HTML & CSS: The fabric of the web

HTML and CSS are **two distinct technologies** that are used almost **always together** by your web browser to **assemble** web pages.

> “Every web page is code made visual by the web browser. Open any page in a web browser and use the View menu to see the page source. There you will see the code used to create that page. This code tells the browser how to render the layout, images, links, and interactivity of the page. Whereas in other applications, we click and drag our way to good design, with web pages we design interface elements and then write the code that describes what we want our interface to look like in the browser.”
>
> from **Digital Foundations**, a creative commons textbook (with modifications). http://www.digital-foundations.net/

HTML = **What** goes on the page; **content**. (headlines, text, images)

CSS = **How** the page looks; **style** (colour, fonts, layout, etc.)

HTML is necessary because without saying what goes on the web page, there is simply no web page.

house metaphor: **HTML is the bricks & beams**.

CSS is also necessary, because in addition to making your page look nice, it sets you apart and makes your page easier to read and navigate.

house metaphor: **CSS is the plastering, paint, furniture, & carpets.**

The ability to **separate content from style** is really important, because sometimes the style needs to be ignored. For example, the case of **visually impaired people, or automated use of HTML**. When HTML was used for style this was nearly impossible, which meant **accessibility** was a real problem, and doing anything cool involving **automatic processing of web page content** was impossible.

1. HTML

HTML is made of small instructions called “tags”.

Tags generally have two parts, an opening tag and a closing tag. For example, in:

```
<h1>This is the title of your page</h1>
```

`<h1>` is the opening part, and `</h1>` is the closing part. Together they tell the web browser that everything between them should be displayed as a headline.

Note the forward slash to distinguish the opening tag from the closing tag. **99% of HTML tags work just like this**. There are very few that don’t, but one of them is important: the image tag.

```
<img src="path/to/the/image.jpg" alt="text alternative for the image" />
```

This tag communicates everything it needs to via **attributes**. Attributes are the parts that use an equals sign to **specify additional pieces of information about the tag**.

So the `<img` bit simply says “here is an image”. The attributes do the rest of the work.

You can put elements inside each other.

`<p>` is the tag used for paragraphs.

`<h1>`, `<h2>`, `<h3>`, `<h4>` are used for headlines, declining in size.

`<a href="http://cutlines.org">Cutlines</a>` is used to make links.

`<span>` is used for any word or collection of words that you want to group together.

`<div>` should really be called “`<box>`” (although it isn’t). You can put other elements inside a to group them together in an invisible box. The box can even be made visible, using CSS.

2. CSS

CSS is made of sets of instructions called "**declarations**".

Each CSS declaration **refers directly to HTML tags**. A declaration can refer to:

1. All tags of a certain type. For example all paragraphs.

2. All tags in a smaller group of tags called a "class". You can put an HTML tag into a new class by adding `class="[name of class]"` to it as an attribute. For example:

```
<a class="external" href="...">External</a>
```

will add a link to a class called "external".

You do not have to do anything else to create a class. Refer to it, and it exists. The name of a class **cannot contain a space**.

3. A single, specific tag. To do this, the tag must have a unique name, which you give it using the id attribute:

```
<p id="my-paragraph">This is a specific paragraph</p>
```

This paragraph now has the ID "my-paragraph" and consequently, CSS can be written to change only this paragraph. An ID also **cannot contain a space**.

So what does a CSS declaration look like?

First, here's one to make all paragraphs turn red:

```
p{
color:#f00;
}
```

Next, here's one to make only paragraphs in the group "red-paragraphs" turn red:

```
.red-paragraphs{
color:#f00;
}
```

so, as far as HTML, one of the paragraphs that this refers to might look like:

```
<p class="red-paragraphs">paragraph 1</p>
```

Finally, here's one to make only a single paragraph, with the id "my-paragraph", turn red:

```
#my-paragraph{
color:#f00;
}
```

so, as far as HTML, if this declaration referred to a paragraph, it would look like:

```
<p id="my-paragraph">paragraph</p>
```

So, a CSS declaration has **two main parts**:

```
#my-paragraph{
   color:#f00;
}
```

1. The **selector**, which says **which bit** of the page we are changing

2. The **properties**, which say **how** the parts named in the selector are to be changed.

(Don't forget the closing curly bracket!)

As for the properties themselves, there are loads of them, which you can find here: http://bit.ly/cssref

Colours

Colours in CSS are specified using hex codes. Hex codes work very similarly to normal numbers, except that while normal numbers have this range: 0,1,2,3,4,5,6,7,8,9
Hex has six extra 'numbers':
0,1,2,3,4,5,6,7,8,9,**a,b,c,d,e,f**
so the next value up from 9 is 'a'.

All colours on a TV or computer screen are stored as **quantities of red, green, and blue, in that order**. Hex colours use the counting system to the left to do this:
`#f00` - red, using just one digit for each colour, **or**
`#ff0000` - red using two digits for each colour.
So the form can either be **`#rgb`** or **`#rrggbb`**.