

Creative Coding

The web as a medium

Part 2: CSS

CSS

Cascading Style Sheets

HTML describes

```
<nav>
  <ul>
    <li>
      <a href="/home">Home</a>
    </li>
    <li>
      <a href="/artists">Artists</a>
    </li>
    <li>
      <a href="/exhibitions">Exhibitions</a>
    </li>
  </ul>
</nav>
```

- [Home](#)
- [Artists](#)
- [Exhibitions](#)

But still has some style. The defaults.

Font, size, bullets, blue, underlining, left padding... everything in fact.

Slightly different in different browsers.

CSS styles

```
ul {  
  font-family: Helvetica, sans-serif;  
  padding-left: 20px;  
  background: #c33;  
}  
li {  
  display: inline-block;  
  margin-right: 20px;  
}  
a {  
  display: block;  
  color: white;  
  text-decoration: none;  
  padding: 0.3em 10px;  
}  
a:hover {  
  background: #900;  
}
```

[Home](#) [Artists](#) [Exhibitions](#)



Styles applied to tags

```
p {  
  font-size: 24px;  
  margin-bottom: 20px;  
}
```

Basic syntax

```
selector {  
  property: value;  
  property: value value value value;  
}
```


Styles applied to tags within tags

```
p a {  
  color: green;  
}
```

Styles applied to multiple tags

```
p,  
a {  
  color: green;  
}
```

HTML classes

```
<p class="heavy">  
  This is a special paragraph.  
</p>
```

Styles applied to classes

```
.heavy {  
  font-size: 24px;  
  font-weight: bold;  
  color: #900;  
}
```

HTML IDs

```
<p id="intro">  
  This is an introductory paragraph.  
</p>
```

Styles applied to IDs

```
#intro {  
  font-family: Georgia, serif;  
  font-size: 1.5em;  
  line-height: 2em;  
}
```

Best practice these days: Use only classes. Mostly.

Tags, classes and IDs

```
p#intro.heavy {  
  text-decoration: underline;  
}
```

Applies *only* to p#intro.heavy

Now try some out on an example.html

Display

```
display: block;
```

```
display: inline;
```

```
display: inline-block;
```

Every element has a display setting. By default, elements are usually block or inline.

display: block;

An `<h1>` is a block level element like this.

A `<p>` is also a block level element and so it stretches the width of the page.

So is a `<div>`.

Block elements by default have 100% width, no matter how much or little is in them. So they always start a new line.

display: inline;

A `<p>` is a block level element but a `` is inline and so are tags like `` and `` (which makes things **bold**). Text in inline tags wrap, and are only as big as their contents.

Inline elements are as small as their contents.

They will wrap if their content stretches over more than one line.

display: inline-block;

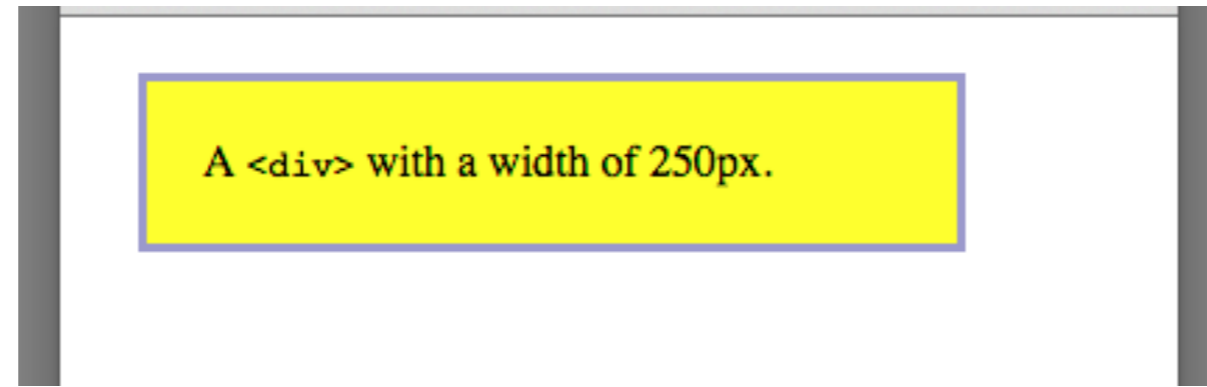
inline-block elements shrink to fit their contents, like *inline elements*. But, like block elements, they don't wrap across lines.

And so if they have a lot of stuff inside they'll stretch the full width and look like block elements with normal text before and after separate.

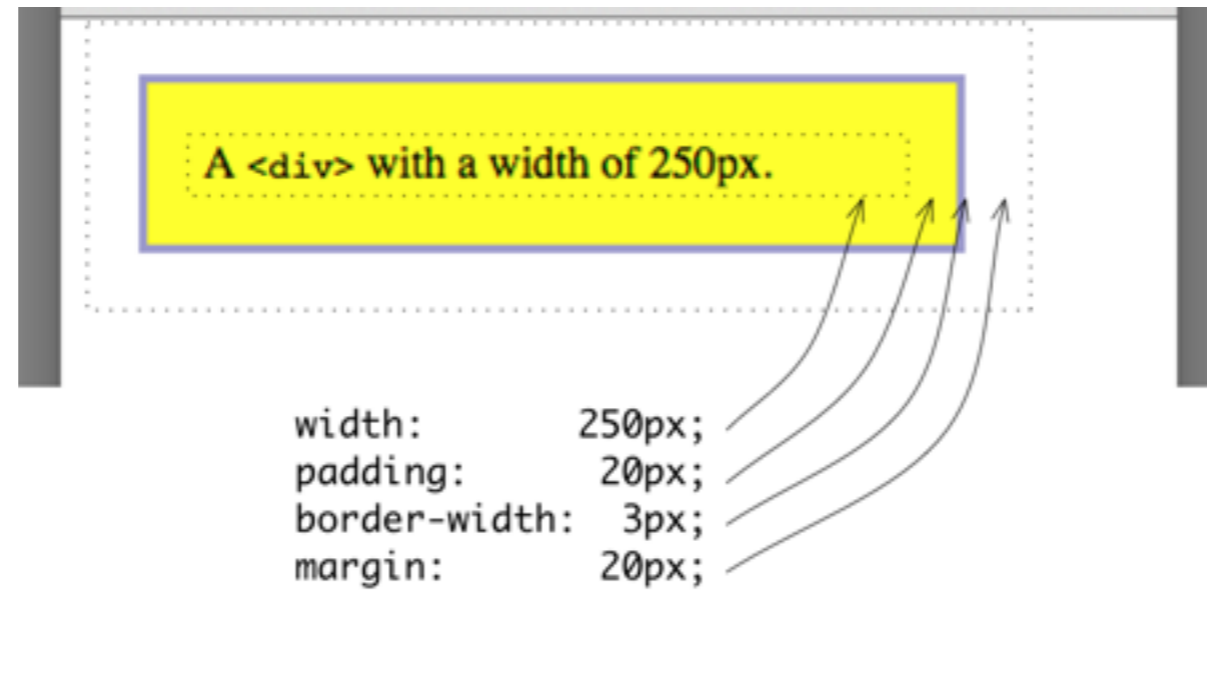
Inline elements are as small as their contents.

They will wrap if their content stretches over more than one line.

The box model



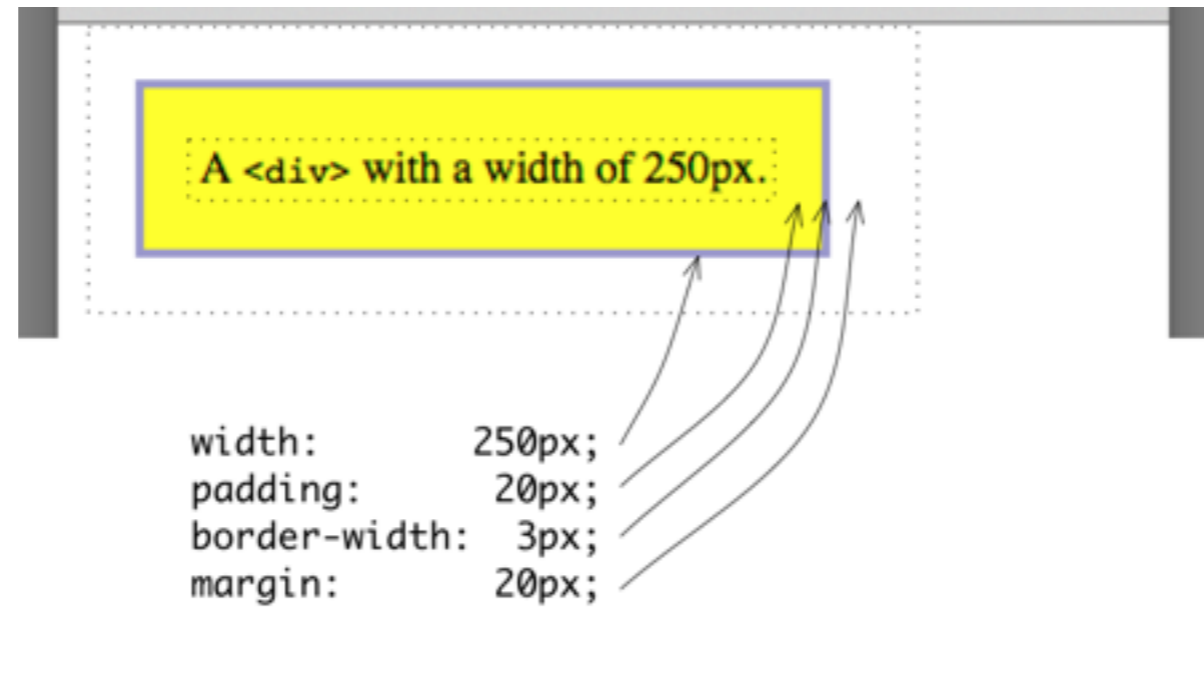
The box model



But...

```
* {  
  box-sizing: border-box;  
}
```

box-sizing: border-box;



More: <https://css-tricks.com/box-sizing/>

Colors (not Colours)

```
color: red;  
  
color: #f03;  
color: #F03;  
  
color: #ff0033;  
color: #FF0033;  
  
color: rgb(255, 0, 51);  
  
color: rgba(255, 0 51, 1);  
  
color: rgb(100%, 0%, 20%);
```

In Atom: Atom > Preferences > Install > “color-picker”

Hexadecimal (base 16)

0 1 2 3 4 5 6 7 8 9 A B C D E F

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Hexadecimal (base 16)

0 1 2 3 4 5 6 7 8 9 A B C D E F

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

23 = two 10s + three 1s

Hexadecimal (base 16)

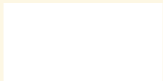





0 1 2 3 4 5 6 7 8 9 A B C D E F

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

23 = two 10s + three 1s

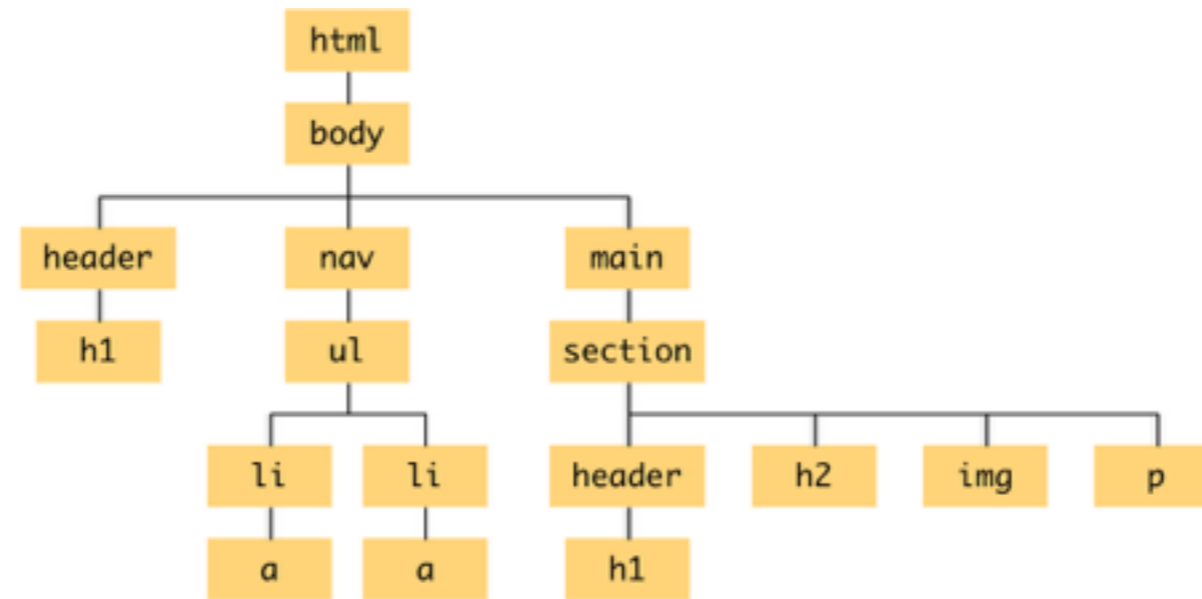
CB = twelve 16s + eleven 1s = $192+11 = 203$

Colors (not Colours)

<code>color: #fff;</code> <code>color: #ffffff;</code> <code>color: rgb(255, 255, 255);</code>	
<code>color: #000;</code> <code>color: #000000;</code> <code>color: rgb(0, 0, 0);</code>	
<code>color: #AAA;</code> <code>color: #AAAAAA;</code> <code>color: rgb(170, 170, 170);</code>	
<code>color: #dc143c;</code> <code>color: rgb(221, 20, 60);</code>	
<code>color: #7fff00;</code> <code>color: rgb(127, 255, 0);</code>	
<code>color: #6495ed;</code> <code>color: rgb(100, 149, 237);</code>	

In Atom: Atom > Preferences > Install > “color-picker”

Inheritance



Some styles, such as color, font-size and font-family are inherited. Others, like border, margin and padding, aren't.

<https://developer.mozilla.org/en-US/docs/Web/CSS/inheritance>

Specificity

#intro	beats	.heavy
.heavy	beats	p
p.heavy a	beats	p a
#intro a	beats	p.heavy a

Useful links (again)

- **Reference:** devdocs.io
- **and:** developer.mozilla.org/en-US/docs/Web/CSS
- **Help:** stackoverflow.com
- **Browser support:** caniuse.com
- **Learn more:** codecademy.com/tracks/htmlcss