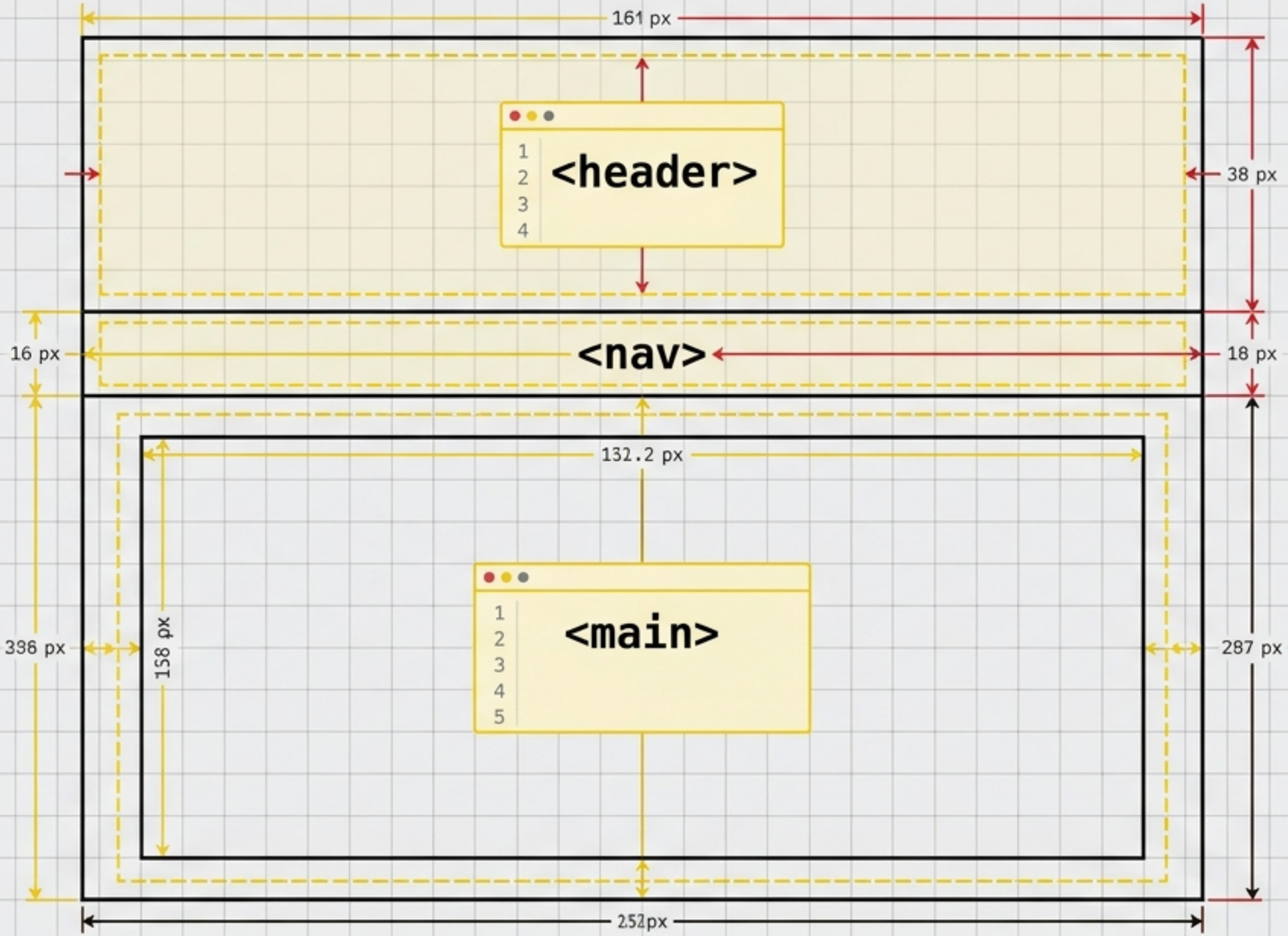
Digital Accessibility: S10E02 — Structure, Semantics and Navigation

Engineering Reference Cards & Implementation Specifications

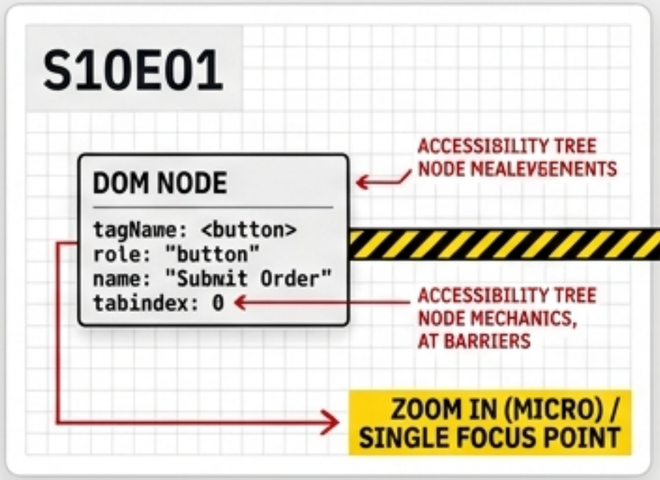


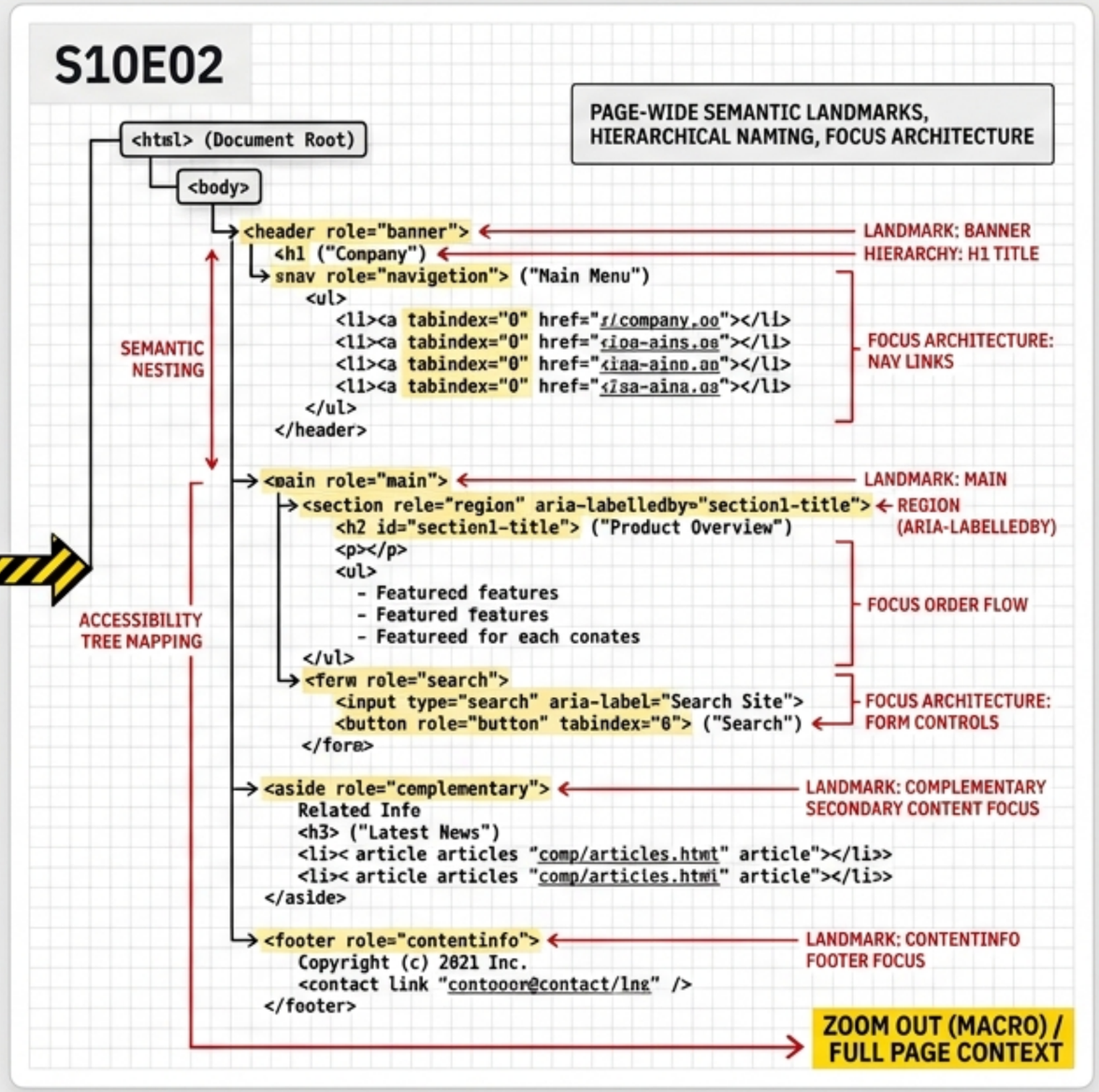
Gemini Notebook

Scaling Accessibility from Node to Page

In S10E01, we dissected Accessibility Tree node mechanics and assistive technology barriers; today, we scale those mechanics up to page-wide semantic landmarks, hierarchical naming, and focus architecture.

Note: Introductory AT overviews are omitted here.
Composite roving tabindex grids will be covered in Lab S10E10.





Gemini Notebook

Ch. 1 — First Rule of ARIA & Landmark Scoping

Escaping Soup: The First Rule of ARIA

The **First Rule of ARIA:** Do not use ARIA if a native HTML element provides the semantics and behavior out-of-the-box.

~~Broken state~~	Fixed state

`<!-- WCAG 1.3.1 Failure -->`
`<div class='btn' tabindex='0'`
`onclick='submitForm()' role='button'`
`>Submit</div>`
→ MANUAL KEYBOARD SUPPORT, ROLE, & BEHAVIOR REQUIRED. HIGH RISK OF ERRORS.
	`<!-- Free Semantics & Keyboard Behavior -->`
	`<button type='submit'>Submit</button>`
	`<a href='/dashboard'>Dashboard</a>`
	→ NATIVE SEMANTICS & KEYBOARD SUPPORT ARE BUILT-IN. ROBUST & ACCESSIBLE.

Gemini Notebook

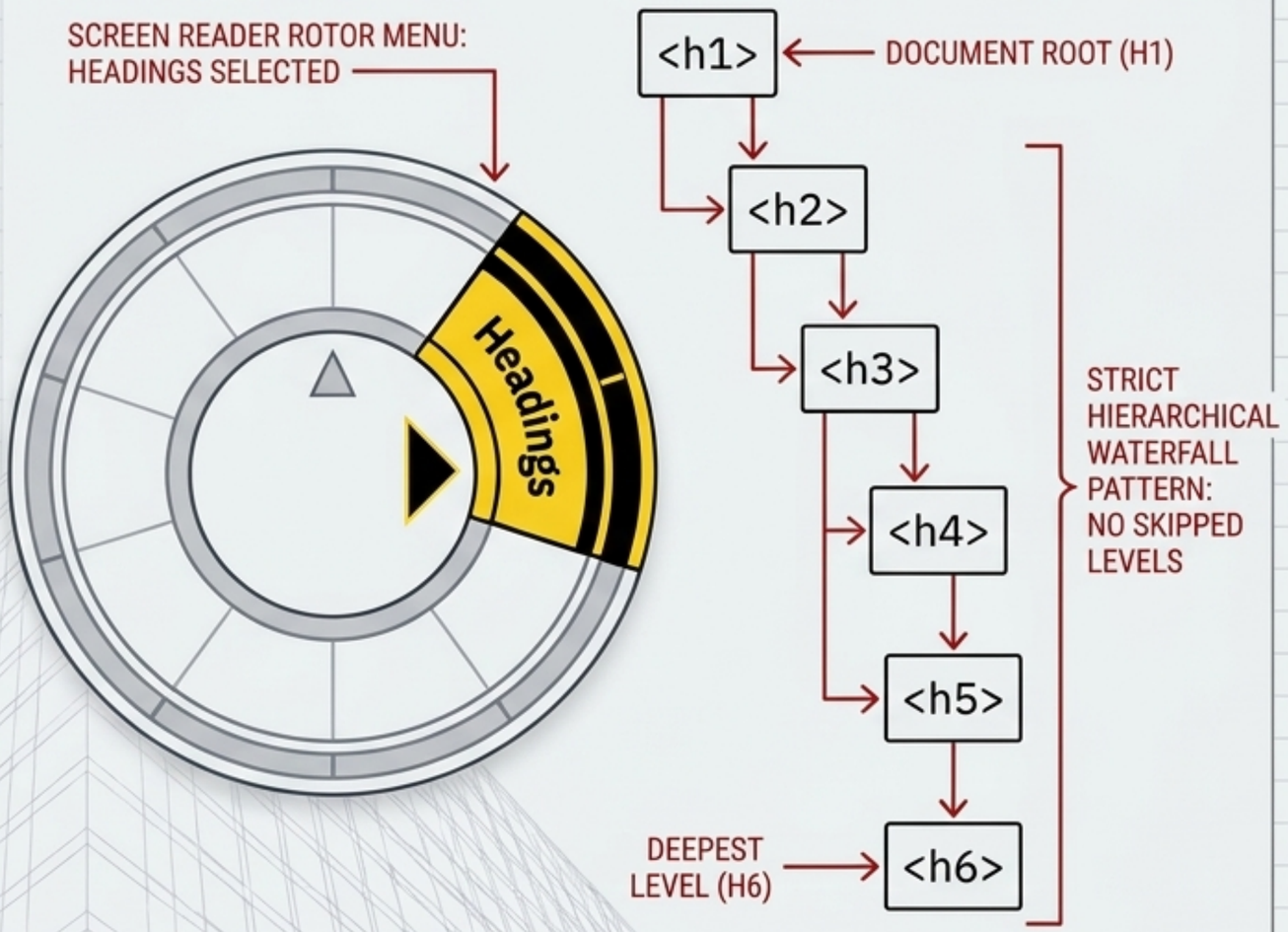
Engineering Blueprint / Redline Spec

Heading Outlines & The Screen Reader Rotor

WCAG 2.4.6 (Headings and Labels): Ensure headings accurately describe the topic or purpose.

Shortcut Navigation: Users rely on H or 1-6 keys to instantly jump across the DOM tree.

The Golden Rule: Never skip heading levels (e.g., jumping from <h2> directly to <h4>). Visual styling should be handled by CSS classes (e.g., `<h2 class='h4-style'>`), not by compromising the structural tree.



Gemini Notebook

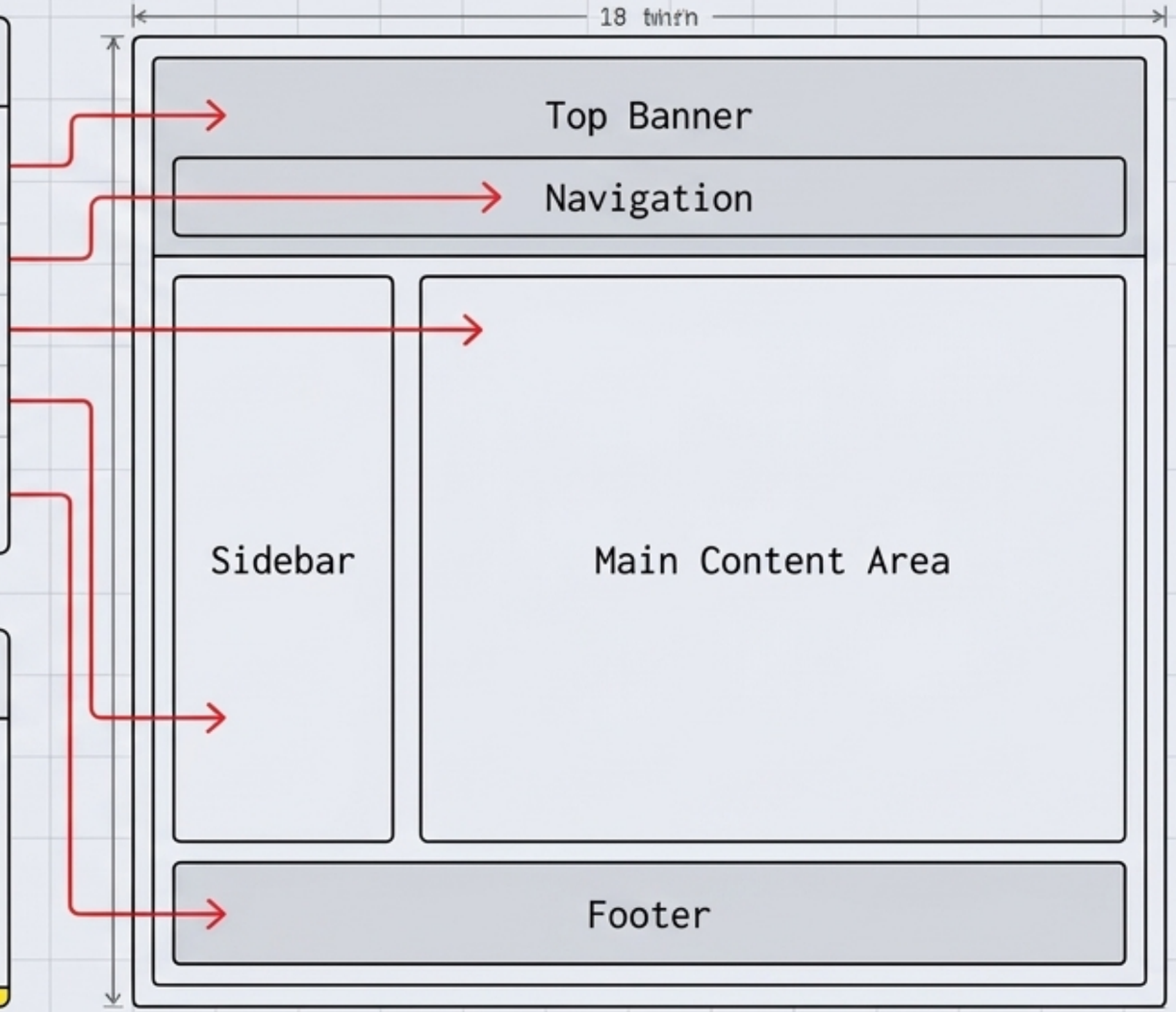
HTML5 Landmark Mapping & Scoping Rules

Mapping Table

<header>	→	banner (only when in the context of the <body> element)
<nav>	→	navigation
<main>	→	main (only one per page)
<aside>	→	complementary
<footer>	→	contentinfo (only when in the context of the <body> element)

Scoping Multiple Regions

```
<nav aria-label='Primary site'>...</nav>
<nav aria-labelledby='toc-heading'>
  <h2 id='toc-heading'>Table of Contents</h2>
  ...
</nav>
```



Gemini Notebook

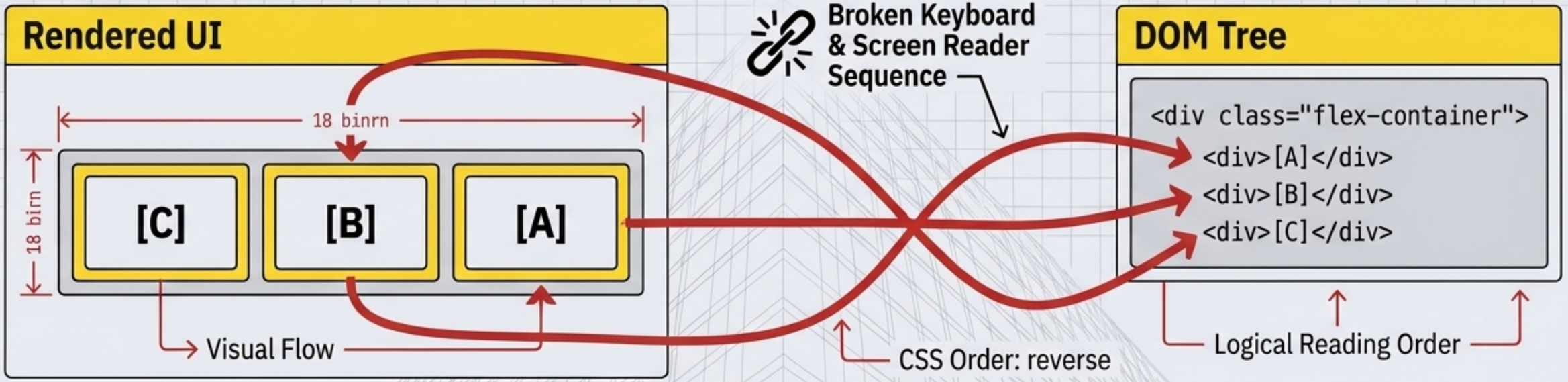
Ch. 2 — DOM Source Order vs. CSS Visual Order

The Disconnect: Visual vs. Logical Reading Order

WCAG 2.4.3 (Focus Order) & WCAG 1.3.2 (Meaningful Sequence)
CSS properties like `flex-direction: row-reverse`, `order: -1`, and placements scramblle screen-reader reading sequence and keyboard tab order, creating a disorienting user experience.

DOM Tree

```
<div class="flex-container">
  ...
</div>
```



Gemini Notebook

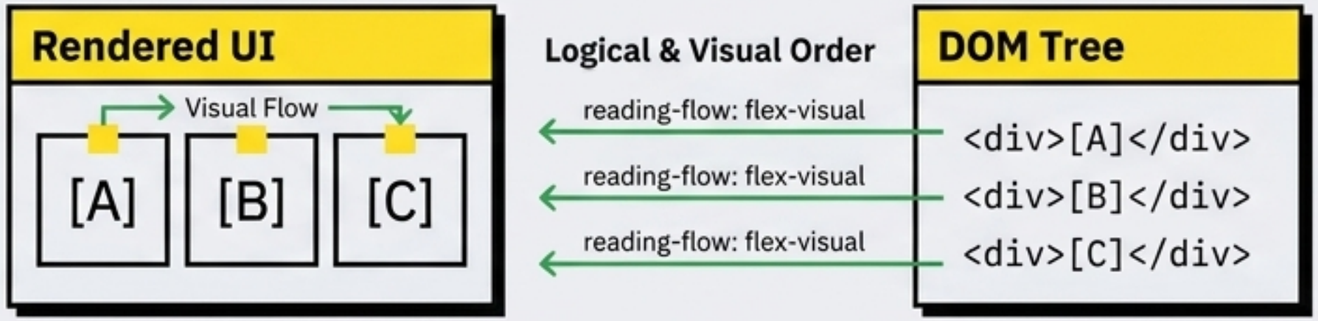
Progressive Alignment: Fixing the Source Order

1. **Fix the HTML First**: The DOM source order must be the single source of truth for text-to-speech and keyboard navigation.
2. **Modern CSS reading-flow:** Use the new CSS Display Level 4 property to align visual rendering with the logical DOM.

```
.container {
  display: flex;
  reading-flow: flex-visual; /* or
      grid-rows / grid-columns */
}
```

A container with reading-flow scopes sequential focus navigation, ignoring positive `tabindex` values for its children to ensure a predictable user journey.

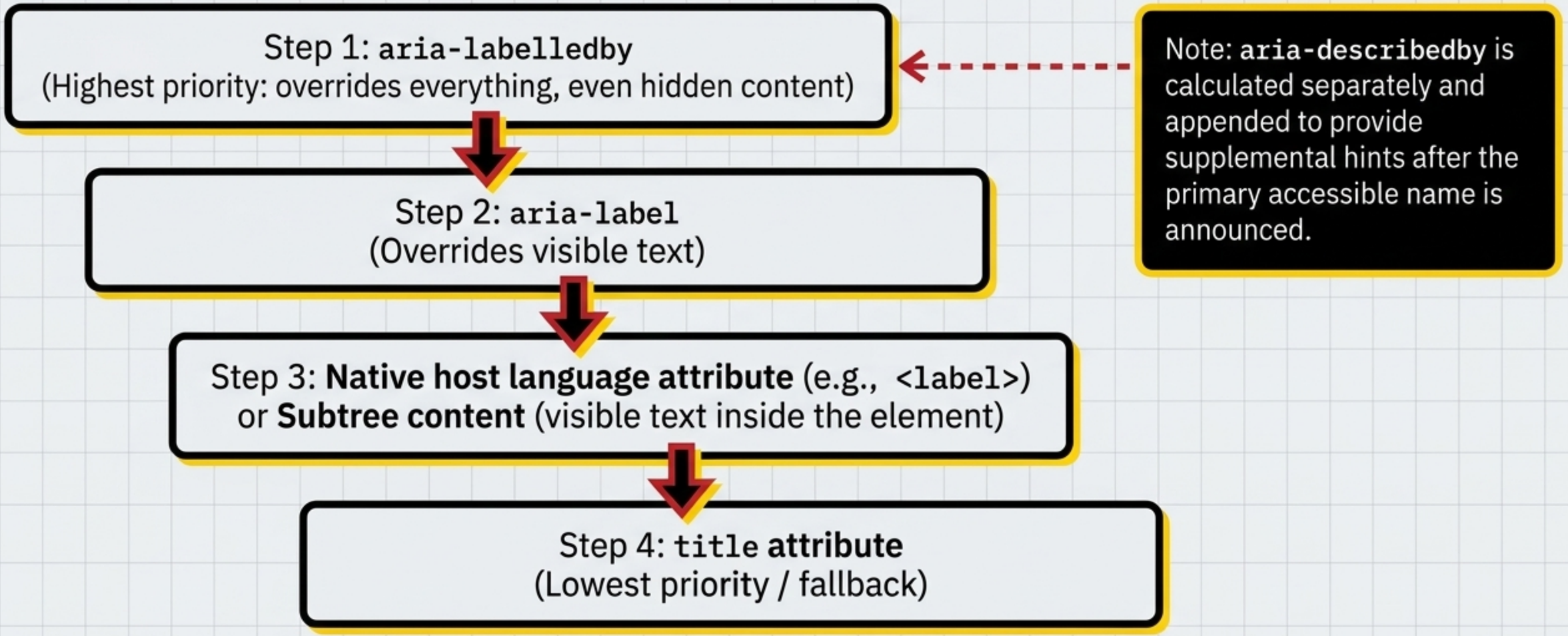
Resolved X-Ray string



Gemini Notebook

Ch. 3 — Accessible Name Computation & Icon Buttons

The accname-1.2 Precedence Cascade



Gemini Notebook

Accessible Icon Buttons: The Standard Pattern

```
<button type='button'>
  <svg aria-hidden='true'
    focusable='false'
    viewBox='0 0 24 24'>
    <!-- SVG paths -->
  </svg>
  <span class='visually-hidden'>
    Close dialog</span>
</button>
```

Prevents double-announcements of the SVG by the screen reader.

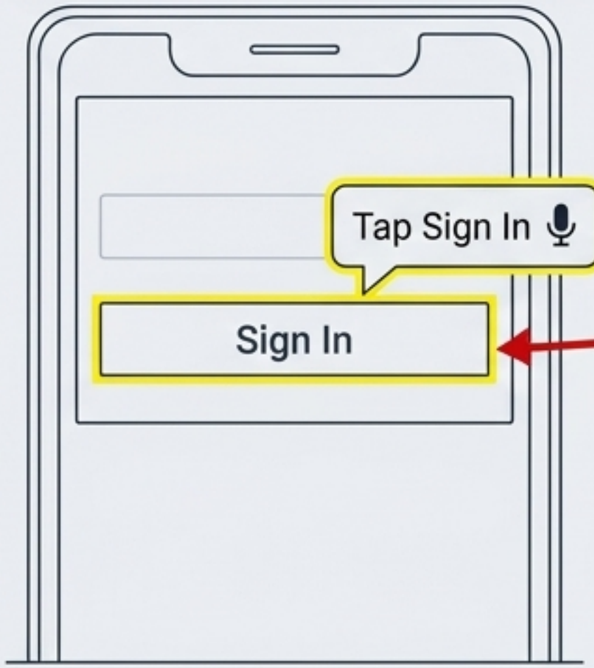
Prevents an Internet Explorer/legacy Edge bug where the SVG itself intercepts keyboard focus.

Placing text directly inside the `<button>` is significantly more robust for translation software than relying purely on `aria-label` for empty buttons.

Gemini Notebook

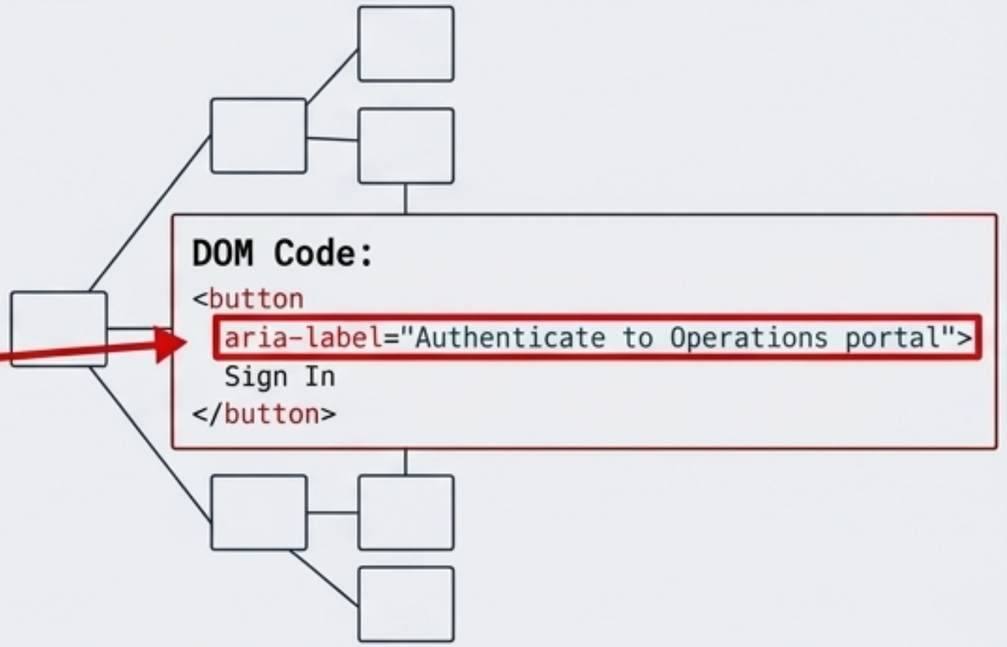
WCAG 2.5.3: Label in Name & Voice Control

Failure Scenario



Visible Button Text:
Sign In

ERROR: VISIBLE TEXT
OVERRIDDEN BY ARIA-LABEL



The Break: Because the accname-1.2 cascade prioritizes **aria-label**, the browser overwrites "Sign In". Voice Control registers the button as "Authenticate to Operations portal". The user's command ("Tap Sign In") fails entirely.

The Fix: Ensure the visible text is always a contiguous part of the computed accessible name.

Gemini Notebook

Ch. 4 — Skip Links, Tabindex & Unobscured Focus

The Strict Discipline of `tabindex`

GOOD

`tabindex='0'`

Use to insert custom interactive elements into the natural keyboard sequence.

Integrates into logical flow

CAUTION

`tabindex='-1'`

Removes elements from natural sequence, but allows them to receive programmatic focus via JavaScript or anchor links (e.g., `<main id='main' tabindex='-1'>`).

Programmatic focus only (e.g., modal opening)

STOP! ANTI-PATTERN

`tabindex='1+' (Values greater than 0)`

⚠ The ultimate anti-pattern. Forces elements to the front of the focus queue, breaking logical flow and overriding the DOM order. Never use this.

Focus order corrupted! Never use.

Gemini Notebook

Engineering Focus-Revealed Skip Links

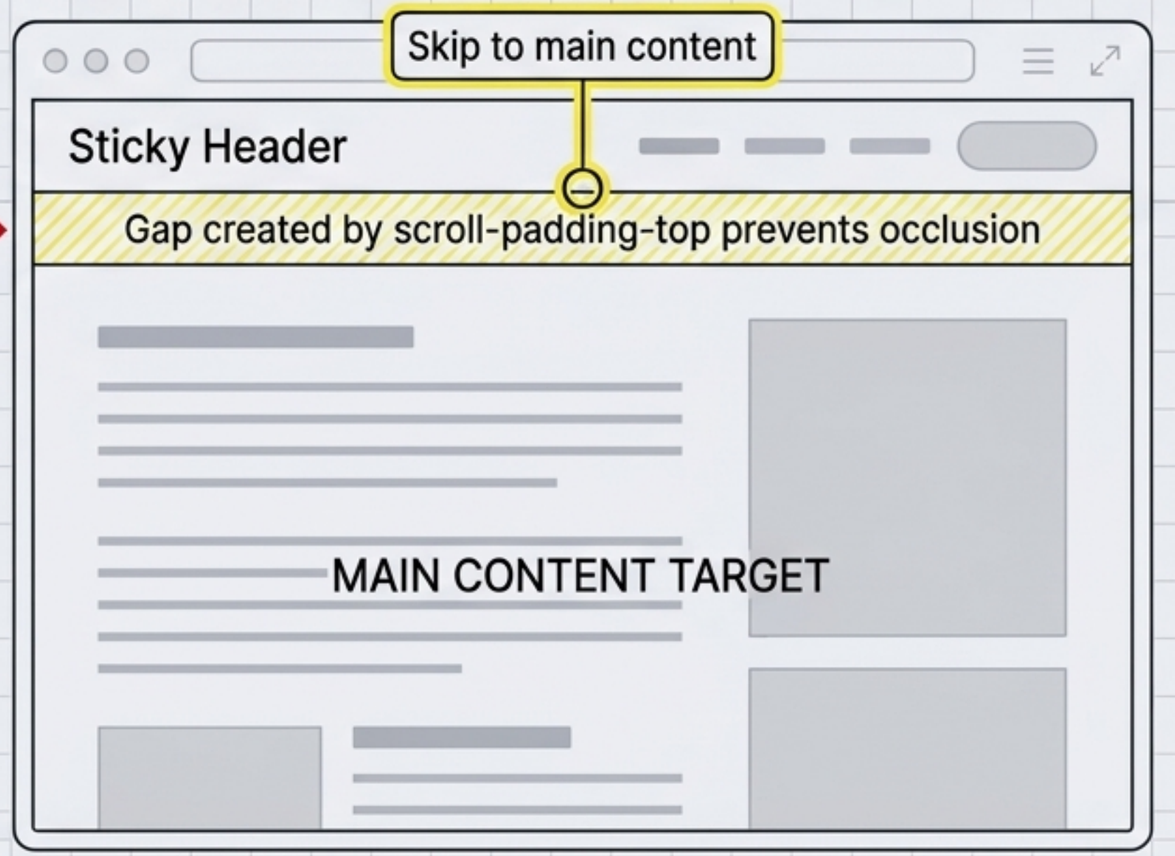
WCAG 2.4.1 (Bypass Blocks): Allows keyboard users to bypass 30+ navigation links.

WCAG 2.4.11 (Focus Not Obscured): When the link jumps to `<main>`, a sticky header will hide the focused content unless accounted for.

```css
.skip-link {
  position: absolute;
  transform: translateY(-100%);
}

.skip-link:focus {
  transform: translateY(0);
}

html {
  /* Prevents sticky header from occluding the target */
  scroll-padding-top: 5rem;
}
```



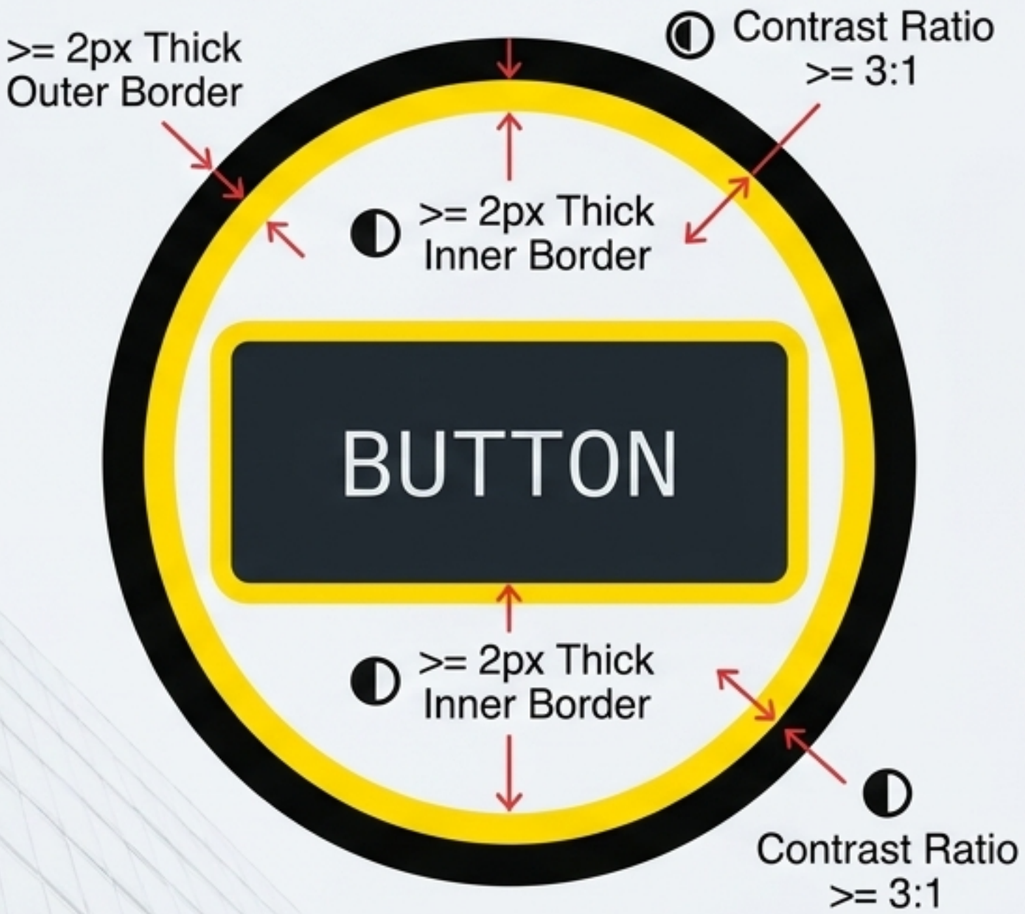
Gemini Notebook

Unobscured :focus-visible Indicators

WCAG 2.4.13 Requirements: Focus ring must be >= 2px thick and meet a >= 3:1 contrast ratio against the adjacent background.

Note: Using `:focus-visible` ensures mouse users don't see the ring, preserving aesthetic preferences without sacrificing keyboard accessibility.

```
button:focus-visible {
  outline: 3px solid #ffdd00; /* High contrast yellow inner */
  box-shadow: 0 0 0 5px #0b0c0c; /* Deep black outer border */
  outline-offset: 2px;
}
```



Gemini Notebook

Ch. 5 — 5-Modality Diagnostic Matrix

Pathway & Ops Desk: The Broken State

	Visual	Fake , outline: none, sticky header occluding target. ↳ Semantically Incorrect ↳ Focus Ring Removed ↳ Target Hidden
	Keyboard	28-stop header slog; tabindex='3' creates unpredictable jumps. ↳ Excessive Tabbing ↳ Non-Sequential Order
	Screen Reader	Announces: “No landmarks found. No next heading. Button. Link, Read more. Navigation.” ⟵ Missing Structure — Uninformative
	400% Zoom (320px viewport)	65% of viewport eaten by the sticky header; skip link overflows screen. ↳ Severe Occlusion ↳ Content Inaccessible
	Mobile / Voice	VoiceOver Rotor is empty; “Click Sign in” fails due to overwriting aria-label. ↳ Navigation Broken ↳ Label Conflict

Gemini Notebook

Pathway & Ops Desk: The Fixed State

5-Modality Success Matrix

	Visual	Real <h2> tags → Semantically Correct, dual-band :focus-visible ring → High Contrast Indicator, scroll-padding-top → Target Visibility Ensured implemented.
	Keyboard	1-press skip link → Efficient Navigation bypasses header; clean, predictable DOM order → Logical Sequence.
	Screen Reader	Announces: “Skip to lesson content. Course Primary, navigation landmark. Lesson Catalog, main landmark...” ← Clear Structure & Context
	400% Zoom (320px viewport)	Unstuck compact header → Occlusion Resolved; skip link text wraps gracefully → Content Accessible.
	Mobile / Voice	VoiceOver Rotor navigation fully restored → Functional Navigation (2 swipes); Voice Control registers visual button labels. → Accurate Interaction

Gemini Notebook

Ch. 6 — Annotated Code Diff & Refactoring

Refactoring Pathway's ** **-Soup Shell

Removed	Added
	Targetable by skip link
~~ ~~	<main id='main' tabindex='-1'>
	natural semantic landmark
~~ ~~	<nav aria-labelledby='toc-h2'>
	Scoped, named navigation region
	CSS file inclusion of
	html {
	scroll-padding-top: 5rem;
	}
	to prevent header overlap upon skip-link activation

Gemini Notebook

Refactoring the Operations Desk Header

Enhancing accessibility and semantic structure.

Removed / Refactored	Added
~~Link~~ ~~Link~~ ↑ tabindex='1' on all primary links	<nav aria-label='Primary'> ↑ aria-label='Primary' to distinguish from TOC <button type='button'> <svg aria-hidden='true' focusable='false' ...> ... </svg> ↑ Wrapped in <button type='button'> for keyboard interaction ↑ `aria-hidden='true' and `focusable='false' to SVGs

~~<svg ...> ...</svg>~~
~~<svg ...> ...</svg>~~
↑ Bare <svg> icons
	/* Global CSS */
	:focus-visible {
	outline: 2px solid #0b0c0c;
	outline-offset: 2px;
	box-shadow: 0 0 0 4px #ffdd00;
	}
	↑ Dual-band :focus-visible outline (replaces outline: none)

Gemini Notebook

Ch. 7 — Verification & Next Up

The 4-Point Navigation Checklist

- **1. Semantic Architecture:** Page possesses unique landmarks (`<main>`, `<nav>`) and an unbroken `h1`—`h6` tree.
- **2. Logical Flow:** DOM order equals visual order. Zero elements possess `tabindex > 0`.
- **3. Naming & Voice Control:** Follows `accname-1.2` hierarchy; respects WCAG 2.5.3 (Label in Name matches visible text).
- **4. Bypass & Focus:** Working skip link with `scroll-padding-top`; unobscured `:focus-visible` ring (>= `2px`, >= `3:1` contrast).

Technical terms

Dual-band ring for clarity

Teaser Card

Next Up in S10E03

Visual and Cognitive Accessibility

Gemini Notebook