# Elemen Struktur dan Dasar

HTML, HEAD, TITLE, BODY - BR, HR, HEADING, PARAGRAP, DIV, SPAN

L. Erawan

# BR (HTML ELEMENT)

| Spec | | |
|---|---|---|
| **Depr.** | **Empty** | **Version** |
| No | Yes | HTML 3.2 |

| Browser support | | | | |
|---|---|---|---|---|
| **IE5.5+** | **FF1+** | **SA1.3+** | **OP9.2+** | **CH2+** |
| Full | Full | Full | Full | Full |

## Syntax

```
<br/>
```

## Description

The `br` element's purpose is very simple: it creates a line break within a block of text, leaving no padding or margins between the two blocks of text created by the line break. While it's still perfectly valid to use this element in XHTML Strict pages (it's not on the list of deprecated elements), you need to take care that you don't misuse it, because:

- It can be used in a presentational manner. For example, you might use a series of `br` elements in succession to create a new paragraph effect, instead of simply using a or a `blockquote`, and applying CSS to set the layout.
- Using `br` elements becomes a real headache if, later, you want to correct visual inconsistencies and have to sweep through hundreds of files to strip them out.

There are some exceptional cases in which you might be forced to use a `br` element:

- In poetry, a new line requires just that: a new line. You can't use a `p` element in this case. (Evidently poetry wasn't high on the list of markup requirements when the HTML recommendations were thrashed out!)
- When you're marking up a postal address, you may need to create single line breaks. However, with the advent of Microformats, there's quite a well-established method for dealing with postal (and other) address types that avoids the use of the `br` while offering additional semantic richness. Refer to the section titled http://reference.sitepoint.com/html/hcard/ for more.

This shows the use of `br` in a poem (well, a poem of sorts):

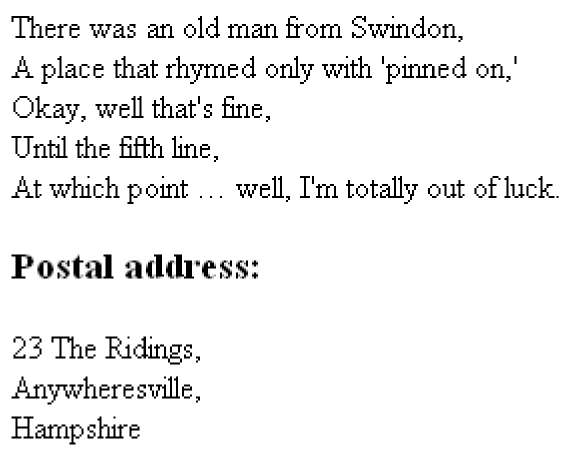
```
<p>There was an old man from Swindon,<br/>
    A place that rhymed only with 'pinned on,'<br/>
    Okay, well that's fine,<br/>
    Until the fifth line,<br/>
    At which point … well, I'm totally out of luck.</p>
```

Here, we use the `br` to create line breaks in a postal address:

```
<h3>Postal address:</h3>
<div class="adr">
```

```
    23 The Ridings,<br/>
    Anywheresville,<br/>
    Hampshire
</div>
```

These examples would render on screen as shown in Figure 1.



There was an old man from Swindon,
A place that rhymed only with 'pinned on,'
Okay, well that's fine,
Until the fifth line,
At which point … well, I'm totally out of luck.

**Postal address:**

23 The Ridings,
Anywheresville,
Hampshire

Figure 1. An example of the `br` element in use

Note that the examples shown here use the XHTML syntax for the `br` element, with a trailing slash to signify that the element is closed:

```
<br/>
```

If you specify an HTML doctype rather than an XHTML doctype, you would use the following (no trailing slash):

```
<br>
```

## Example

This code shows the `br` in action:

```
<p>If I wanted to create a line break right here,<br/>I
    could use a br element. But I'd feel dirty doing it. There may
    be no good reason for using one.</p>
```

## Use This For …

The `br` element is a self-closing element and doesn't contain any content or values.

## Compatibility

| Internet Explorer | | | | Firefox | | | | | Safari | | | | Opera | | | Chrome |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5.5 | 6.0 | 7.0 | 8.0 | 1.0 | 1.5 | 2.0 | 3.0 | 3.5 | 1.3 | 2.0 | 3.1 | 4.0 | 9.2 | 9.5 | 10.0 | 2.0 |
| **Full** | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full |

Every browser listed supports this element type.

## In this Section

- **clear**
  ensures that the line break occurs after any right- or left-aligned HTML elements

# h1 *(HTML ELEMENT)*

| Spec | | |
|---|---|---|
| Depr. | Empty | Version |
| **No** | No | HTML 3.2 |

| Browser support | | | | |
|---|---|---|---|---|
| IE5.5+ | FF1+ | SA1.3+ | OP9.2+ | CH2+ |
| **Full** | Full | Full | Full | Full |

## SYNTAX

```
<h1>

</h1>
```

## DESCRIPTION

The `h1` element is used to indicate the most important (or highest-level) heading on the page.

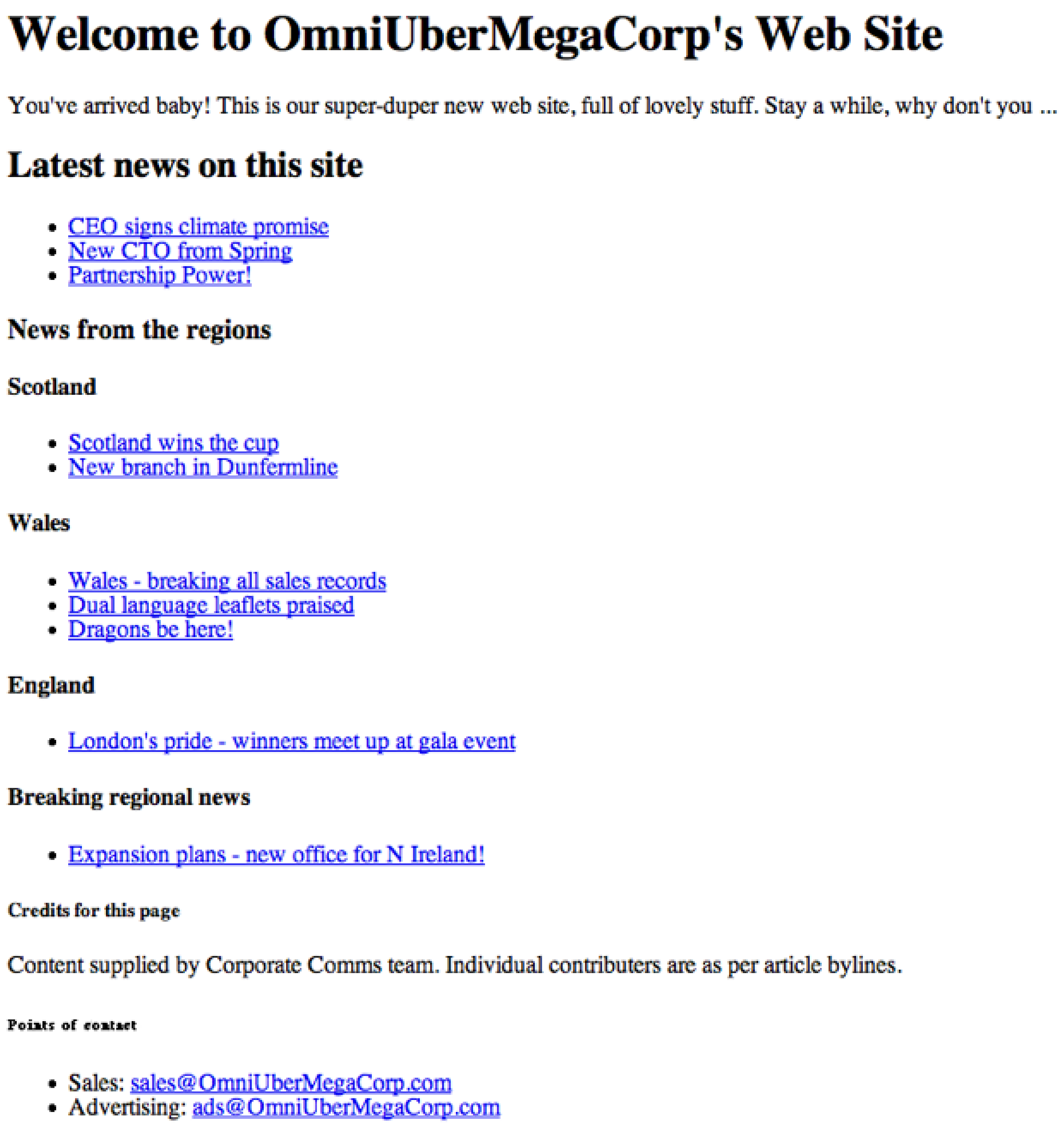
In total, we have six heading levels to choose from—`h1` to `h6`—to add structure to the web page. `h1` is the highest heading level (and, by default, the largest in terms of font size) and `h6` the lowest (and smallest).

A document's first heading should be an `h1`, followed by one or more `h2` headings; each of these `h2` headings can then have a further series of `h3` headings below them, and so on, right on down to heading level 6. The HTML 4 spec states that heading levels should not be skipped (that is, you shouldn't have a series of headings in the order `h1`, `h2`, `h2`, `h4`, which skips the `h3` entirely), although it isn't always possible to guarantee such rigidity in the markup, particularly if your pages are generated by a CMS. However, this goal is certainly one for which you should ***aim***.

Headings add semantic richness to a document, which can help with search engines' understanding of the makeup of that document, and provide users of assistive devices (such as screen readers) with an additional—and quick—method by which to navigate through a document: they can skip from heading to heading.

Figure 1 shows a comparison of the six heading levels in a fictional web site, as rendered by Firefox's default browser style sheet.

Figure 1. A comparison of heading sizes in Firefox



## Welcome to OmniUberMegaCorp's Web Site

You've arrived baby! This is our super-duper new web site, full of lovely stuff. Stay a while, why don't you ...

### Latest news on this site

- CEO signs climate promise
- New CTO from Spring
- Partnership Power!

#### News from the regions

##### Scotland

- Scotland wins the cup
- New branch in Dunfermline

##### Wales

- Wales - breaking all sales records
- Dual language leaflets praised
- Dragons be here!

##### England

- London's pride - winners meet up at gala event

##### Breaking regional news

- Expansion plans - new office for N Ireland!

###### Credits for this page

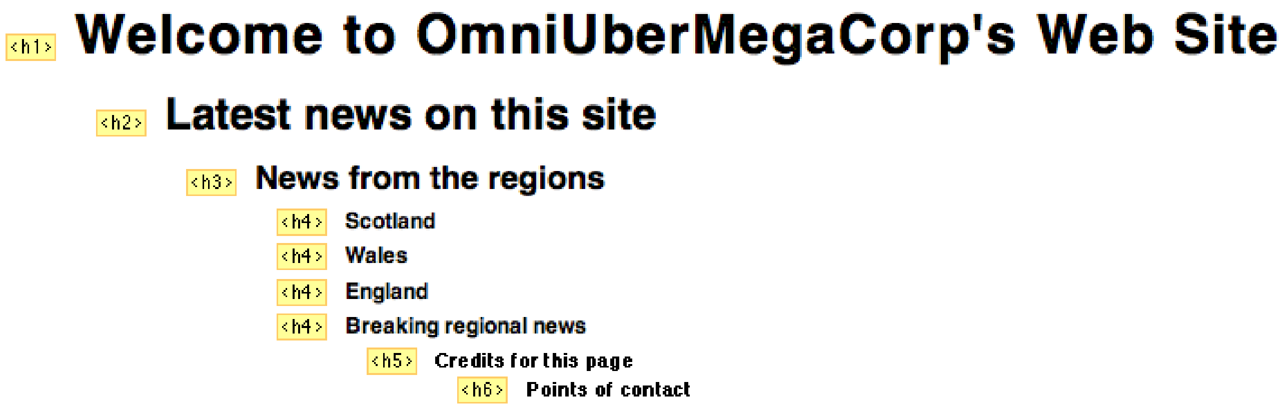
Content supplied by Corporate Comms team. Individual contributers are as per article bylines.

###### Points of contact

- Sales: sales@OmniUberMegaCorp.com
- Advertising: ads@OmniUberMegaCorp.com

The document outline, showing the heading levels, is clear to see using Firefox's Web Developer extension (via that tool's Information > View Document Outline command), as shown in Figure 2.

Figure 2. Headings/document outline revealed using Firefox's Web Developer Toolbar



<h1> Welcome to OmniUberMegaCorp's Web Site

<h2> Latest news on this site

<h3> News from the regions

<h4> Scotland

<h4> Wales

<h4> England

<h4> Breaking regional news

<h5> Credits for this page

<h6> Points of contact

## EXAMPLE

This `h1` element is used to present a fluffy welcome message:

```
<h1>Welcome to OmniUberMegaCorp's Web Site</h1>
```

## USE THIS FOR ...

This element may contain any text content, but it can’t include any block-level elements: only inline or phrase elements can be included.

## COMPATIBILITY

| Internet Explorer | | | | Firefox | | | | | Safari | | | | Opera | | | Chrome |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5.5 | 6.0 | 7.0 | 8.0 | 1.0 | 1.5 | 2.0 | 3.0 | 3.5 | 1.3 | 2.0 | 3.1 | 4.0 | 9.2 | 9.5 | 10.0 | 2.0 |
| Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full |

There are no compatibility issues with the `h1` element—all browsers listed support it.

## IN THIS SECTION

- **align**
  aligns the content inside an h1 element

# h2 *(HTML ELEMENT)*

| Spec | | |
|---|---|---|
| Depr. | Empty | Version |
| No | No | HTML 3.2 |

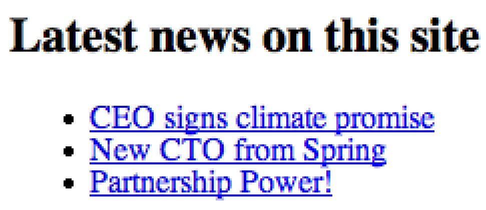
| Browser support | | | | |
|---|---|---|---|---|
| IE5.5+ | FF1+ | SA1.3+ | OP9.2+ | CH2+ |
| Full | Full | Full | Full | Full |

## SYNTAX

```
<h2>

</h2>
```

## DESCRIPTION

The `h2` element is used to indicate a heading whose level of importance is exceeded only by `h1`. A document may have several `h2` elements, all of which share the same level of importance. The default heading size is shown in Figure 1.



**Latest news on this site**

- CEO signs climate promise
- New CTO from Spring
- Partnership Power!

Figure 1. The default level 2 heading

In total, we have six heading levels to choose from—`h1` to `h6`—to add structure to the web page. `h1` is the highest heading level (and, by default, the largest in terms of font size) and `h6` the lowest (and smallest).

A document's first heading should be an `h1`, followed by one or more `h2` headings; each of these `h2` headings can then have a further series of `h3` headings below them, and so on, right on down to heading level 6. The HTML 4 spec states that heading levels should not be skipped (that is, you shouldn't have a series of headings in the order `h1`, `h2`, `h2`, `h4`, which skips the `h3` entirely), although it isn't always possible to guarantee such rigidity in the markup, particularly if your pages are generated by a CMS. However, this goal is certainly one for which you should ***aim***.

Headings add semantic richness to a document, which can help with search engines' understanding of the makeup of that document, and provide users of assistive devices (such as screen readers) with an additional—and quick—method by which to navigate through a document: they can skip from heading to heading.

## Example

In this example, an `h2` element is used to define a section heading:

```
<h2>Latest news on this site</h2>
```

## Use This For ...

This element may contain any text content, but it can’t include any block-level elements: only inline or phrase elements can be included.

## Compatibility

| Internet Explorer | | | | Firefox | | | | | Safari | | | | Opera | | | Chrome |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5.5 | **6.0** | **7.0** | **8.0** | **1.0** | **1.5** | **2.0** | **3.0** | **3.5** | **1.3** | **2.0** | **3.1** | **4.0** | **9.2** | **9.5** | **10.0** | **2.0** |
| **Full** | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full |

This element has no compatibility issues. All browsers listed support the `h2` element.

## In This Section

- **align**
  aligns the content inside the h2 element

# H3 *(HTML ELEMENT)*

| Spec | | |
|---|---|---|
| Depr. | Empty | Version |
| **No** | No | HTML 3.2 |

| Browser support (more...) | | | | |
|---|---|---|---|---|
| IE5.5+ | **FF1+** | **SA1.3+** | **OP9.2+** | **CH2+** |
| **Full** | Full | Full | Full | Full |

## SYNTAX

```
<h3>

</h3>
```

## DESCRIPTION

The `h3` element is used to indicate a heading whose level of importance is exceeded by `h1` and `h2`. A document may have several `h3` elements, all of which share the same level of importance. The default heading size is shown in Figure 1.

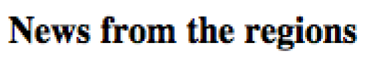

News from the regions

Figure 1. A level 3 heading

In total, we have six heading levels to choose from—`h1` to `h6`—to add structure to the web page. `h1` is the highest heading level (and, by default, the largest in terms of font size) and `h6` the lowest (and smallest).

A document's first heading should be an `h1`, followed by one or more `h2` headings; each of these `h2` headings can then have a further series of `h3` headings below them, and so on, right on down to heading level 6. The HTML 4 spec states that heading levels should not be skipped (that is, you shouldn't have a series of headings in the order `h1`, `h2`, `h2`, `h4`, which skips the `h3` entirely), although it isn't always possible to guarantee such rigidity in the markup, particularly if your pages are generated by a CMS. However, this goal is certainly one for which you should ***aim***.

Headings add semantic richness to a document, which can help with search engines' understanding of the makeup of that document, and provide users of assistive devices (such as screen readers) with an additional—and quick—method by which to navigate through a document: they can skip from heading to heading.

## EXAMPLE

In this example, an `h3` element is used to define a section heading:

```
<h3>News from the regions</h3>
```

## USE THIS FOR ...

This element may contain any text content, but it can’t include any block-level elements: only inline or phrase elements can be included.

## COMPATIBILITY

| Internet Explorer | | | | Firefox | | | | | Safari | | | | Opera | | | Chrome |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5.5 | **6.0** | **7.0** | **8.0** | **1.0** | **1.5** | **2.0** | **3.0** | **3.5** | **1.3** | **2.0** | **3.1** | **4.0** | **9.2** | **9.5** | **10.0** | **2.0** |
| **Full** | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full |

The `h3` element suffers no compatibility issues: all browsers listed support it.

## IN THIS SECTION

- **align**
  aligns the content inside h3 element

# H4 *(HTML ELEMENT)*

| Spec | | |
|---|---|---|
| Depr. | Empty | Version |
| No | No | HTML 3.2 |

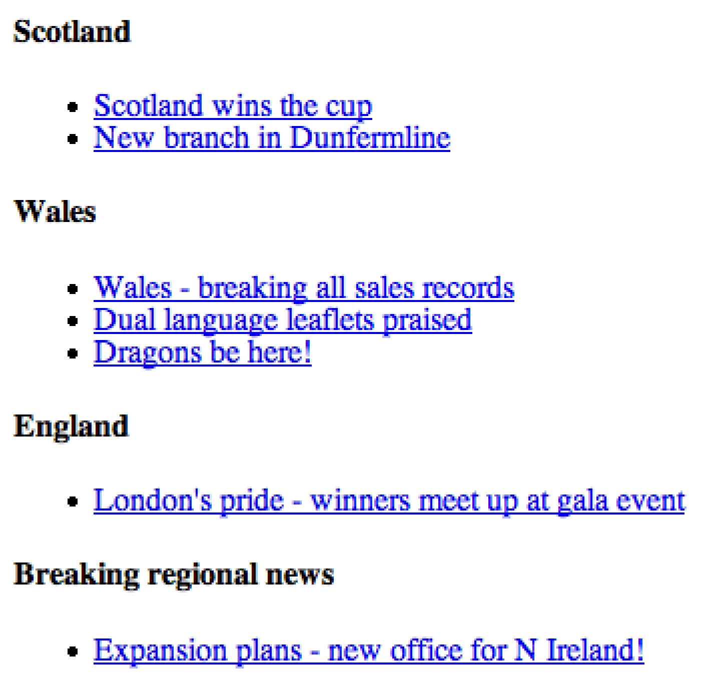
| Browser support | | | | |
|---|---|---|---|---|
| IE5.5+ | FF1+ | SA1.3+ | OP9.2+ | CH2+ |
| Full | Full | Full | Full | Full |

## SYNTAX

```
<h4>

</h4>
```

## DESCRIPTION

The `h4` is used to indicate a heading whose level of importance is exceeded by `h1`, `h2`, and `h3`. A document may have several `h4` elements, all of which share the same level of importance. The default heading size is shown in Figure 1.



**Scotland**

- Scotland wins the cup
- New branch in Dunfermline

**Wales**

- Wales - breaking all sales records
- Dual language leaflets praised
- Dragons be here!

**England**

- London's pride - winners meet up at gala event

**Breaking regional news**

- Expansion plans - new office for N Ireland!

Figure 1. A level 4 heading

In total, we have six heading levels to choose from—`h1` to `h6`—to add structure to the web page. `h1` is the highest heading level (and, by default, the largest in terms of font size) and `h6` the lowest (and smallest).

A document's first heading should be an `h1`, followed by one or more `h2` headings; each of these `h2` headings can then have a further series of `h3` headings below them, and so on, right on down to heading level 6. The HTML 4 spec states that heading levels should not be skipped (that is, you shouldn't have a series of headings in the order `h1`, `h2`, `h2`, `h4`, which skips the `h3` entirely), although it isn't always possible to guarantee such rigidity in the markup, particularly if your pages are generated by a CMS. However, this goal is certainly one for which you should ***aim***.

Headings add semantic richness to a document, which can help with search engines' understanding of the makeup of that document, and provide users of assistive devices (such as screen readers) with an additional—and quick—method by which to navigate through a document: they can skip from heading to heading.

## EXAMPLE

In this example, an `h4` element is used to define a section heading:

```
<h4>Breaking regional news</h4>
```

## USE THIS FOR ...

This element may contain any text content, but it can't include any block-level elements: only inline or phrase elements can be included.

## COMPATIBILITY

| Internet Explorer | | | | Firefox | | | | | Safari | | | | Opera | | | Chrome |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5.5 | **6.0** | **7.0** | **8.0** | **1.0** | **1.5** | **2.0** | **3.0** | **3.5** | **1.3** | **2.0** | **3.1** | **4.0** | **9.2** | **9.5** | **10.0** | **2.0** |
| **Full** | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full |

The `h4` element suffers no compatibility issues: all browsers listed support it.

## IN THIS SECTION

- **align**
  aligns the content inside h4 element

# H5 *(HTML ELEMENT)*

| Spec | | |
|---|---|---|
| Depr. | **Empty** | **Version** |
| **No** | No | HTML 3.2 |

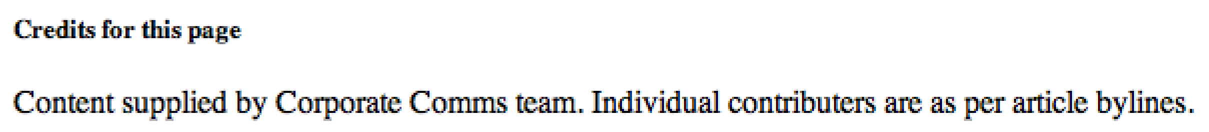
| Browser support | | | | |
|---|---|---|---|---|
| IE5.5+ | **FF1+** | **SA1.3+** | **OP9.2+** | **CH2+** |
| **Full** | Full | Full | Full | Full |

## SYNTAX

```
<h5>

</h5>
```

## DESCRIPTION

The `h5` element is used to indicate a heading whose level of importance is exceeded by `h1`, `h2`, `h3`, and `h4`. A document may have several `h5` elements, all of which share the same level of importance. The default heading size is shown in Figure 1.



**Credits for this page**

Content supplied by Corporate Comms team. Individual contributers are as per article bylines.

Figure 1. A level 5 heading

In total, we have six heading levels to choose from—`h1` to `h6`—to add structure to the web page. `h1` is the highest heading level (and, by default, the largest in terms of font size) and `h6` the lowest (and smallest).

A document's first heading should be an `h1`, followed by one or more `h2` headings; each of these `h2` headings can then have a further series of `h3` headings below them, and so on, right on down to heading level 6. The HTML 4 spec states that heading levels should not be skipped (that is, you shouldn't have a series of headings in the order `h1`, `h2`, `h2`, `h4`, which skips the `h3` entirely), although it isn't always possible to guarantee such rigidity in the markup, particularly if your pages are generated by a CMS. However, this goal is certainly one for which you should ***aim***.

Headings add semantic richness to a document, which can help with search engines' understanding of the makeup of that document, and provide users of assistive devices (such as screen readers) with an additional—and quick—method by which to navigate through a document: they can skip from heading to heading.

## EXAMPLE

In this example, an `h5` element is used to define a section heading:

```
<h5>Credits for this page</h5>
```

## USE THIS FOR ...

This element may contain any text content, but it can’t include any block-level elements: only inline or phrase elements can be included.

## COMPATIBILITY

| Internet Explorer | | | | Firefox | | | | | Safari | | | | Opera | | | Chrome |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5.5 | **6.0** | **7.0** | **8.0** | **1.0** | **1.5** | **2.0** | **3.0** | **3.5** | **1.3** | **2.0** | **3.1** | **4.0** | **9.2** | **9.5** | **10.0** | **2.0** |
| **Full** | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full |

The `h5` element suffers no compatibility issues: all browsers listed support it.

## IN THIS SECTION

- **align**
  aligns the content inside h5 element

# H6 *(HTML ELEMENT)*

| Spec | | |
|---|---|---|
| Depr. | Empty | Version |
| No | No | HTML 3.2 |

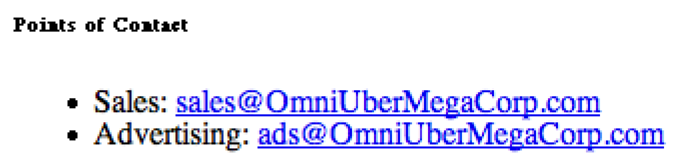
| Browser support | | | | |
|---|---|---|---|---|
| IE5.5+ | FF1+ | SA1.3+ | OP9.2+ | CH2+ |
| Full | Full | Full | Full | Full |

## SYNTAX

```
<h6>

</h6>
```

## DESCRIPTION

The `h6` element is used to indicate a heading whose level of importance is exceeded by `h1`, `h2`, `h3`, `h4`, and `h5` (or to look at it another way, it has the ***lowest*** level of importance). A document may have several `h6` elements, all of which share the same level of importance. The default heading size is shown in Figure 1.



**Points of Contact**

- Sales: sales@OmniUberMegaCorp.com
- Advertising: ads@OmniUberMegaCorp.com

Figure 1. A level 6 heading

In total, we have six heading levels to choose from—`h1` to `h6`—to add structure to the web page. `h1` is the highest heading level (and, by default, the largest in terms of font size) and `h6` the lowest (and smallest).

A document's first heading should be an `h1`, followed by one or more `h2` headings; each of these `h2` headings can then have a further series of `h3` headings below them, and so on, right on down to heading level 6. The HTML 4 spec states that heading levels should not be skipped (that is, you shouldn't have a series of headings in the order `h1`, `h2`, `h2`, `h4`, which skips the `h3` entirely), although it isn't always possible to guarantee such rigidity in the markup, particularly if your pages are generated by a CMS. However, this goal is certainly one for which you should ***aim***.

Headings add semantic richness to a document, which can help with search engines' understanding of the makeup of that document, and provide users of assistive devices (such as screen readers) with an additional—and quick—method by which to navigate through a document: they can skip from heading to heading.

## Example

In this example, an `h6` element is used to define a section heading:

```
<h6>Points of Contact</h6>
```

## Use This For ...

This element may contain any text content, but it can’t include any block-level elements: only inline or phrase elements can be included.

## Compatibility

| Internet Explorer | | | | Firefox | | | | | Safari | | | | Opera | | | Chrome |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5.5 | **6.0** | **7.0** | **8.0** | **1.0** | **1.5** | **2.0** | **3.0** | **3.5** | **1.3** | **2.0** | **3.1** | **4.0** | **9.2** | **9.5** | **10.0** | **2.0** |
| **Full** | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full |

The `h6` element suffers no compatibility issues: all browsers listed support it.

## In this Section

- **align**
  aligns the content inside h6 element

# HR *(HTML ELEMENT)*

| Spec | | |
|---|---|---|
| Depr. | **Empty** | **Version** |
| **No** | Yes | HTML 3.2 |

| Browser support | | | | |
|---|---|---|---|---|
| IE5.5+ | **FF1+** | **SA1.3+** | **OP9.2+** | **CH2+** |
| **Full** | Full | Full | Full | Full |

## SYNTAX

```
<hr/>
```

## DESCRIPTION

The `hr` element creates in the document a highly visible break that renders as a slim horizontal line running the width of the area to which it's applied. While it's still perfectly valid to use this element in XHTML strict pages, as it's not on the list of deprecated elements, it isn't used a great deal these days, because:

- It's difficult to style consistently across browsers through CSS, or via its own presentational attributes.
- In many cases, it may be better to use a combination of headings and lists to define the document structure, as this will promote ease of navigation for users of assistive technology; the CSS `border` property can be used to visually style a break in the document.

## EXAMPLE

In this example, an `hr` is used to separate body content from footer information:

```
<p>And with that the actress and the bishop made their
    eventful departure.</p>
<hr/>
<div id="footer">&copy; All content copyright 2007. Even
    the unfunny stuff.</div>
```

## USE THIS FOR ...

The `hr` element can be used to create a break in a document at a point where there may be a change of thought or meaning, but where it may not necessarily be appropriate to introduce a subheading. A real-world equivalent for this element can be found in books where a divider may appear as three asterisks, or some other collection of characters to indicate a change of scene or momentum.

## COMPATIBILITY

| Internet Explorer | | | | Firefox | | | | Safari | | | | Opera | | | Chrome |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5.5 | **6.0** | **7.0** | **8.0** | **1.0** | **1.5** | **2.0** | **3.0** | **3.5** | **1.3** | **2.0** | **3.1** | **4.0** | **9.2** | **9.5** | **10.0** | **2.0** |
| **Full** | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full |

Every browser listed supports this element type.

However, the way that the `hr` element renders differs greatly between browsers, so it doesn't pay to be too precious about design consistency in this case.

## IN THIS SECTION

- **color**
  applies a color to the horizontal rule
- **noshade**
  removes 3D bevel/shading effect on hr element
- **size**
  sets a height for the horizontal rule
- **width**
  sets a width for the horizontal rule

# P (HTML ELEMENT)

| Spec | | |
|---|---|---|
| Depr. | Empty | Version |
| No | No | HTML 3.2 |

| Browser support | | | | |
|---|---|---|---|---|
| IE5.5+ | FF1+ | SA1.3+ | OP9.2+ | CH2+ |
| Full | Full | Full | Full | Full |

## SYNTAX

```
<p>

</p>
```

## DESCRIPTION

The `p` element is one of the most commonly used building blocks of HTML. When you use the `p` element to begin a new paragraph in HTML, it automatically creates some space above and below the content. This space is applied by the browser's built-in style sheets, but you can override it as you see fit using CSS.

In XHTML, it's necessary to wrap the contents of a paragraph in opening `<p>` and closing `</p>` tags, while in HTML 4 and earlier versions, it was enough to signify a new paragraph using the opening <p>—no closing tag was needed.

For example, this markup would be perfectly valid in HTML 3.2:

```
<p>So, having failed miserably to manage something as simple as
    "order a breakfast and pay for it," we set about our next task:
    submitting a visa application for our impending visit to
    Thailand. Heaven help us! We're not normally scatty like this,
    so we really must have needed that breakfast!
<p>Thankfully, the rest of the day was nothing like our poor start.
    We got the visa submitted and approved without a hitch, so
    that's one less task for us to do now.
```

The same markup would not be acceptable in XHTML, but this would:

```
<p>So, having failed miserably to manage something as simple as
    "order a breakfast and pay for it," we set about our next task:
    submitting a visa application for our impending visit to
    Thailand. Heaven help us! We're not normally scatty like this,
    so we really must have needed that breakfast!</p>
<p>Thankfully, the rest of the day was nothing like our poor start.
    We got the visa submitted and approved without a hitch, so
    that's one less task for us to do now.</p>
```

Importantly, while the first example would fail validation as XHTML, the second example will pass, and will ***also*** pass validation as HTML 3.2.

For the purposes of forwards-compatibility and general good coding practice, it's advisable to use both the opening and closing tags even if you're writing a document that uses an earlier, looser HTML DTD, such as 3.2 or 4.01—this approach supports completeness, rather than shortcuts.

Some developers perceive similarities between the `p` and the `div` elements, seeing them as being interchangeable, but this isn't the case. The `p` element offers more semantic information ("this is a paragraph of text, a small collection of thoughts that are grouped together; the next paragraph outlines some different thoughts"), while the `div` element can be used to group almost any elements together. Indeed, it can contain almost any other element, unlike `p`, which can only contain inline elements.

## EXAMPLE

`p` is used, quite simply, to mark up paragraphs of text:

```
<p>This is a paragraph of text…</p>
<p>…and this is another paragraph.</p>
```

## USE THIS FOR ...

This element may contain any text content, but it can't include any block-level elements: only inline or phrase elements can be included.

## COMPATIBILITY

| Internet Explorer | | | | Firefox | | | | | Safari | | | | Opera | | | Chrome |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5.5 | **6.0** | **7.0** | **8.0** | **1.0** | **1.5** | **2.0** | **3.0** | **3.5** | **1.3** | **2.0** | **3.1** | **4.0** | **9.2** | **9.5** | **10.0** | **2.0** |
| **Full** | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full |

It causes no compatibility issues, and has excellent support across all tested browsers.

## IN THIS SECTION

- **align**
  aligns the content inside p element

# SPAN *(HTML ELEMENT)*

| Spec | | |
|---|---|---|
| Depr. | Empty | Version |
| No | No | HTML 4 |

| Browser support | | | | |
|---|---|---|---|---|
| IE5.5+ | FF1+ | SA1.3+ | OP9.2+ | CH2+ |
| Full | Full | Full | Full | Full |

## SYNTAX

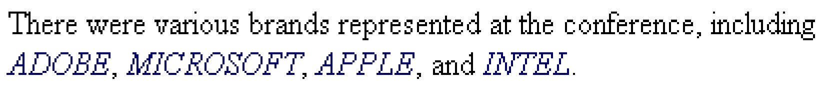
```
<span>

</span>
```

## DESCRIPTION

For an element that offers no semantic information about the content inside and also provides no styling change, or any other visual change to speak of, the lowly `span` element is one of the most useful elements in your HTML toolbox.

When you wrap text with an opening `<span>` and closing `</span>`, you're simply providing a hook—one that allows you to add styles (by adding a `class` attribute and using CSS to define the look of that class), or interact with the element via JavaScript and the Document Object Model (DOM).

In the example shown here, the author has decided that all brand names should be classed as `"brandname"` and styled differently via CSS, in italic, uppercase letters, as Figure 1 shows.



There were various brands represented at the conference, including *ADOBE*, *MICROSOFT*, *APPLE*, and *INTEL*.

Figure 1. Company names styled via span elements

A `span` is an inline element, and must only contain text content or nested inline or phrase elements. It shouldn't be used to surround block-level elements—a usage that's often seen in Content Management Systems which attempt to apply styling to almost any element by throwing a span around it.

The `span` is closely related to the `div` element, which is a block-level generic container, as opposed to `span`, which is an inline generic container.

A note of caution: it's not unheard of for people to go crazy with `span`s, using them all over the place. `span`-itis is a bad practice—it's just as bad as a dose of `div`-itis. Be sure to check that you're using the span element appropriately. For example, if you find yourself applying spans like this, you're in trouble:

```
He said it was <span class="important">really</span> important
    to know the difference.
```

It's clear that in the example above, the `em` element would have been more appropriate, as it implies emphasis in all browsers. On the other hand, without CSS styling, `<span class="important">` would be all but meaningless.

## EXAMPLE

Here's an example of the `span` element being used for CSS styling purposes:

```
.brandname {font-style:italic;color:#006;text-transform:uppercase;}

⋮
<p>There were various brands represented at the
    conference, including <span class="brandname">Adobe</span>,
    <span class="brandname">Microsoft</span>, <span
    class="brandname">Apple</span>, and <span
    class="brandname">Intel</span>.</p>
```

## USE THIS FOR ...

This element can be used to mark up text content of any kind.

## COMPATIBILITY

| Internet Explorer | | | | Firefox | | | | | Safari | | | | Opera | | | Chrome |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5.5 | **6.0** | **7.0** | **8.0** | **1.0** | **1.5** | **2.0** | **3.0** | **3.5** | **1.3** | **2.0** | **3.1** | **4.0** | **9.2** | **9.5** | **10.0** | **2.0** |
| **Full** | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full |

The `span` has full support in the browsers tested.

# `div` *(HTML element)*

| Spec | | |
|---|---|---|
| **Depr.** | **Empty** | **Version** |
| **No** | No | HTML 3.2 |

| Browser support | | | | |
|---|---|---|---|---|
| **IE5.5+** | **FF1+** | **SA1.3+** | **OP9.2+** | **CH2+** |
| **Full** | Full | Full | Full | Full |

## Syntax

```
<div> </div>
```

## Description

The `div` is a generic block-level element. It doesn't convey any meaning about its contents (unlike a `p` element that signifies a paragraph, or an `h1` or `h2` element that would indicate a level 1 or level 2 heading, respectively); as such, it's easy to customize it to your needs. The `div` element is currently the most common method for identifying the structural sections of a document and for laying out a web page using CSS.

Some developers perceive similarities between the `p` and the `div` elements, seeing them as being interchangeable, but this isn't the case. The `p` element offers more semantic information ("this is a paragraph of text, a small collection of thoughts that are grouped together; the next paragraph outlines some different thoughts"), while the `div` element can be used to group almost any elements together. Indeed, it can contain almost any other element, unlike `p`, which can only contain inline elements.

## Example

The HTML below shows two `div`s being used in conjunction with `id` attributes to identify different sections of a web page:

```
<div id="main_navigation">....</div>
<div id="body_content">
  <h1>Page heading</h1>
  <p>Body content</p>
</div>
```

## Use This For …

The `div` is an "anything-goes" element—it can contain any inline or block-level elements you choose, so it has no typical content.

## Compatibility

| Internet Explorer | | | | Firefox | | | | | Safari | | | | Opera | | | Chrome |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **5.5** | **6.0** | **7.0** | **8.0** | **1.0** | **1.5** | **2.0** | **3.0** | **3.5** | **1.3** | **2.0** | **3.1** | **4.0** | **9.2** | **9.5** | **10.0** | **2.0** |
| **Full** | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full | Full |

This element has no compatibility issues. All the browsers listed support the `div` element.

## In this Section

- **align**
  aligns the content inside a div element