORG-06-021 · Use degrees in trigonometry · C-u C-c * · org-table-recalculate (whole table)
- ORG-06-022 · Switch to radians · C-u C-c * · org-table-recalculate (whole table)
- ORG-06-023 · Name a file-level constant · C-u C-c * · org-table-recalculate (whole table)
- ORG-06-024 · Give columns meaningful names · C-u C-c * · org-table-recalculate (whole table)
- ORG-06-025 · Keep a parameter beside the data · C-u C-c * · org-table-recalculate (whole table)
- ORG-06-026 · Read a heading property · C-u C-c * · org-table-recalculate (whole table)
- ORG-06-027 · Chain formulas from left to right · C-u C-c * · org-table-recalculate (whole table)
- ORG-06-028 · See a dependency that needs another pass · C-u C-c * · org-table-recalculate (whole table)
- ORG-06-029 · Iterate until values stabilize · C-u C-u C-c * · org-table-iterate
- ORG-06-030 · Recalculate only the current row · C-c * · org-table-recalculate (current row)
- ORG-06-031 · Aggregate a different table · C-u C-c * · org-table-recalculate (whole table)
- ORG-06-032 · Look up a row by its key · C-u C-c * · org-table-recalculate (whole table)
- ORG-06-033 · Transform text with Emacs Lisp · C-u C-c * · org-table-recalculate (whole table)
- ORG-06-034 · Use numeric Lisp arguments · C-u C-c * · org-table-recalculate (whole table)
- ORG-06-035 · Add durations · C-u C-c * · org-table-recalculate (whole table)
- ORG-06-036 · Convert a duration to decimal hours · C-u C-c * · org-table-recalculate (whole table)
- ORG-06-037 · Understand multiple formula lines · C-u C-c * · org-table-recalculate (whole table)
- ORG-06-038 · Apply a different formula line · C-c C-c on the second #+TBLFM line · org-ctrl-c-ctrl-c
- ORG-06-039 · Generate row numbers · C-u C-c * · org-table-recalculate (whole table)
- ORG-06-040 · Make arithmetic grouping explicit · C-u C-c * · org-table-recalculate (whole table)
- ORG-07-001 · Mark a heading as a task
- ORG-07-002 · Complete a task · C-c C-t · choose DONE
- ORG-07-003 · Define a workflow · C-c C-t · choose NEXT
- ORG-07-004 · Set a priority
- ORG-07-005 · Start a checklist
- ORG-07-006 · Check one item · C-c C-c · toggle checkbox
- ORG-07-007 · Count completed items · C-c C-c on the cookie
- ORG-07-008 · Show percentage progress · C-c C-c on the cookie
- ORG-07-009 · See one day in the agenda · M-x org-agenda-list · one day
- ORG-07-010 · Move the scheduled task · Agenda for 12 October 2026
- ORG-07-011 · Show all unfinished tasks · M-x org-todo-list
- ORG-07-012 · Add a repeating date · Source inspection · no command
- ORG-08-001 · Tag a heading
- ORG-08-002 · Add a property drawer
- ORG-08-003 · Read a property · (org-entry-get nil "OWNER")
- ORG-08-004 · Inherit a parent property · (org-entry-get nil "OWNER" t)

- ORG-08-005 · Override an inherited value · (org-entry-get nil "OWNER" t)
- ORG-08-006 · Set a property with a command · C-c C-x p · OWNER · Alex
- ORG-08-007 · Find headings with a tag · org-map-entries · tag match
- ORG-08-008 · Filter by a property value · org-map-entries · property match
- ORG-09-001 · Insert an image
- ORG-09-002 · Constrain the HTML image width
- ORG-09-003 · Add an image caption
- ORG-09-004 · Quote a paragraph
- ORG-09-005 · Preserve a literal example
- ORG-09-006 · Create a named special block
- ORG-10-001 · Hide numbers and the contents list
- ORG-10-002 · Number the headings
- ORG-10-003 · Add a shallow contents list
- ORG-10-004 · Include child headings in the contents
- ORG-10-005 · Number only the main headings
- ORG-10-006 · Move the contents list
- ORG-10-007 · Exclude a section
- ORG-10-008 · Select sections explicitly
- ORG-10-009 · Hide TODO keywords in export
- ORG-10-010 · Remove tasks from export
- ORG-10-011 · Require braces for subscripts
- ORG-10-012 · Change the PDF margins
- ORG-10-013 · Use larger PDF type
- ORG-10-014 · Insert an explicit PDF page break
- ORG-11-001 · Display a source block
- ORG-11-002 · Run a Bash block
- ORG-11-003 · Return multiple output lines
- ORG-11-004 · Return an Emacs Lisp value
- ORG-11-005 · Return a table
- ORG-11-006 · Supply an input variable
- ORG-11-007 · Change only the input
- ORG-11-008 · Read a named table as input
- ORG-11-009 · Run Python with a value result
- ORG-11-010 · Export only the code
- ORG-11-011 · Export only the result
- ORG-11-012 · Export code and result
- ORG-11-013 · Append a new result
- ORG-11-014 · Set defaults for a whole file
- ORG-11-015 · Create a plot from a named table
- ORG-11-016 · Change the plot style
- ORG-12-001 · Expand a macro without arguments
- ORG-12-002 · Pass one macro argument
- ORG-12-003 · Pass two macro arguments
- ORG-12-004 · Keep a comma inside an argument
- ORG-12-005 · Return markup from a macro
- ORG-12-006 · Compute a macro replacement
- ORG-12-007 · Reuse document metadata
- ORG-12-008 · Load shared macro definitions
- ORG-13-001 · Define a text macro
- ORG-13-002 · Pass a macro argument
- ORG-13-003 · Include another Org file
- ORG-13-004 · Include a file as source code
- ORG-13-005 · Tangle one Bash script
- ORG-13-006 · Add an executable shebang
- ORG-13-007 · Combine blocks into one file
- ORG-13-008 · Expand a named code fragment
- ORG-13-009 · Expand Noweb only when tangling
- ORG-13-010 · Tangle to two destinations
- ORG-14-001 · Call a block without parameters · C-c C-c · org-babel-lob-execute-maybe
- ORG-14-002 · Call a function with a default input · C-c C-c · org-babel-lob-execute-maybe
- ORG-14-003 · Override one function parameter · C-c C-c · org-babel-lob-execute-maybe
- ORG-14-004 · Pass two named parameters · C-c C-c · org-babel-lob-execute-maybe
- ORG-14-005 · Keep an omitted parameter at its default · C-c C-c · org-babel-lob-execute-maybe
- ORG-14-006 · Pass a string to a function · C-c C-c · org-babel-lob-execute-maybe
- ORG-14-007 · Pass a named table to a function · C-c C-c · org-babel-lob-execute-maybe
- ORG-14-008 · Collect a function return value · C-c C-c · org-babel-lob-execute-maybe

- ORG-14-009 · Override how the function collects results · C-c C-c · org-babel-lob-execute-maybe
- ORG-14-010 · Format a call result as a table · C-c C-c · org-babel-lob-execute-maybe
- ORG-14-011 · Call a function inline without parameters · C-c C-c · org-babel-lob-execute-maybe
- ORG-14-012 · Pass a parameter in an inline call · C-c C-c · org-babel-lob-execute-maybe
- ORG-14-013 · Export prose containing an inline result · C-c C-c · org-babel-lob-execute-maybe
- ORG-14-014 · Evaluate code directly inside a sentence · C-c C-c · org-babel-execute-src-block
- ORG-14-015 · Call a function from another Org file · C-c C-c · org-babel-lob-execute-maybe
- ORG-14-016 · Feed one function result into another · C-c C-c · org-babel-execute-src-block
- ORG-14-017 · Export an inline result as plain text · C-c C-c · org-babel-lob-execute-maybe
- ORG-14-018 · Ingest a reusable function library · C-c C-v i · org-babel-lob-ingest; then C-c C-c on CALL
- ORG-15-001 · Return a plain scalar
- ORG-15-002 · Preserve a string literally
- ORG-15-003 · Render the same string as Org markup
- ORG-15-004 · Return a table
- ORG-15-005 · Return a bullet list
- ORG-15-006 · Wrap output in a results drawer
- ORG-15-007 · Return HTML markup
- ORG-15-008 · Return LaTeX markup
- ORG-15-009 · Write a text file as the result
- ORG-15-010 · Export a downloadable file link
- ORG-15-011 · Replace an earlier result
- ORG-15-012 · Append to earlier results
- ORG-15-013 · Prepend to earlier results
- ORG-15-014 · Evaluate without inserting a result
- ORG-16-001 · Cite a publication and print its reference
- ORG-16-002 · Cite two publications together
- ORG-16-003 · Make the author part of the sentence
- ORG-16-004 · Show author and year separately
- ORG-16-005 · Add a prefix and page locator
- ORG-16-006 · Add context around a citation group
- ORG-16-007 · Switch to numeric references
- ORG-16-008 · Include an uncited publication
- ORG-16-009 · Put a citation in a footnote
- ORG-16-010 · Export author–year references to PDF
- ORG-16-011 · Export numeric references to PDF
- ORG-16-012 · Inspect the publication metadata
- ORG-17-001 · Build the report with visible analysis
- ORG-17-002 · Publish a result-focused edition
- ORG-17-003 · Change the data and rebuild
- ORG-17-004 · Export the report to native PDF

Content: CC0 1.0. Structure and publishing framework: CC BY 4.0 by Lukas C. Bossert. Repository source · Publishing framework source · License scope and third-party notices.