Concept

- When to use it
- Title & description
- Scaffold this page
- Component guidance
- Frontmatter
- Writing for AI and agents

import { Render } from "~/components";

A concept page builds the reader's mental model of a topic: what a thing is, why it works the way it does, and where its boundaries lie. It serves anyone getting oriented or already operating the product and filling in the why. The tone is instructional, descriptive, approachable, and supportive.

When to use it

Write a concept page when readers keep needing the same explanation in the middle of other pages, because that recurring detour is the signal the model deserves its own home. It is not:

- **A how-to.** A concept carries no procedural steps or configuration walkthroughs. Code that shows the idea is welcome, code the reader follows along with is not.
- **A reference.** Reference is complete and neutral, whereas a concept is selective and opinionated, so "we recommend" belongs here.
- **An overview.** An overview routes the reader onward, whereas a concept explains. Keep one concept per page.

For the full comparison, refer to Content types.

Title & description

- **Title**: a concise noun phrase naming the concept. Use "About" for a high-level product concept page. Otherwise use a feature name, functionality, or Internet concept such as Health checks or CDN. Do not use "Overview", "Introduction", or "How it works", because they name the genre rather than the subject. As a self-check, a good title still reads naturally with "About" in front of it.
- **Description**: state what the concept is and what it means for the reader's code or choices.

Scaffold this page

<Render file="scaffold-with-recipe" product="style-guide" params={{ type: "concept", recipe: "content-concept" }} />

Component guidance

- **Prose is the primary component.** Short paragraphs, one idea per section: this is the type where writing quality carries the page.
- **Diagrams and illustrative code** fit when they show the model, and a diagram always ships with a text equivalent.
- **Comparison tables** fit where there is a genuine either/or, alongside the boundaries the concept is confused with.
- **What does not fit:** Steps (a concept has no procedural steps or walkthroughs), Tabs (a concept does not vary by platform, and if it does it is two concepts), and Cards.

Frontmatter

```
pcx_content_type: concept
products:
  - product-a
  - product-b
```

For more details, refer to `pcx_content_type`.

Writing for AI and agents

- **Self-contained definition.** Lead with the contract in checkable terms such as "at least once" or "not ordered" rather than reassuring adjectives, because the definition

paragraphs are what an agent retrieves and quotes, so they must stand on their own.

- **Literal payloads.** Keep illustrative code and payloads in fenced blocks with complete, realistic values rather than paraphrase.
- **Declarative boundaries.** Write the boundaries as flat declarative bullets stating what the concept is not, so an agent can lift them without reconstructing the prose.

□□□ #2

□ □□□ □□□ 2026-11-01 04:40:18 UTC

□ □□□ □□□ 2026-11-01 04:40:20 UTC