

HTML Tag Examples

Intro

In this module, we are going to look at a small subset of tags in detail so we can look at how they are actually used. Then we will look at a larger collection of tags and talk about what they are used for in more general terms.

Headers (h1-h6)

Headers are a pretty intuitive tag. Things within a header tag are represented as a header. This usually means they are given a larger font and sometimes a bold font or even a different typeface. A header tag is created like so: `<h1>An Important Header</h1>`. The opening and closing tags are always required. `h1` is the most important or highest level header. Headers run from `h1` through `h6` with `h6` being the lowest level header that you might find deep in a paragraph. In the document you are reading right now the title at the top of the page is an `h1` and the header at the top of this section is an `h2`. If there are subsections they are `h3` headers.

By default headers are block level elements. This means they get their section of the page. If a page break does not exist before or after it one will be created to ensure it is on its own line.

When to Use Headers

If you would potentially find something listed in the table of contents it is a great candidate for a header. Headers should be used to add organization to a document. So any section or sub section that needs a title can use a header for it. One could make a good argument that they could also be used to add a title to something like an image. But it would not be good to use as the title for a table because tables have their own element for adding a title called `caption`.

When not to Use Headers

Headers make text big and obvious. It might be tempting to use something like an `h5` or `h6` to add emphasis to a word in a paragraph or maybe use it as a way to highlight dialog. This is not what headers are to be used for. It might look good, but things like screen readers are going to use headers to section content for visually impaired users. It will really confuse those tools if headers are used to just add emphasis to text or just to style it.

Section, Article and Div

These elements along with a few others are used to section off content. Visually these do very little on their own. They will usually break their content into a block so that a line break comes before and after. Beyond that how they are styled really depends on the CSS that is used. Although they often all do the same thing visually it is important to understand why they are used.

The Div

The `div` element `<div>Content Here</div>` is used to divide content. It should be used as a last resort when no other element makes sense. If you want to divide content purely for stylistic reasons this is the element to use. It conveys no meaning and just generically divisions content.

The Section

The `section` element:

```
<section>
  <h2>Some Section</h2>
  <p>I am content in the section</p>
</section>
```

is used to make a thematic grouping of content. In this document I have "Section, Article and Div" as a section and "Headers" as a section. It is a group of content that is all related but does not quite stand on its own. Usually a sections first child will be a header that describes what is in that section. If there is not a good way to classify the content in the section using a header, you may want to consider using a `div` instead. A section should only be used if all the content is related.

The Article

The `article` element is structurally the same as the section except that it is an `<article>` tag instead of a `<section>` tag. The requirements for a article are stricter than a section. Not only should the content all be related but the content should generally sand on its own as a composition.

For example, this section of the page just describing the article element makes references to other parts of the page. So it is all thematically grouped around the article element. But it is not a complete composition. So it is left as a section rather than made into an article. This page as a whole could be an article as it is a complete composition describing element examples.



Anchors (a) and Images (img)

Anchors

The anchor `A link to foo` is how one links from one page to another. Both the opening and closing tag are required. Anything between the tags will represent the link someone can click on to navigate to the target specified by the anchor. The `href` attribute is what is used to specify where the link will take the user when it is clicked.

Images

The `img` tag is used to display images: ``. The `src` attribute indicates where the image is located. No closing tag is used for images. By default images display in-line. So no new line is made for them. We will talk a bit more about specifying the source for images later on as identifying the location of resources can get a bit tricky at times.

Strong, B, Em, i...

This family of elements is for adding meaning or style to specific text within a paragraph. This is different from the previous elements which generally served to group text.

Strong

The `strong` element: `I am important` marks text **that is more important**. It requires an opening and closing tag. The important thing about `strong` is that it signifies that text is important. Usually browsers will make it bold by default. But it could make it red, it could make it underlined, it could make it blink (don't do this...). By marking text with `strong` you are saying it is more important than the text around it.

B

The `b` element makes text stylistically different from other text. Look at me, I am `different`. This could be used to highlight **keywords** in a paragraph or conform to style guidelines set by some journal. It does not give additional meaning to the text. Text inside a `b` element is no more important than text outside of the `b` element. It just is styled differently.

Em

The `em` element adds emphasis to a word. Often making it italic. "I love kittens" suggests that the emphasis is on the I. So maybe someone is differentiating themselves in a room of dog lovers. On the other hand "I love *kittens*" suggests that it is only kittens that I love and that I may not be such a big fan of older cats. It is adding additional meaning to the word or phrase that is emphasized. There is also an `i` element which will make text italic but not convey additional meaning. Again this might be used to conform to style guides or the like but should not be used when wanting to add emphasis to a word.

Lists and Tables

So far we have looked at elements that really were self contained or could include all kinds of different stuff. But there are elements that are more complex. Lists and Tables fit into this category as they are really collections of several nested elements.

Lists

Lists come in several different flavors, ordered, unordered or definition list. We will use an unordered list in our example here:

```
<ul>
  <li>first item</li>
  <li>second item</li>
  <li>third item</li>
</ul>
```

The `ul` element starts and ends the unordered list. It can only contain list items (`li`) as children. Those list items can however contain most other HTML elements. The above list will look like this:

- first item
- second item
- third item

Often times styling is used to add or remove bullet points to the list. Ordered lists will be numbered or otherwise sequenced, unordered lists will not. Here is the same list, now ordered (`ol` instead of `ul`):

1. first item
2. second item
3. third item

Definition lists (`dl`) are similar but have pairs of terms and definitions

```
<!-- Code example from https://developer.mozilla.org -->
<dl>
  <dt>Name</dt>
  <dd>Godzilla</dd>
  <dt>Born</dt>
  <dd>1952</dd>
  <dt>Birthplace</dt>
  <dd>Japan</dd>
  <dt>Color</dt>
  <dd>Green</dd>
</dl>
```

This is rendered as:

Name
Godzilla
Born
1952
Birthplace

Japan
Color
Green

The Table

To finish this off we will look at what is probably the most structurally complex HTML element out there, the `table`. Here is an example table we will look at. Below is a sample table from the w3c standard.

```
<table>
<caption>Characteristics with positive and negative sides</caption>
<thead>
<tr>
<th> Characteristic
<th> Negative
<th> Positive
<tbody>
<tr>
<th> Mood
<td> Sad
<td> Happy
<tr>
<th> Grade
<td> Failing
<td> Passing
</table>
```

And here is what that table looks like when it is rendered in your browser:

Characteristics with positive and
negative sides

Characteristic	Negative	Positive
Mood	Sad	Happy
Grade	Failing	Passing

So lets use a definition list here to talk about what all these pieces are doing.

Table

This opens and closes the table, it is pretty straight forward. Everything related to the table will be inside these tags.

Caption

This represents the title and description of the table. It can include just the title or additional explanation of the table.

thead

This is the table header. It can contain rows that represent the header of the table. So these would be rows that do not contain actual data but instead contain things like labels signifying the data that will be in those columns. The rows in this section would typically contain `th` elements.

tbody

This is the body of the table and contains the rows that actually have data in them rather than the descriptive rows that would be in the header. The rows in this section would typically contain `td` elements.

tr

This is a row of the table. It contains either `td` elements (which are data cells) or `th` elements (which are header cells).

td

This is a data cell, it should contain data from the table. Each cell in a row will typically signify a column if every row has the same number of cells. If you want to have a cell span multiple columns you can use the `colspan` attribute.

th

This is a header cell it should signify what is in the column below it or the row beside it.



A Sample Skeleton

Beyond the above tags there are some basic tags that should exist in every page. Below is a skeleton you can use to get started when making an HTML page.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>title</title>
    <link rel="stylesheet" href="stylesheet.css" type="text/css">
  </head>
  <body>

  </body>
</html>
```

The content in the head is metadata. It will help the browser and robots understand what is on the page. The link tag is where we can link to a file that contains CSS rules to style the page. The actual content the user will see all goes in the body of the page.



Activity

As an activity you should try to make a moderately complex table. To do this you will certainly want a reference to look up all the options. I strongly recommend Mozilla's table content reference (https://developer.mozilla.org/en-US/docs/Web/HTML/Element#Table_content). Your task is to as close as possible, recreate the table shown below.

Edit in JSFiddle

- [Result](#)

Column 1 heading	Column 2 heading	Column 3 heading
Row 2, cell 1	Row 2, cell 2, also spanning Row 2, cell 3	
Row 3, cell 1,	Row 3, cell 2	Row 3, cell 3

Go to this fiddle (<http://jsfiddle.net/wolfordj/s6apw402/>) at jsfiddle.com and using just the HTML panel put in HTML that will create the example table. CSS is added to create the borders so it is easier to see what is going on. If you get stuck you can see how it was made below by clicking on the HTML panel

Edit in JSFiddle

- [Result](#)
- [HTML](#)

Column 1 heading	Column 2 heading	Column 3 heading
Row 2, cell 1	Row 2, cell 2, also spanning Row 2, cell 3	
Row 3, cell 1,	Row 3, cell 2	Row 3, cell 3

Review

This should give you a good understanding of the basic HTML elements. There are lots of other elements but with these you should be able to put together a complete page and it should give you an idea of the different categories of elements you might find in HTML.