

<h1>–<h6>: The HTML Section Heading elements

The <h1> to <h6> [HTML](#) elements represent six levels of section headings. <h1> is the highest section level and <h6> is the lowest. By default, all heading elements create a [block-level](#) box in the layout, starting on a new line and taking up the full width available in their containing block.

Try it

HTML Demo: <h1-h6> RESET

HTML

CSS

```
1 <h1>Beetles</h1>
2 <h2>External morphology</h2>
3 <h3>Head</h3>
4 <h4>Mouthparts</h4>
5 <h3>Thorax</h3>
6 <h4>Prothorax</h4>
7 <h4>Pterothorax</h4>
8
```

OUTPUT

Beetles

External morphology

Content categories	Flow content , heading content, palpable content.
Permitted content	Phrasing content .
Tag omission	None, both the starting and ending tag are mandatory.
Permitted parents	Any element that accepts flow content .
Implicit ARIA role	heading
Permitted ARIA roles	tab , presentation or none
DOM interface	HTMLHeadingElement

Attributes

These elements only include the [global attributes](#).

Usage notes

- Heading information can be used by user agents to construct a table of contents for a document automatically.
- Do not use heading elements to resize text. Instead, use the [CSS font-size](#) property.
- Do not skip heading levels: always start from <h1> , followed by <h2> and so on.

Avoid using multiple <h1> elements on one page

While using multiple <h1> elements on one page is allowed by the HTML standard (as long as they are not [nested](#)), this is not considered a best practice. A page should generally have a single <h1> element that describes the content of the page (similar to the document's [<title> element](#)).

Note: Nesting multiple `<h1>` elements in nested [sectioning elements](#) was allowed in older versions of the HTML standard. However, this was never considered a best practice and is now non-conforming. Read more in [There Is No Document Outline Algorithm](#) .

Prefer using only one `<h1>` per page and [nest headings](#) without skipping levels.

Examples

All headings

The following code shows all the heading levels, in use.

HTMLPlay

```
<h1>Heading level 1</h1>
<h2>Heading level 2</h2>
<h3>Heading level 3</h3>
<h4>Heading level 4</h4>
<h5>Heading level 5</h5>
<h6>Heading level 6</h6>
```

Play

Heading level 1

Heading level 2

Heading level 3

Heading level 4

Heading level 5

Heading level 6

Example page

The following code shows a few headings with some content under them.

HTMLPlay

```
<h1>Heading elements</h1>
<h2>Summary</h2>
<p>Some text here...</p>

<h2>Examples</h2>
<h3>Example 1</h3>
<p>Some text here...</p>

<h3>Example 2</h3>
<p>Some text here...</p>

<h2>See also</h2>
<p>Some text here...</p>
```

Play

Heading elements

Summary

Some text here...

Examples

Example 1

Some text here...

Example 2

Some text here...

See also

Some text here...

Accessibility concerns

Navigation

A common navigation technique for users of screen reading software is to quickly jump from heading to heading in order to determine the content of the page. Because of this, it is important to not skip one or more heading levels. Doing so may create confusion, as the person navigating this way may be left wondering where the missing heading is.

Don't do this:

HTML

```
<h1>Heading level 1</h1>
<h3>Heading level 3</h3>
<h4>Heading level 4</h4>
```

Prefer this:

HTML

```
<h1>Heading level 1</h1>
<h2>Heading level 2</h2>
<h3>Heading level 3</h3>
```

Nesting

Headings may be nested as subsections to reflect the organization of the content of the page. Most screen readers can also generate an ordered list of all the headings on a page, which can help a person quickly determine the hierarchy of the content:

- 1. h1 Beetles
 - i. h2 Etymology
 - ii. h2 Distribution and Diversity
 - iii. h2 Evolution
 - i. h3 Late Paleozoic
 - ii. h3 Jurassic
 - iii. h3 Cretaceous
 - iv. h3 Cenozoic

- iv. h2 External Morphology
 - i. h3 Head
 - i. h4 Mouthparts
 - ii. h3 Thorax
 - i. h4 Prothorax
 - ii. h4 Pterothorax
 - iii. h3 Legs
 - iv. h3 Wings
 - v. h3 Abdomen

When headings are nested, heading levels may be "skipped" when closing a subsection.

- [Headings • Page Structure • WAI Web Accessibility Tutorials](#)
- [MDN Understanding WCAG, Guideline 1.3 explanations](#)
- [Understanding Success Criterion 1.3.1 | W3C Understanding WCAG 2.0](#)
- [MDN Understanding WCAG, Guideline 2.4 explanations](#)
- [Understanding Success Criterion 2.4.1 | W3C Understanding WCAG 2.0](#)
- [Understanding Success Criterion 2.4.6 | W3C Understanding WCAG 2.0](#)
- [Understanding Success Criterion 2.4.10 | W3C Understanding WCAG 2.0](#)

Labeling section content

Another common navigation technique for users of screen reading software is to generate a list of [sectioning content](#) and use it to determine the page's layout.

Sectioning content can be labeled using a combination of the [aria-labelledby](#) and [id](#) attributes, with the label concisely describing the purpose of the section. This technique is useful for situations where there is more than one sectioning element on the same page.

Sectioning content examples

HTML

Play

```
<header>
  <nav aria-labelledby="primary-navigation">
    <h2 id="primary-navigation">Primary navigation</h2>
    <!-- navigation items -->
  </nav>
</header>

<!-- page content -->

<footer>
  <nav aria-labelledby="footer-navigation">
    <h2 id="footer-navigation">Footer navigation</h2>
    <!-- navigation items -->
  </nav>
</footer>
```

Play

Primary navigation

Footer navigation

In this example, screen reading technology would announce that there are two `<nav>` sections, one called "Primary navigation" and one called "Footer navigation". If labels were not provided, the person using screen reading software may have to investigate each `nav` element's contents to determine their purpose.

- [Using the aria-labelledby attribute](#)
- [Labeling Regions](#) • [Page Structure](#) • [W3C WAI Web Accessibility Tutorials](#)

Specifications

Specification
HTML Standard # the-h1,-h2,-h3,-h4,-h5,-and-h6-elements

Browser compatibility

[Report problems with this compatibility data on GitHub](#)

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet
h1	Chrome 1	Edge 12	Firefox 1	Opera 15	Safari 4	Chrome 18 Android	Firefox 4 for Android	Opera 14 Android	Safari 3.2 on iOS	Samsu Interr

Tip: you can click/tap on a cell for more information.

Full support

See also

- [<p>](#)
- [<div>](#)
- [<section>](#)

This page was last modified on Aug 11, 2023 by [MDN contributors](#).