



The Deixis Manual

Notes, Highlights, And Connections

Mark a passage. Share its explanation. Draw the relationship.

DEIXIS 0.2.0 TYPST 0.15.0+

A visual guide with worked examples and a compact API reference.
Detailed recipes and complete record contracts live in docs/guides.

Contents

Annotating With Deixis	3
First Steps	4
Marks, Bodies, And Connections	5
Choose A Placement Policy	6
Drawings, Slides, And Next Steps	7
Worked Examples	8
API At A Glance	23
Shared Note Options	24
Connector Records	25
Setup And Defaults	26
Marks And Identity	27
Bodies And Collections	28
Connections And Drawing	31
Pins And Attachments	34
Scopes And Counters	35
References And Custom Views	36
Rendering Extension Points	37
Low-Level Layout Hooks	39
Integration Checklist	40

Annotating With Deixis

Deixis brings notes, spatial highlights, and visual connections to Typst. It began with presentation annotations and now also serves technical documents, editorial notes, and drawings made from ordinary document content. Start with one marked passage and one explanation. Give the annotation a name when other passages should share it. Add a drawn connection when the relationship needs to be visible on the page.

Keep The Detail Close

Let the main argument stay readable, with explanations beside it.

Our `method`¹ uses one annotation for one idea.

The `result`² has its own explanation and color.

A related passage¹ shares the first number and body. See also ¹.

A small detail³ belongs at the bottom of the page.

¹ A short explanation stays beside the passage it describes.

² Each new annotation receives the next palette color.

³Footnotes use the native page footer.

First Notes: two colored margin annotations, one reused passage, and a native footnote. The editable source is `examples/first-notes.typ`.

The package runtime needs no external drawing library. It includes native routes and arrowheads, while containers, body renderers, and connector painters remain customizable. Public document APIs share one identity and scope model.

This manual describes **Deixis 0.2.0**, requiring **Typst 0.15.0+**. The downloadable `docs/manual.pdf` is built and checked from this source. For rebuild and release commands, see `docs/development/release.md`. A compact reference to all public APIs follows this tour; the task-oriented guides in `docs/guides` cover advanced integration and limits.

First Steps

Install

Import Deixis from Typst Universe:

```
#import "@preview/deixis:0.2.0": *
```

Typst downloads the package automatically. When upgrading from 0.1.1, read [docs/guides/compatibility.md](#) for migration guidance.

Write A Note

```
#import "@preview/deixis:0.2.0": *
#show: deixis-setup-notes
```

```
A useful result#deixis-footnote[A short explanation.]
is easier to read when its details have a place.
```

The wrapper creates a numbered mark and its body; no public ID is needed here. Footnotes use the native page footer: long bodies paginate and look like `std.footnote` entries, even when declared in a heading. Source indentation in a body folds away, while explicit paragraph breaks remain. A long bordered body needs a breakable block container rather than `rect`. See [docs/guides/footnotes.md](#) and [examples/long-footnote.typ](#).

Move An Explanation Into The Margin

Set the page geometry before setup, leaving room for the body:

```
#import "@preview/deixis:0.2.0": *
#set page(margin: (left: 25mm, right: 50mm, y: 25mm))
#show: deixis-setup-notes
```

```
Our #deixis-margin-note(id: <method>, width: 38mm, side: right,
  connector: "right-angle",
)[method][The explanation stays beside the passage it describes.]
keeps the main text short.
```

Two content arguments supply the marked passage and explanation. With one content argument, a wrapper creates a solo numbered mark and uses that argument as the explanation. Omit `connector:` for a note without a drawn connector.

Marks, Bodies, And Connections

An annotation has one original mark, one number, and at most one attached body. The original owns its logical identity and counter; the body owns its explanation, formatting, and placement policy. A connection owns how two endpoints are joined.

Share One Explanation

Use `id:` to create the original; use `of:` to attach a body or reuse the mark:

```
#deixis-mark(id: <detail>)[First passage]
#deixis-margin-note-body(of: <detail>)[One explanation.]
#deixis-mark(of: <detail>)[A related passage]
See also @detail.
```

The original, body, and reused marks may appear in any order. Only the original advances the counter. Another `id:` with the same name is an error. A body-only `id:` creates a standalone explanation without a numbered original. A reused mark adds a return-link origin and keeps the original number, body, and captured mark style. It does not add another drawn connector. Set `backlink: true` on the note/body to show return links. See [docs/guides/identities.md](#) and [docs/guides/shared-notes.md](#).

Add A Connection Deliberately

A margin or inset body draws one connector from its original mark: a route name is enough, as in `connector: "curve"`, and a record adds detail, as in `connector: (route: "curve", heads: (body: "stealth"))`. For a connection from another marked passage, name that occurrence:

```
#deixis-mark(of: <detail>, occurrence: <again>)[Another passage]
#deixis-connect(
  from: (mark: <detail>, occurrence: <again>), to: <detail>,
  connector: (route: "curve", heads: (body: "stealth")),
)
```

Independent connections can target accepted margin/inset body frames, framed inline bodies, live body points, pins, or attached objects. They neither create nor move a body. Connections use their own foreground overlay by default; `connector.layer` can select background independently of the body's layer.

Choose A Placement Policy

Identity lookup and placement are different decisions. Declaring a body before its original does not make every adapter follow that mark to another page.

Body	What Selects Its Position?
Margin, overlay	The original mark or an explicit anchor; packing may spill to later existing pages.
Margin, flow	Immediate placement on the declaration page, with no collision packing or cross-page relocation.
Page footnote	The native footer, allocated at the body declaration. Long bodies paginate.
Inset	The declaration page by default, with optional anchors, offsets, or a placement callback.
Inline	Native flow: right after its passage for the wrapper, or where a <code>-body</code> call appears.
Endnote	An eligible <code>deixis-print-endnotes</code> call.

Immediate/native flow is useful when the body belongs in normal layout or must stay on its declaration page.

Native placement can affect the surrounding paragraph; immediate margin notes do not use overlay packing.

Foreground/background overlays resolve document geometry before painting. Overlay-only declarations add no text-flow space. Overlays can use existing pages, but cannot create new pages. Within an overlay, `z-index` chooses drawing priority; the body and its connectors have independent priorities.

Collected flow lets native layout own a local collection and its pagination: block footnotes collect at the block's end, while endnotes use explicit printers. See `docs/guides/ownership.md`, `rendering.md`, and `blocks.md`.

Configure Appearance

```
#deixis-set(palette: (rgb("3B6D9A"), rgb("387C69"), rgb("9A663F")))
```

Each annotation captures one accent; reuse does not consume another color. Explicit styles win. `deixis-set` applies to subsequent declarations; a `deixis-block` provides a local settings scope, while ordinary Typst brackets do not. Body marker styles resolve at the body declaration; reused marks keep the original mark style. See `docs/guides/configuration.md` and `appearance.md`.

Drawings, Slides, And Next Steps

Route Your Own Objects

The same routing core works without notes or numbering. Register content where it is drawn, then declare each connection independently:

```
#import "@preview/deixis:0.2.0": *
#show: deixis-setup-notes
#show: deixis-routing.with(policy: "avoid")

#grid(columns: (auto, 1fr, auto),
  deixis-attach(id: <source>)[Source], [],
  deixis-attach(id: <target>)[Target],
)
#deixis-connect(from: (node: <source>), to: (node: <target>),
  connector: (heads: (body: "stealth")))
```

`manual` keeps authored routes; `avoid` searches around registered obstacles; `separate` also considers nearby connectors. Objects stay fixed. Pin ports, exact waypoints, and optional cuts/crossing gaps refine the drawing without replacing the connector model. For known coordinates, `deixis-route` returns native geometry without document setup. Start with `examples/routing-in-place.typ` and `docs/guides/document-routing.md`.

Integrate With A Theme

Configure native foreground/background before applying setup. After replacing either later, reapply `deixis-setup-notes`; existing identities and counters persist. Host-owned themes can install explicit overlay hooks. Changing page dimensions or margins alone does not require setup again. See `docs/guides/setup.md`.

Touying reveals retain hidden annotation ownership, but repeated subslides need frame-local identities for explicit IDs/references. Use the maintained examples and `docs/guides/visibility.md` / `touying-identities.md` for that integration.

Read By Task

- **Prepare a final copy:** `review-comments.md`; write `deixis-comment` notes and use `deixis-set(omit: (series: ("review"),))` to keep passages without them. Omitted originals consume no numbers or colors; attached bodies/reused marks follow them. References to omitted IDs report an error. `omit: none` includes subsequent series; a Deixis block restores its parent's policy.
- **Style annotations:** `appearance.md`, `configuration.md`, `containers.md`.
- **Write long documents:** `collected-notes.md`, `blocks.md`, `counters.md`.
- **Customize drawing:** `connectors.md`, `document-routing.md`, `numeric-routing.md`.
- **Build an index:** `queries.md`, `outlines.md`, `note-data.md`.
- **Edit a text or build a glossary:** `editorial-recipes.md`, `review-comments.md`.
- **Replace body output:** `custom-renderers.md`, with live destination/content helpers.
- **Migrate or check limits:** `compatibility.md`.

These guides live in `docs/guides`; `docs/README.md` is the task-oriented index. The compact reference follows this tour. A searchable documentation website can be built locally from the guides and source API catalog; hosting remains undecided.

Worked Examples

Read the source on the left and compare the result on the right. Each example is compiled as its own document, with independent labels, counters and page geometry. The SVG previews show actual output; they do not execute notes inside this manual. Open the named source for its full setup and working PDF links.

The specimens import Deixis and use a muted accent. “No Setup” drawings omit the final show rule:

```
#import "@preview/deixis:0.2.0": *  
#let teal = rgb("32736D")  
#show: deixis-setup-notes
```

Page size/margins and text size are chosen for the small specimen. Margin-note examples reserve a wide right margin. The displayed source is the exact authored body; page geometry and the common preamble are in the named editable file. For Touying, its theme/overlay configuration is in the complete slide example.

The Touying comparison gives both frames the same scale in a wider output column.

The following pages progress from ordinary notes to shared identities and styling, then custom content and drawings. Optional effects are shown explicitly; they are not enabled by ordinary note calls.

A Native Footnote

SETUP REQUIREDEDITABLE SOURCE · [examples/footnote.typ](#)

SOURCE

```
A useful result#deixis-footnote[A short
explanation.]
is easier to read.
```

RENDERED OUTPUT

A useful result¹ is easier to read.

¹A short explanation.

A single content argument creates a solo mark and its explanation. Native flow reserves the footer slot.

Choose An Inline Mark

SETUP REQUIREDEDITABLE SOURCE · [examples/inline-modes.typ](#)

SOURCE

```
#for mode in ("box", "highlight",
"underline", "parentheses",
"text-fill", "none") {
  deixis-mark(inline-mode: mode,
    stroke: teal)[A marked passage]
  linebreak()
}
```

RENDERED OUTPUT

A marked passage

A marked passage

A marked passage

A marked passage

A marked passage

A marked passage

Decoration changes the passage without adding a note body. These anonymous marks have no number.

One Body, Several Passages

SETUP REQUIRED

EDITABLE SOURCE · examples/shared-endnote.typ

SOURCE

```
An idea#deixis-endnote(id: <idea>)[
  One explanation for both uses.
] appears again#deixis-mark(of: <idea>).

A new idea#deixis-endnote[
  A second explanation.
] gets a new number.

#deixis-print-endnotes()
```

RENDERED OUTPUT

An idea¹ appears again¹.

A new idea² gets a new number.

- 1 One explanation for both uses.
- 2 A second explanation.

Reuse keeps the original number and body; the next original advances the counter. The printer chooses the body position.

Declare The Body First

SETUP REQUIRED

EDITABLE SOURCE · examples/forward-margin.typ

SOURCE

```
#deixis-margin-note-body(
  of: <later>, width: 70pt,
  side: right,
)[A forward explanation.]

Our #deixis-mark(id: <later>)[method]
has an explanation.
```

RENDERED OUTPUT

Our method¹ has an explanation.

¹ A forward explanation.

Forward lookup resolves identity. An overlay margin follows the original even when its body was declared earlier.

Cycle Annotation Colors

SETUP REQUIRED

EDITABLE SOURCE · examples/palette.typ

SOURCE

```
#deixis-set(palette: (  
  rgb("3B6D9A"), rgb("387C69"),  
  rgb("9A663F"),  
)  
)  
#for i in range(3) {  
  deixis-margin-note(width: 70pt,  
    side: right)[Passage][  
    Explanation #(i + 1).  
  ]  
  v(20pt)  
}
```

RENDERED OUTPUT

Passage¹¹ Explanation
1.Passage²² Explanation
2.Passage³³ Explanation
3.

Each original takes one accent. Shared marks retain it and explicit component styles can override it.

A Callout With An Arrow

SETUP REQUIRED

EDITABLE SOURCE · examples/inset.typ

SOURCE

```
An #deixis-inset-note(  
  layer: "foreground", width: 90pt,  
  dx: 95pt, dy: 38pt,  
  connector: (route: "curve",  
    heads: (body: "stealth")),  
)[idea][A compact explanation.]  
can point to a callout.
```

RENDERED OUTPUT

An idea¹ can point to a callout.¹ A compact
explanation.

The mark and body use one wrapper. Placement offsets belong to the body; the connector has its own route and head.

Highlight A Formula

SETUP REQUIRED

EDITABLE SOURCE · examples/region.typ

SOURCE

```
#deixis-endnote(
  mark-type: "region", inline: true,
  layer: (mark: "background"),
  fill: teal.lighten(92%),
  stroke: teal,
)[$a^2 + b^2 = c^2$][
  A relation worth explaining.
]

#deixis-print-endnotes()
```

RENDERED OUTPUT

$$a^2 + b^2 = c^2$$

1 A relation worth explaining.

A region wrapper measures the actual content. Mark and body layers can be selected independently.

Attach Region Corners

SETUP REQUIRED

EDITABLE SOURCE · examples/pins.typ

SOURCE

```
#deixis-attach(prefix: none,
  pins: (
    begin: (anchor: top + left),
    end: (anchor: bottom + right),
  ),
  rect(inset: 12pt, stroke: none)[
    $sum_(i=1)^n i = (n(n+1))/2$
  ],
)
#deixis-region-mark(
  pins: (<begin>, <end>),
  padding: 3pt, stroke: teal,
)
```

RENDERED OUTPUT

$$\sum_{i=1}^n i = \frac{n(n+1)}{2}$$

Pins can be attached around an existing object. Their padding is geometric: it adds no space to the object.

Restore Local Settings

SETUP REQUIRED EDITABLE SOURCE · examples/scoped-settings.typ

SOURCE

```
#deixis-set(stroke: blue)
#deixis-mark[Outer Style]

#deixis-block(inset: 5pt)[
  #deixis-set(stroke: orange)
  #deixis-mark[Local Style]
]

#deixis-mark[Outer Style Again]
```

RENDERED OUTPUT

Outer Style

Local Style

Outer Style Again

A Deixis block scopes its settings and numbering. After it ends, the outer defaults resume.

Keep Notes With A Table

SETUP REQUIRED EDITABLE SOURCE · examples/local-notes.typ

SOURCE

```
#deixis-block[
  #figure(caption: [Measurements.])[
    #table(columns: (1fr, 1fr),
      [Sample], [Result],
      [Control], [12.4],
      [Trial], [18.7#deixis-footnote(
        numbering: "a",
      )][Excludes one damaged sample.]],
    )
  ]
]
```

RENDERED OUTPUT

Sample	Result
Control	12.4
Trial	18.7 ^a

Table 1: Measurements.

^aExcludes one damaged sample.

One block owns the whole figure and its caption. Its footnotes collect below the object with independent numbering.

Give A Body A Container

SETUP REQUIREDEDITABLE SOURCE · [examples/container.typ](#)

SOURCE

```
A detail#deixis-endnote(  
  container-func: deixis-rect-container,  
  container-args: (inset: 7pt),  
  stroke: (mark: none, body: teal),  
) [A boxed explanation.]  
  
#deixis-print-endnotes(  
  render-group:  
    deixis-native-render-group,  
)
```

RENDERED OUTPUT

A detail¹

1 A boxed explanation.

Container options stay with the body. A stacked endnote group delegates to its single-note container.

Customize The Body Layout

SETUP REQUIREDEDITABLE SOURCE · [examples/custom-renderer.typ](#)

SOURCE

```
#let card(n, ..args) = block(  
  inset: 8pt, fill: teal.lighten(94%),  
  stroke: (left: teal + 2pt),  
) [  
  #deixis-generate-body-marker(n)  
  #deixis-generate-body-content(n)  
]  
  
#deixis-endnote(render-single: card) [  
  A custom card with native links.  
]  
#deixis-print-endnotes(  
  render-group:  
    deixis-native-render-group,  
)
```

RENDERED OUTPUT

1

1 A custom card with native links.

The marker helper supplies the destination; the content helper retains live spans. Use each once.

Build A Custom Note List

SETUP REQUIRED

EDITABLE SOURCE · examples/catalogue.typ

SOURCE

```
#deixis-endnote(id: <method>,  
  data: (title: [Explain The Method]),  
)[The method needs context.]  
#deixis-mark(of: <method>)  
#deixis-print-endnotes()  
  
#deixis-note-outline(  
  title: [Review Topics],  
  summary: n => n.data.title,  
)
```

RENDERED OUTPUT

11

1 The method needs context.

Review Topics

1 Explain The Method

1

Author data stays separate from identity and layout. Query records can supply custom catalogues without replaying body content.

Connect Ordinary Content

SETUP REQUIRED

EDITABLE SOURCE · examples/attached-nodes.typ

SOURCE

```
#grid(columns: (1fr, 1fr),  
  gutter: 45pt,  
  deixis-attach(id: <a>,  
    rect(inset: 5pt)[Source]),  
  deixis-attach(id: <b>,  
    rect(inset: 5pt)[Target]),  
)  
#deixis-connect(  
  from: (node: <a>),  
  to: (node: <b>),  
  connector: (heads: "stealth"),  
)
```

RENDERED OUTPUT



Node IDs register bounds where objects are rendered. No detached list of object positions is needed.

Choose A Native Arrowhead

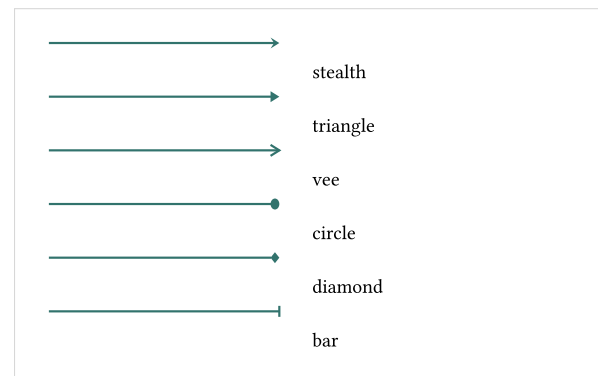
NO SETUP

EDITABLE SOURCE · [examples/head-shapes.typ](#)

SOURCE

```
#for shape in ("stealth", "triangle",
  "vee", "circle", "diamond", "bar") {
  context {
    let path = deixis-route(
      from: (0pt, 0pt),
      to: (105pt, 0pt))
    let paint = deixis-prepare-connector(
      path, stroke: teal + 1pt,
      heads: (body: shape))
    box(width: 120pt, height: 16pt,
      deixis-render-connector(paint))
    text(size: 9pt, shape)
    linebreak()
  }
}
```

RENDERED OUTPUT



Head size, fill, stroke and trimming are configurable. Numeric geometry and its painter work without note setup.

Cut A Curve Without Moving It

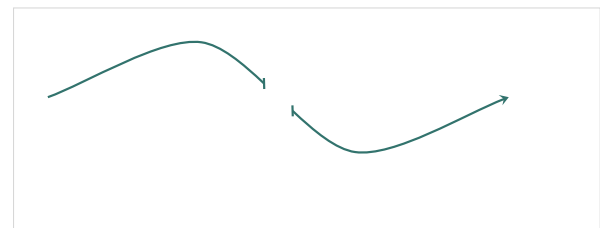
NO SETUP

EDITABLE SOURCE · [examples/curve-cuts.typ](#)

SOURCE

```
#context {
  let path = deixis-route(
    from: (0pt, 25pt),
    to: (210pt, 25pt),
    route: "curve",
    via: ((70pt, 0pt), (140pt, 50pt)))
  let paint = deixis-prepare-connector(
    path, stroke: teal + 1pt,
    heads: (body: "stealth"),
    cuts: (at: 50%, length: 18pt),
    gap-marks: "slash")
  box(width: 220pt, height: 70pt,
    place(top + left, dy: 25pt,
      deixis-render-connector(paint)))
}
```

RENDERED OUTPUT



Cuts hide intervals on the completed curve. Optional indicators are independent; the remaining pieces retain the original geometry.

Route Around An Obstacle

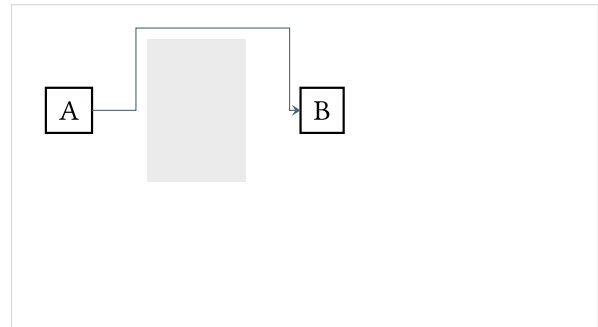
SETUP REQUIRED

EDITABLE SOURCE · [examples/avoidance.typ](#)

SOURCE

```
#show: deixis-routing.with(
  policy: "avoid", clearance: 5pt)
#let node(id, body) = deixis-attach(
  id: id, rect(inset: 6pt, body))
#grid(columns: (auto, auto, auto),
  gutter: 25pt, align: horizon,
  node(<a>)[A],
  deixis-attach(behavior: "avoid",
    rect(width: 45pt, height: 65pt,
      fill: luma(235), stroke: none)),
  node(<b>)[B],
)
#deixis-connect(from: (node: <a>),
  to: (node: <b>),
  connector: (heads: "stealth"))
```

RENDERED OUTPUT



The wrapper captures a policy for individually declared connections. Registered nodes stay fixed; the sharp orthogonal route avoids the middle rectangle.

Continue On Another Page

SETUP REQUIRED

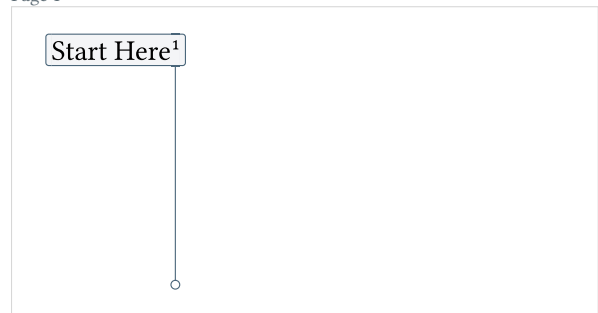
EDITABLE SOURCE · [examples/page-portals.typ](#)

SOURCE

```
#deixis-mark(id: <across>)[Start Here]
#pagebreak()
#deixis-inset-note-body(of: <across>,
  layer: "foreground", width: 95pt,
  dx: 95pt, dy: 40pt,
  connector: (route: "orthogonal",
    continuation: "portal",
    heads: (body: "stealth")),
)[The explanation is on page two.]
Destination Page.
```

RENDERED OUTPUT

Page 1



Page 2

Destination Page.

¹ The explanation is on page two.

A portal marks each accepted page boundary. The two fragments use each page's geometry; no line is drawn across the intervening document.

Follow Endnotes In The Header

SETUP REQUIRED

 EDITABLE SOURCE · [examples/adaptive-header.typ](#)

SOURCE

```
#let header = context {
  let notes = deixis-query-notes(
    include-types: "endnote",
    page: here().page()
  )
  if notes.len() > 0 {
    let pages = notes.map(n =>
      n.marks.map(m =>
        m.dest.page()
      ).flatten().dedup().sorted()
    )
    text(size: 9pt)[
      Notes For Pages
      #pages.map(str).join([, ])
    ]
  }
}

#set page(header: header)

Our #deixis-endnote(
  id: <method>,
)[
  method
][An explanation of the method.]
needs context.
#pagebreak()
Use the #deixis-mark(
  of: <method>,
)[
  method again
].
#deixis-print-endnotes()
```

RENDERED OUTPUT

Page 1

Our `method1` needs context.

1

Page 2

Notes For Pages 1, 2

Use the `method again1`.

1 An explanation of the method.

2

The header reads complete body spans on its physical page, then gathers all original and reused mark pages. Querying never renders an explanation again. These source page numbers are physical; native page-label formatting is a separate step.

Reveal An Annotation With Its Passage

SETUP REQUIRED

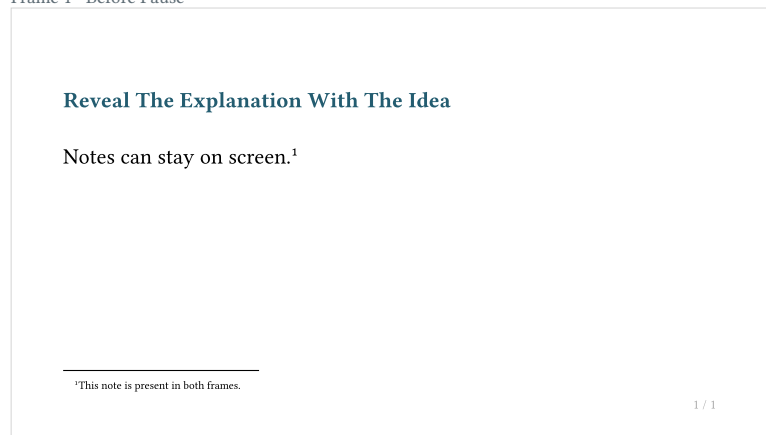
 EDITABLE SOURCE · [../examples/touying-reveals.typ](#)

SOURCE

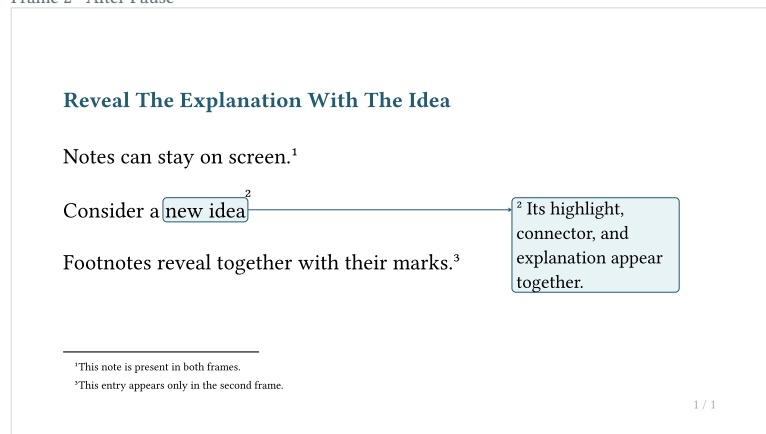
```
#pause
#v(0.6em)
Consider a #deixis-inset-note(
  mark-type: "region",
  inline: true,
  layer: (
    mark: "background",
    body: "foreground",
  ),
  width: 190pt,
  dx: 300pt, dy: 40pt,
  fill: rgb("e8f3f6"),
  stroke: rgb("235c70"),
  connector: (
    route: "curve",
    ports: (
      mark: "right",
      body: "left",
    ),
    heads: "triangle",
  ),
)[new idea][
  Its highlight, connector,
  and explanation appear
  together.
]
```

RENDERED OUTPUT

Frame 1 · Before Pause



Frame 2 · After Pause



Touying renders two subslides. The second reveals the region, connector and callout together; hidden declarations retain their semantic ownership.

Separate Readings From Commentary

SETUP REQUIRED

EDITABLE SOURCE · examples/editorial-apparatus.typ

SOURCE

```
#let apparatus(n, ..args) =
block[
  #deixis-generate-body-marker(n)
  #emph(n.data.lemma)#h(0.3em)
  #deixis-generate-body-
content(n)
]
#let source = deixis-
endnote.with(
  series: "sources", numbering:
"a",
  render-single: apparatus)
#let editor = deixis-
endnote.with(
  series: "editorial", numbering:
"1")

The #source(id: <reading>, data:
(lemma: [harbour],))[
  harbour][MS A; MS B reads
"haven".]
was quiet. The #deixis-mark(of:
<reading>)[harbour]
returns in the next
stanza.#editor[The repetition is
deliberate.]

== Source Readings
#deixis-print-endnotes(series:
"sources",
  render-group: deixis-native-
render-group)
== Editorial Notes
#deixis-print-endnotes(series:
"editorial")
```

RENDERED OUTPUT

The harbour^a was quiet. The harbour^a returns in the next stanza.¹

Source Readings

a *harbour* MS A; MS B reads “haven”.

Editorial Notes

1 The repetition is deliberate.

Independent series give source readings and editorial comments their own numbering. A custom renderer adds a lemma; reused passages share one reading.

Find Open Review Comments

SETUP REQUIRED

EDITABLE SOURCE · [examples/review-index.typ](#)

SOURCE

```
#let comment = deixis-
endnote.with(series: "review")
The result is #comment(data: (
  author: "Ada", status: "open",
  title: [Check Scope],
))[general][Does the argument
cover the boundary case?].
The #comment(data: (
  author: "Lin", status:
"resolved", title: [Name The
Source],
))[source][The citation has been
added.] is now identified.
#deixis-print-endnotes(series:
"review")

== Open Comments By Ada
#context {
  let pending = deixis-query-
notes(series: "review")
  .filter(n => n.data.author ==
"Ada"
    and n.data.status ==
"open")
  for n in pending {
    block(link(n.dest)[#n.marker
#n.data.title])
  }
}
```

RENDERED OUTPUT

The result is [\[general\]¹](#). The [\[source\]²](#) is now identified.

- 1 Does the argument cover the boundary case?
- 2 The citation has been added.

Open Comments By Ada

- 1 Check Scope

Author and status are ordinary annotation data. A filtered linked view selects open comments without removing resolved notes from the draft.

Link Terms To Their First Use

SETUP REQUIRED

 EDITABLE SOURCE · [examples/glossary.typ](#)

SOURCE

```
#deixis-mark(id: <lemma>, series:
"GLOSSARY",
  marker: none, data: (term:
"Lemma",))[*Lemma*]
means the dictionary form of a
word.
#deixis-endnote-body(of: <lemma>)
[A dictionary headword.]
Later, #context {
  let n = deixis-query-
notes(series: "GLOSSARY")
    .find(n => n.id == <lemma>)
  link(n.dest)[lemma]
} links to its first use.

#deixis-mark(id: <token>, series:
"GLOSSARY",
  marker: none, data: (term:
"Token",))[*Token*]
means one occurrence of a word.
#deixis-endnote-body(of: <token>)
[One occurrence in a text.]

== Glossary
#context {
  let terms = deixis-query-
notes(series: "GLOSSARY")
    .sorted(key: n =>
n.data.term)
  for n in terms {
    block[#strong(link(n.dest,
n.data.term)): #n.summary]
  }
}
```

RENDERED OUTPUT

Lemma means the dictionary form of a word. Later, lemma links to its first use.

Token means one occurrence of a word.

Glossary

Lemma: A dictionary headword.

Token: One occurrence in a text.

Explicit originals choose the first use. A sorted catalogue displays safe definition summaries without printing or executing their endnote bodies.

API At A Glance

Start with a note wrapper. Reach for marks/bodies when authoring positions differ, and independent connections when one explanation needs more edges. The entries below cover every public export, with shared/preset functions grouped together.

SETUP REQUIRED

Uses the document annotation system; apply `deixis-setup-notes` first.

NO SETUP

Works with supplied content or numeric geometry, without note setup.

CONDITIONAL

Numeric/supplied data needs no setup; document lookup or live annotation behavior does. See the entry.

INSTALLS SETUP

Installs the document machinery.

EXTENSION HOOK

Low-level callback or integration API; normal note calls usually suffice.

Badges classify **setup**, not Typst context. Queries and document geometry need context; relative styles and custom painting can need it even on a “No Setup” path. Direct render helpers do not initialize or register a note.

This is a compact reference: displayed parameters/defaults come from each entry’s first function in runtime source.

Long option sets are shared below; task guides explain records, edge cases and recipes. A listed auto is an argument default, not necessarily the final value: factory settings and the current Deixis scope can resolve it.

```
deixis-absolute-place
deixis-aligned-render-group
deixis-arrowhead
deixis-block
deixis-comment
deixis-connect
deixis-endnote
deixis-footnote
deixis-generate-backlinks
deixis-generate-body-marker
deixis-grid-render-group
deixis-inline-note
deixis-inset-note
deixis-margin-note
deixis-margin-note-body
deixis-margin-note-flow-layout
deixis-minipage
deixis-native-render-single
deixis-note-outline
deixis-phantom-mark
deixis-place-anchored
deixis-plain-container
deixis-print-endnotes
deixis-rect-container
deixis-ref
deixis-render-connector
deixis-route
deixis-routing
deixis-set
deixis-sidenote
deixis-update-note-counter
```

```
deixis-alert-container
deixis-aligned-render-single
deixis-attach
deixis-chicago-render-single
deixis-configure
deixis-ellipse-region
deixis-endnote-body
deixis-footnote-body
deixis-generate-body-content
deixis-generate-body-meta
deixis-inline-mark
deixis-inline-note-body
deixis-inset-note-body
deixis-margin-note-adaptive-layout
deixis-margin-note-exact-layout
deixis-mark
deixis-native-render-group
deixis-note-counter
deixis-page-overlay
deixis-pin
deixis-plain
deixis-prepare-connector
deixis-query-notes
deixis-rect-region
deixis-region-mark
deixis-render-head
deixis-route-group
deixis-routing-scene
deixis-setup-notes
deixis-sidenote-body
short
```

Shared Note Options

Wrappers take [body] or [passage][body]; body-only functions take [body]. Options apply only where accepted by that adapter. Inherited options stay omitted in the compact entries rather than repeated for every note type.

id	Original/standalone identity; omit for private wrapper identity.
of	Attach a body or reuse a mark; exclusive with id.
data	Opaque author fields on creation; inherited by attachments/reuse.
target	Rendering/counter scope. Attachments inherit their original instance; target: 0 deliberately selects the caller.
series	Counter/collection stream. Attachments inherit it; original wrappers choose their adapter's stream.
marker	Automatic numbering, a literal marker, or none to hide it.
numbering	Pattern/callback, or (restart: "page", pattern: "1") for original-page display ordinals. Assigned counters stay unchanged; reuse keeps the original marker.
marker-style	Callback for marker content; may select mark/body components.
stroke / fill / radius	Shared visual styles; component maps distinguish mark, body and connector.
layer / z-index	Placement/paint layer and within-layer priority; connector layer stays independent.
backlink	Return links from the body: true, false, always, never/none, or multiple.
container-func / container-args	Wrap the rendered body; native/custom options stay in container-args.
render-single / render-group	Replace body/group formatting; preserve live destination/content helpers.

Settings And Defaults

deixis-configure is an identical alias of deixis-set. For short calls, import the short module, then import short: *: use configure, footnote and note-block. Native block remains available. Other aliases omit deixis-; short as dx supports qualified calls such as dx. footnote.

deixis-set accepts palette, numbering, marker-style, body-style, backlink, marker-position, inline-mode, stroke, fill, radius, connector, separator, clearance, gap, marker-gap, indent, margin-layout, min-margin-width, side-strategy, mark-align, mark-align-strictness, spillover, overflow, region-shape, layer, z-index, render-single, render-group and container-func. Every argument defaults to auto, leaving current settings intact. Type/component dictionaries keep the API small; see configuration.md.

Factory marks use muted blue-gray and a faint tint. Body marker gaps are 0.05em for footnotes, 0.25em for margin/inset/inline, and 0.75em for endnotes. Inset and page-footnote bodies default to flow; margin bodies to foreground. Page footnotes use the native footer's typography, separator and gap. Read the placement table in the tour before changing layers. A palette is opt-in.

Connector Records

A margin/inset body accepts connector: (...); `deixis-connect` uses the same record. A route name is shorthand, as in connector: "curve". All fields are optional and merge with `deixis-set(connector: ...)` defaults field by field; enabling a head does not enable a disabled route.

route	Polyline, orthogonal/chamfer, interpolating curve/Catmull-Rom/Hobby, or B-spline influenced by controls. Route records use kind and algorithm-specific settings.
ports	Mark/body sides, side intervals, exact percentages, or a pin-defined custom socket. Outward normals describe departure/arrival direction.
via / segments	Exact visits or route-section records. Use one representation at a time; controls are distinct from exact waypoints.
heads	Independent mark/body heads: a native shape, head specification, callback, or none.
stroke / radius	Single line stroke and turn radius, independent of rectangular node edge/corner styles.
cuts / gap-marks	Hide intervals on finished curve arc length; optional slash/bar indicators decorate surviving gaps.
crossings	Optional automatic gaps on participating lower-priority shafts. It does not enable obstacle avoidance.
continuation / spill	Optional dot/portal/custom page markers and page-local allowed intervals. Margin destinations default to their physical channel, others to content width.
scene	Prepared scene or page callback for explicit interactions outside collected routing, or under manual policy.
render	Custom painter receiving prepared geometry, styles, heads and decorations. Delegate to <code>deixis-render-connector</code> .
layer / z-index	Foreground by default, or background; independent body/edge priorities. No connector flow layer.

Geometry And Failures

Pins and hard visits stay exact. Scene avoidance and collected repair require sharp orthogonal routes. Avoidance checks centerlines; thick strokes/heads may extend outside them. Node behaviors are avoid, ignore, cut or cross; indicators are always opt-in. Independent page fragments may have different continuation positions when page dimensions change. No shaft is drawn on intermediate pages.

The numeric API returns geometry/failure records; inspect failure status before preparing paint. Document APIs diagnose unresolved endpoints and unsupported geometry. Budgets bound repair work; they do not guarantee global optimality. See `connectors.md`, `document-routing.md`, and `routing-scenes.md` for details.

Setup And Defaults

Install setup before document annotations. Configuration can precede it. Reapply setup after replacing a native page foreground/background; identities and counters persist.

Guide: docs/guides/setup.md

deixis-setup-notes

INSTALLS SETUP

Returns content → content

Document show rule: installs registration, references, visibility, validation and optional page overlays.

```
#show: deixis-setup-notes
```

SELECTED PARAMETERS · SOURCE DEFAULTS

```
override-page-footnotes = false      page-overlays = true
foreground = none                  background = none
```

Set page geometry/layers first. Page footnotes default to native flow; override-page-footnotes: true makes the foreground overlay their default. Host themes may install both overlay hooks with page-overlays: false.

deixis-page-overlay

SETUP REQUIRED

Returns content

Host integration hook for one page overlay; does not initialize the annotation registry.

```
deixis-page-overlay(layer: "foreground")
```

SELECTED PARAMETERS · SOURCE DEFAULTS

```
layer = "foreground"
```

Install background too: it records page geometry. See the setup guide for host-owned/scoped composition.

deixis-set

deixis-configure

NO SETUP

Returns content

Updates subsequent defaults in the current Deixis scope; ordinary Typst brackets do not create a settings scope.

```
deixis-set(palette: "okabe-ito")
```

SELECTED PARAMETERS · SOURCE DEFAULTS

```
omit = auto                    palette = auto
numbering = auto              stroke = auto
marker-style = auto          connector = auto
layer = auto                  render-single = auto
```

Top-level `auto` leaves settings unchanged. Type/component maps merge compatible branches; empty maps and branch `auto` have option-specific reset semantics. Resolved `numbering: none` takes no counter value; `deixis-set(numbering: (footnote: "\*\*"))` selects a note family. Palette names are `deixis`, `okabe-ito`, `tol-bright`, `tol-muted` and `tableau`. `omit: (series: ("todo"))` removes selected annotations while preserving passages; `none` includes all series. `deixis-configure` and `short.configure` are identical aliases.

Guide: docs/guides/configuration.md

short

NO SETUP

Returns module

Opt-in module of short aliases. Use configure for settings and note-block for Deixis scopes; native block remains available.

```
#import "@preview/deixis:0.2.0": short
#import short: *
```

Import with `#import "@preview/deixis:0.2.0": short` as `dx` for qualified calls, or import `short` and then `#import short: \*`. Wildcard aliases shadow native `footnote`/`ref` and math `attach`/`arrowhead`; use `std.footnote`, `std.ref`, `std.math.attach` and `std.sym.arrowhead`. `configure` avoids the `set` keyword; `note-block` preserves native `block`.

Guide: docs/guides/getting-started.md

Marks And Identity

Use `id` to create the original counter owner; or reuses it. Names accept nonempty strings or labels. An original, its one body and reused marks may appear in any order. Reuse does not add a drawn connector.

Guide: docs/guides/identities.md

deixis-mark

SETUP REQUIRED Returns content

Unified original/reused mark. Defaults to inline; pins select region mode. A call without `id/of` is anonymous decoration.

```
deixis-mark(id: <detail>)[A passage]
```

SELECTED PARAMETERS · SOURCE DEFAULTS

<code>id</code> = none	<code>of</code> = none
<code>mark-type</code> = auto	<code>occurrence</code> = none
<code>marker</code> = auto	<code>numbering</code> = auto

`mark-type`: "region" highlights a passage or pins; "phantom" is a semantic origin. Set data on creation, not reuse. Resolved numbering: none takes no counter value; an explicit marker may still display. See shared options for styles, targets and region controls.

deixis-inline-mark

SETUP REQUIRED Returns content

Specialized inline form of `deixis-mark`, with an optional marked passage.

```
deixis-inline-mark(of: <detail>)[Another passage]
```

SELECTED PARAMETERS · SOURCE DEFAULTS

<code>inline-mode</code> = auto	<code>marker-position</code> = auto
---------------------------------	-------------------------------------

Box, highlight, underline, parentheses, text-fill and none are available. Reused marks retain the original styles beneath explicit overrides.

Guide: docs/guides/shared-notes.md

deixis-region-mark

SETUP REQUIRED Returns content

Highlights a passage or a pin-defined region, with page-local fragment geometry.

```
deixis-region-mark(inline: true)[$x^2 + y^2$]
```

SELECTED PARAMETERS · SOURCE DEFAULTS

<code>pins</code> = ()	<code>padding</code> = auto
<code>layer</code> = auto	<code>inline</code> = false
<code>region-shape</code> = auto	

Supply passage or pins, never both. A flow region must fit one page. Foreground/background regions may span pages. Region marker numerals remain foreground.

Guide: docs/guides/regions.md

deixis-phantom-mark

SETUP REQUIRED Returns content

Creates/reuses an invisible semantic original; its body may still carry a numbered marker.

```
deixis-phantom-mark(id: <quiet>)
```

SELECTED PARAMETERS · SOURCE DEFAULTS

<code>id</code> = none	<code>of</code> = none
<code>occurrence</code> = none	<code>marker</code> = auto

Phantom reuse accepts no passage or appearance overrides. An anonymous phantom draws nothing and consumes no palette slot.

Bodies And Collections

Wrappers accept [body] or [passage][body]. A -body call takes [body]. Wrapper id creates original + body; -body id creates a standalone body. With of, both forms attach only a body. Margin, inset and endnote bodies capture declaration typography; page footnotes use the native footer.

Guide: docs/guides/ownership.md

deixis-footnote

deixis-footnote-body

SETUP REQUIRED Returns content

Allocates a native footer slot at the body declaration, even when its original mark is on another page.

```
deixis-footnote[An explanation.]
```

SELECTED PARAMETERS · SOURCE DEFAULTS

```
layer = auto                                backlink = auto
container-args = ( : )
```

Native flow (the default) paginates long bodies and uses the footer's typography; separator/gap are native footnote.entry settings. Overlay footnotes reserve space but paint whole units. Local block footnotes collect at the block end. numbering: (restart: "page", pattern: "1") restarts displayed markers on each original page; assigned counts stay stable. HTML uses the native footnote section for every layer. Notes nested in Deixis footnote bodies exceed the HTML layout-pass budget.

Guide: docs/guides/rendering.md

deixis-margin-note

deixis-margin-note-body

SETUP REQUIRED Returns content

Overlay bodies follow the original mark or anchor pin; margin packing may spill onto later existing pages.

```
deixis-margin-note(width: 30mm, side: right)[Passage][Explanation.]
```

SELECTED PARAMETERS · SOURCE DEFAULTS

```
width = 90%                                side = auto
dx = auto                                  dy = 0pt
spillover = auto                           overflow = auto
mark-align = auto                           mark-align-strictness = auto
anchor-pin = none
```

Leave a wide page margin. Flow places exactly on the declaration page with no collision packing. overflow: "error" checks the final frame. Connector shorthand selects the original mark. HTML appends margin bodies to the document's notes list.

Guide: docs/guides/margin-overflow.md

deixis-sidenote

deixis-sidenote-body

SETUP REQUIRED Returns content

Margin-note presets for the outside margin and loose mark alignment.

```
deixis-sidenote[A brief aside.]
```

SELECTED PARAMETERS · SOURCE DEFAULTS

```
side = "outside"                            mark-align-strictness = "loose"
```

Uses margin-note configuration/query type. Explicit arguments override the preset; mark-align-strictness: auto resumes configured strictness.

Guide: docs/guides/configuration.md

deixis-comment

SETUP REQUIRED Returns content

Review-comment preset: a margin note in the "review" series with lettered markers, author/status data and a bold header.

```
deixis-comment(author: "Ada", status: "open")[Passage][Comment.]
```

SELECTED PARAMETERS · SOURCE DEFAULTS

<code>author = none</code>	<code>status = none</code>
<code>series = "review"</code>	<code>numbering = "A"</code>
<code>header = auto</code>	

Remove comments in the series for a final copy with ``deixis-set(omit: (series: ("review",)))``. Other options are `margin-note` options; ``of`` is rejected. Pass ``numbering: auto`` to inherit configured numbering, or ``numbering: none`` for no automatic marker and no counter value.

Guide: docs/guides/review-comments.md

deixis-inset-note

deixis-inset-note-body

SETUP REQUIRED Returns content

Callout whose page defaults to its body declaration; use anchors, offsets or a placement callback for deliberate positioning.

```
deixis-inset-note(dx: 100pt, dy: 40pt)[Passage][Callout.]
```

SELECTED PARAMETERS · SOURCE DEFAULTS

<code>width = auto</code>	<code>layer = auto</code>
<code>placement = none</code>	<code>dx = none</code>
<code>dy = none</code>	<code>anchor = (mark: horizon + right, body: horizon + left)</code>
<code>anchor-pin = none</code>	

Flow is the factory body layer. Overlay declarations are invisible in flow. The connector layer is independent. Custom containers receive `container-args`.

deixis-inline-note

deixis-inline-note-body

SETUP REQUIRED Returns content

An explanation in native flow: the wrapper places it right after its passage as an inline aside; a `-body` call places it where called.

```
deixis-inline-note[term][An inline explanation.]
```

SELECTED PARAMETERS · SOURCE DEFAULTS

<code>inline = true</code>	<code>of = none</code>
<code>breakable = true</code>	<code>container-args = (:)</code>

The wrapper defaults to `inline: true` and keeps the sentence; `inline: false` gives a callout after the line (the `-body` default). Inline decoration bypasses block containers/custom single renderers. `breakable: false` supplies one frame for an explicit body endpoint.

Guide: docs/guides/containers.md

deixis-endnote

deixis-endnote-body

SETUP REQUIRED Returns content

Declares a collected explanation, rendered only by an eligible endnote printer.

```
deixis-endnote[A collected explanation.]
```

SELECTED PARAMETERS · SOURCE DEFAULTS

```
series = auto                                backlink = auto
container-args = ( : )
```

Wrapper series defaults to "endnote"; attachments inherit the original series. The printer controls where it appears. Body settings are retained until printing. Margin, inset, inline and footnotes nested in endnote bodies converge in PDF and HTML within the documented layout limits.

Guide: docs/guides/collected-notes.md

deixis-print-endnotes

SETUP REQUIRED Returns content

Prints unprinted eligible bodies in source order; successive printers do not duplicate output.

```
deixis-print-endnotes()
```

SELECTED PARAMETERS · SOURCE DEFAULTS

```
target = 0                                series = auto
after = none                              before = none
render-group = auto                      container-args = auto
```

Omitted overrides retain body settings. The grid bypasses per-note containers for default aligned singles; a series with a custom single renderer is stacked. Select `deixis-native-render-group` to use containers with default singles. `after`/`before` are exclusive source boundaries.

Guide: docs/guides/collected-notes.md

Connections And Drawing

A body connector is the short path for margin/inset notes. Independent edges select document endpoints without creating or moving them. Connections paint in their own foreground/background layer, below bodies at matching priorities.

Guide: docs/guides/connectors.md

deixis-connect

SETUP REQUIRED Returns content

Draws an independent edge between marks, bodies, attached objects or pins; declarations may precede endpoints.

```
deixis-connect(from: (node: <a>), to: (node: <b>))
```

SELECTED PARAMETERS · SOURCE DEFAULTS

```
from = none                                     to = none
connector = auto
```

Bare from IDs select original marks; bare to IDs select bodies. Use (mark: id, occurrence: name), (body: id, point: "end"), (node: id) or (pin: target) explicitly. Default route is straight, or sharp orthogonal inside deixis-routing.

Guide: docs/guides/independent-connections.md

deixis-routing

SETUP REQUIRED Returns content → content

Show-rule wrapper that captures a policy for individual connections and enabled margin/inset shorthand.

```
#show: deixis-routing.with(policy: "avoid")
```

SELECTED PARAMETERS · SOURCE DEFAULTS

```
policy = "separate"                             nodes = ()
clearance = 4pt                                  spacing = 3pt
rounds = 2                                       max-searches = 64
```

manual keeps native routes; avoid searches independently; separate adds bounded group repair. Search needs sharp orthogonal geometry. Registered objects stay fixed. Nested wrappers form separate groups.

Guide: docs/guides/document-routing.md

deixis-route

NO SETUP Returns route / failure record

Builds source-local line/cubic geometry from finite absolute coordinates, optionally using a prepared scene.

```
deixis-route(from: (0pt, 0pt), to: (100pt, 40pt), route: "curve")
```

SELECTED PARAMETERS · SOURCE DEFAULTS

```
from = none                                     to = none
route = "straight-line"                         via = none
segments = none                                radius = 0pt
scene = none                                    page = none
```

Coordinates use x-right/y-down. origin is the source; segments are relative to it. No measurement or painting. Scene routes can fail: inspect status/detail before preparation. Avoidance requires sharp orthogonal paths.

Guide: docs/guides/numeric-routing.md

deixis-routing-scene

CONDITIONAL Returns scene record

Prepares reusable fixed rectangles/points in caller-defined page planes.

```
deixis-routing-scene(nodes: ((id: <a>, bounds: (left: 0pt, top: 0pt, right: 30pt, bottom: 20pt)),))
```

SELECTED PARAMETERS · SOURCE DEFAULTS

nodes = ()	clearance = 4pt
escape = 4pt	bend-cost = 12pt
bounds = none	pages = ()

Numeric nodes need no setup. Selectors for document marks/bodies/pins need setup + context. Node behavior is avoid, ignore, cut or cross; the latter two do not add visual indicators.

Guide: docs/guides/routing-scenes.md

deixis-route-group

NO SETUP Returns (routes, quality, stats)

Plans up to sixteen explicit sharp orthogonal page fragments together against a prepared scene.

```
deixis-route-group(((from: <a>, to: <b>)), scene: scene)
```

SELECTED PARAMETERS · SOURCE DEFAULTS

edges required	scene = none
spacing = 3pt	weight = 4
max-detour = 25%	rounds = 2
max-searches = 64	max-expansions = 50000

Result routes retain input order. quality has before/after metrics; stats reports bounded work. No registration or painting.

Complete separation/global optimality is not guaranteed. Preparing a document-derived scene still needs setup.

Guide: docs/guides/route-groups.md

deixis-prepare-connector

NO SETUP Returns painter request

Prepares existing geometry for native/custom paint without repeating routing. Head callbacks run here.

```
deixis-prepare-connector(path, heads: (body: "stealth"))
```

SELECTED PARAMETERS · SOURCE DEFAULTS

route required	stroke = black + 1pt
heads = none	cuts = none
gap-marks = none	

Place the painted request at path.origin yourself. Relative styles/custom heads may need context. It consults no document defaults, page layer or automatic crossing group.

Guide: docs/guides/numeric-routing.md

deixis-render-connector

NO SETUP Returns content

Paints a prepared connector request, including shafts, heads and optional decorations.

```
deixis-render-connector(request)
```

SELECTED PARAMETERS · SOURCE DEFAULTS

request required

Numeric painting adds no layout dimensions; reserve the drawing area and place at the source origin. In document callbacks this is the default-painter delegation helper.

Guide: docs/guides/numeric-routing.md

deixis-arrowhead

NO SETUP Returns head specification

Constructs a reusable head specification; stealth is a filled triangular head with an open notch.

```
deixis-arrowhead(shape: "stealth", length: 7pt, width: 5pt)
```

SELECTED PARAMETERS · SOURCE DEFAULTS

shape = "stealth"	length = auto
width = auto	fill = auto
stroke = auto	trim = auto
render = none	

Other native shapes include triangle, vee, circle, diamond and bar. Custom render(request) returns body, trim and bounds in a tip-local +x frame. Omitted dimensions scale with the shaft. Disable endpoint heads with heads: none.

deixis-render-head

NO SETUP **EXTENSION HOOK** Returns content

Paints a prepared head in its local frame. A missing head or zero direction emits nothing.

```
deixis-render-head(request.heads.body, request.route.end, request.route.end-tangent)
```

SELECTED PARAMETERS · SOURCE DEFAULTS

head required	tip required
forward required	

Use prepared heads from the connector request, not the raw specification returned by deixis-arrowhead.

Pins And Attachments

Pin names and attachment IDs accept string/label spellings. They have independent namespaces from note IDs. A pin is a point; an attached routing object registers one live, nonbreaking frame.

Guide: docs/guides/pins.md

deixis-pin

SETUP REQUIRED Returns content

Publishes a geometric anchor at its declaration; padding expands bounds around that anchor.

```
deixis-pin(<anchor>, padding: 0pt)
```

SELECTED PARAMETERS · SOURCE DEFAULTS

<code>name</code> required	<code>scope = none</code>
<code>padding = "text"</code>	<code>extra-data = (:)</code>

Default "text" padding follows local text metrics. Percentages have no pin-size basis. Scope namespaces pins independently of note/block identities.

deixis-attach

SETUP REQUIRED Returns content

Wraps rendered content to attach pins, scan marked tokens, or register a selectable routing node.

```
deixis-attach(id: <card>)[A card]
```

SELECTED PARAMETERS · SOURCE DEFAULTS

<code>id = none</code>	<code>behavior = auto</code>
<code>pins = (:)</code>	<code>scope = none</code>
<code>inline = false</code>	<code>padding = auto</code>
<code>pin-padding = (:)</code>	

id registers a routing frame with avoid behavior by default. Generated node ports are the four side midpoints. pins can define custom outlets. Explicit avoid/ignore/cut/cross also registers anonymous nodes. Pin-only attachments keep their native breakability.

Guide: docs/guides/document-routing.md

Scopes And Counters

Scope names and note IDs are independent. Blocks own rendering instances and counter streams; thread shares a rendering domain, while counter synchronization only shares numbering.

Guide: docs/guides/scopes.md

deixis-block
deixis-minipage

SETUP REQUIRED Returns content

Creates local annotation rendering/settings/counter scope around native content.

```
deixis-block(id: <section>)[Local content.]
```

SELECTED PARAMETERS · SOURCE DEFAULTS

id = auto	thread = none
sync-counters-with = auto	breakable = auto
width = auto	height = auto
inset = auto	

Minipage maps margin to internal inset and fixes outset at 0pt. Named IDs are unique; automatic scopes stay private. Breakable blocks use per-page margin frames and collect local footnotes at the block end. Page-local numbering restarts displayed markers at each original page, including inside multi-page blocks. HTML prints local footnote lists where their block ends.

Guide: docs/guides/blocks.md

deixis-note-counter

SETUP REQUIRED Returns integer

Reads the assigned counter value for a scope and literal series. Use inside context.

```
#context deixis-note-counter()
```

SELECTED PARAMETERS · SOURCE DEFAULTS

target = 0	series = "default"
------------	--------------------

Unused streams start at zero. This is not a total annotation count. Page numbering resets do not reset note counters. Opt-in page-local display markers do not change this assigned value.

Guide: docs/guides/counters.md

deixis-update-note-counter

SETUP REQUIRED Returns content

Sets a stream to an integer or applies current $\rightarrow$ integer. The next numbered original increments that value.

```
deixis-update-note-counter(0)
```

SELECTED PARAMETERS · SOURCE DEFAULTS

value required	target = 0
series = "default"	

Unnumbered originals, bodies, references and reuse never increment. Earlier notes keep assigned numbers. Endnote wrappers use the "endnote" series; select it explicitly for their resets.

Guide: docs/guides/counters.md

References And Custom Views

Queries return safe semantic records and summaries instead of replaying live annotation content. An annotation remains one entry even when several marks share its body.

Guide: docs/guides/queries.md

deixis-ref

SETUP REQUIRED Returns content

References one or more annotations, retaining original marker styling and native fallback for ordinary labels.

`deixis-ref(<detail>)`

SELECTED PARAMETERS · SOURCE DEFAULTS

```
..targets variadic supplement = auto
form = "normal" marker-style = auto
```

@detail uses the native bridge installed by setup. Repeated reference targets deduplicate; compatible numeric ranges can collapse. Explicit marker-style overrides formatting.

Guide: docs/guides/references.md

deixis-query-notes

SETUP REQUIRED Returns array of records

One safe structured record per selected annotation. Page queries follow native body order; the unfiltered catalogue follows source order. Call inside context.

`#context deixis-query-notes(page: here().page())`

SELECTED PARAMETERS · SOURCE DEFAULTS

```
target = auto thread = auto
include-types = auto series = auto
include-unpaired = auto after = none
before = none page = auto
order = auto
```

Page queries include standalone bodies and require complete unique live spans. order: "source" / "body" and include-unpaired override automatic defaults. Each record retains all shared marks, safe summaries and optional body metadata; no live body content is returned.

deixis-note-outline

SETUP REQUIRED Returns content

Ready-made linked list over the annotation catalogue; custom callbacks can replace summaries or presentation.

`deixis-note-outline()`

SELECTED PARAMETERS · SOURCE DEFAULTS

```
title = [List of Notes] include-types = auto
include-unpaired = false group-by = none
summary = auto
```

One entry per annotation, with links to all marked pages. Use query-notes for cards/tables or occurrence indexes. data can supply a custom title/status.

Guide: docs/guides/outlines.md

deixis-plain

SETUP REQUIRED Returns content

Shows repeated content with its marked passages but without creating marks, bodies or connections again.

`deixis-plain(heading.body)`

SELECTED PARAMETERS · SOURCE DEFAULTS

```
body required
```

Native outlines apply it to heading and caption entries automatically. Use it for running headers and custom tables of contents that repeat annotated headings.

Guide: docs/guides/outlines.md

Rendering Extension Points

These hooks consume supplied records or content; they do not create a registered annotation. When used in live document annotations, keep setup active and preserve destination/content helpers. Relative typography and backlinks may require context.

Guide: docs/guides/custom-renderers.md

deixis-native-render-single
deixis-aligned-render-single

CONDITIONAL **EXTENSION HOOK** Returns content

Built-in single-body layouts. Native keeps a natural text flow; aligned accepts a shared marker column.

```
deixis-native-render-single(note, ..args.named())
```

SELECTED PARAMETERS · SOURCE DEFAULTS

data required **inner-width** = auto

Use supplied data unchanged when delegating. Aligned additionally accepts max-marker-width/max-marker-gap. The container and body-style come from the body record.

deixis-chicago-render-single

CONDITIONAL **EXTENSION HOOK** Returns content

Chicago endnotes with a first-line indent and a full-size linked marker followed by a period.

```
deixis-print-endnotes(render-single: deixis-chicago-render-single)
```

SELECTED PARAMETERS · SOURCE DEFAULTS

data required **inner-width** = auto

Defaults to a 0.5in first-line indent unless an explicit indent setting or printer override applies. Preserves inherited paragraph settings, backlinks, styles and containers. Unnumbered notes omit the marker and period. The default group stacks custom single renderers with the configured gap. HTML uses the native semantic renderer without paragraph indentation.

Guide: docs/guides/collected-notes.md

deixis-native-render-group
deixis-aligned-render-group
deixis-grid-render-group

CONDITIONAL **EXTENSION HOOK** Returns content

Stack bodies, align stacked markers, or lay markers/content directly in two grid columns.

```
deixis-native-render-group(notes, ..args.named())
```

SELECTED PARAMETERS · SOURCE DEFAULTS

notes-data required **gap** = auto

Native/aligned groups delegate to each single renderer. Grid paints default aligned records directly; a series with another single renderer is stacked natively. Containers alone do not change the grid path. Empty groups produce no content. Supplied records may contain live document behavior.

Guide: docs/guides/collected-notes.md

deixis-generate-body-meta
deixis-generate-body-marker
deixis-generate-body-content
deixis-generate-backlinks

CONDITIONAL **EXTENSION HOOK** Returns content

Composable helpers for a custom body renderer: one destination, a marker, live content span, and return links.

```
[#deixis-generate-body-marker(note)#deixis-generate-body-content(note)]
```

SELECTED PARAMETERS · SOURCE DEFAULTS

data required

Emit meta OR marker once; marker already includes meta. Content publishes start/end spans and preserves nested palette ownership. Measurement records suppress live metadata. Backlinks read document origins and need setup/context.

deixis-plain-container
 deixis-rect-container
 deixis-alert-container

NO SETUP **EXTENSION HOOK** Returns content

Body wrappers: identity, native rect alias, or a left-bar alert. Container arguments are forwarded separately from shared styles.

```
deixis-alert-container(stroke: teal + 0.5pt)[Body]
```

SELECTED PARAMETERS · SOURCE DEFAULTS

body required **..args** variadic
 `container-func` accepts `"plain"`, `"rect"`, `"alert"`, `"stripe"`, and `"tinted"`, or a custom function. The first three select these exports; `"stripe"`, and `"tinted"`, are named accent presets. Explicit body `stroke`/`fill`/`radius` override preset styles; `container-args` supplies options such as `inset`. `deixis-rect-container` has native `rect` parameters. Alert doubles scalar border thickness on the left; native edge maps remain literal. Built-in containers emit plain text in HTML.
 Guide: docs/guides/containers.md

deixis-rect-region
 deixis-ellipse-region

NO SETUP **EXTENSION HOOK** Returns content

Region-shape callbacks for rectangular or elliptical bounds.

```
deixis-rect-region(width: 40pt, height: 20pt, fill: teal)
```

SELECTED PARAMETERS · SOURCE DEFAULTS

..args variadic
 Shape records provide width, height, fill/stroke and rectangle radius. Region registration/fragment selection belongs to the calling mark, not these painters. Use region marks with stroke: none for a highlight; add region-shape: deixis-ellipse-region for an ellipse.
 Guide: docs/guides/regions.md

Low-Level Layout Hooks

Advanced hooks use supplied measured records and geometry. Most documents should choose note adapters, container/render callbacks, or routing APIs instead. Native layout and document lookup are separate responsibilities.

Guide: docs/guides/rendering.md

deixis-margin-note-exact-layout
deixis-margin-note-adaptive-layout
deixis-margin-note-flow-layout

CONDITIONAL **EXTENSION HOOK** Returns content

Numeric margin adapter hooks: exact ignores collisions; adaptive packs; flow distributes available space.

SELECTED PARAMETERS · SOURCE DEFAULTS

notes required	top-bound required
bottom-bound required	start-x required
text-width required	end-y required
left-space required	right-space required
current-page required	

All share min-margin-width, pushed-notes and extra arguments. Supply accepted page geometry and prepared notes. They paint locally through the layout adapter. These hooks are not the body layer: "flow" switch.

deixis-place-anchored

SETUP REQUIRED **EXTENSION HOOK** Returns content

Immediate placement relative to selected pin bounds on the call's page.

SELECTED PARAMETERS · SOURCE DEFAULTS

body required	dx = 0pt
dy = 0pt	anchor = bottom + right
pin = none	body-size = auto

Pin lookup reads document geometry. Native placement can disturb flow.

Guide: docs/guides/pins.md

deixis-absolute-place

CONDITIONAL **EXTENSION HOOK** Returns content

Immediate placement in physical page coordinates, ignoring margins.

```
deixis-absolute-place(top + left, dx: 30pt, dy: 40pt)[Content]
```

SELECTED PARAMETERS · SOURCE DEFAULTS

..args variadic

Accepts content, or alignment + content; dx/dy default to 0pt. Fixed page dimensions need no setup. An auto page axis needs setup-provided page frames for accepted dimensions. This is placement, not dimension reservation.

Integration Checklist

- Configure native page layers before setup; reapply setup after replacing them. Host-owned overlays still need initialization. Changing dimensions/margins alone does not need another setup call.
- Use flow for long footnotes. Overlays cannot create pages. Immediate margin placement has no collision packing and native placement may disturb paragraphs.
- A body declaration controls formatting and adapter-specific allocation, even with forward identity lookup. Collected bodies must be printed to be endpoints.
- Multi-page region highlights need overlays. Arbitrary native columns/fragments are not automatically discovered. Deep deferred contexts have a finite native layout-pass budget; do not replace missing geometry with guessed bounds.
- In custom renderers, emit destination/marker once and live body content once. Query summaries are safe; replaying raw executable body content is not.
- For Touying, use the frame identity/reveal recipes. For migration from 0.1.1, read docs/guides/compatibility.md.

Read the task-oriented index in docs/README.md. Source examples live in examples/ and docs/examples/; the worked pairs name their editable sources. The examples import @preview/deixis:0.2.0 from Typst Universe; Typst downloads it automatically. Build this source with `python3 scripts/render-manual.py`.

Bug reports and source: <https://github.com/inspiros/typst-deixis>.