

CSS Typography

How do we change the default font?

System or “Web Safe” fonts

Arial

Verdana

Trebuchet

Calibri

Sans-serif fonts

Times New Roman

Georgia

Courier New

Cambria

Serif fonts

A browser can only display the fonts specified in CSS rules if that font is available on the client computer.*

System fonts are those that we can be sure exist on the viewer’s computer. For example, all the standard Windows or MacOS fonts that are installed with the operating system.

*Unless the font is being served from a web font service such as [Google Fonts](#).

The font-family property

```
h1{  
    font-family: Georgia, "Times New Roman", Times, serif;  
    font-size: 3.0em;  
    color: #534741;  
    padding-bottom: 5px;  
}
```

The font-family property is used to specify fonts. Rather than specifying one font, it's good practice to specify a "font stack". If the first font is not available, the second will be used and so on. In the example above, Georgia is the preferred font but if it's not available, Times New Roman will be used. In general, a stack should have a minimum of 3 fonts; one for Windows, one for MacOS and a default generic font for anything else.

The font-size property

```
h1{  
    font-family: Georgia, "Times New Roman", Times, serif;  
    font-size: 3.0em;  
    color: #534741;  
    padding-bottom: 5px;  
}
```

The font-size property is used to set text at the height specified. You might think that font sizes would be set using pixels (px), and you can do that if you want but, best practice is to use a proportional unit. Here, we're using the "em". By default, one em is equal to 16px but the beauty (and sometimes frustration) is that the default can change, depending on context.

Let's have a go. We'll specify a font stack using the font-family property.

The CSS Box Model

How do we create space between elements?

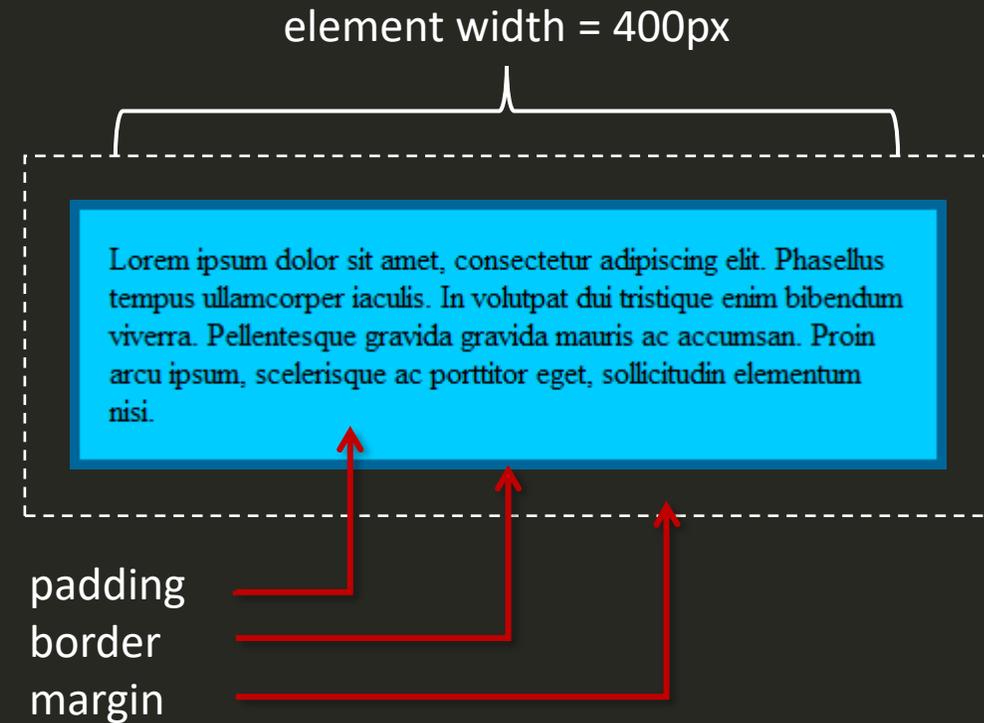
The CSS “Box Model”



The box model applies to every HTML element (block-level *and* inline), so you can think of every paragraph, heading or image as a box.

Box Model sizing

```
p {  
  width: 400px;  
  padding: 15px;  
  border: 5px solid #069;  
  margin: 20px;  
  background-color:#0CF;  
}
```



By default, element width does **not** include padding, border or margin, so in this case, the total width is $400 + 30 + 10 + 40 = 480\text{px}$. This default method for calculating element width is often problematic because simply changing the margin on an element could break your page layout.

Note that padding takes the element background colour, the border has an independent colour and the margin is transparent.

Box-sizing in CSS

The traditional CSS box model, which adds padding and border to the element width and height often causes confusion. Surely it would make more sense to have a model where the stated width of an element is fixed and the border and padding are subtracted from that width and not added to them. The result would be an element whose dimensions remain the same irrespective of the padding and border.

Well, you can now choose to have the box model behave in this way if you wish, using the CSS *box-sizing* property and setting the value to “border-box”:

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

This sounds like a perfect solution and the only problem is that this property will not be recognised by older browsers but current support is good. In the example above, we’re using the universal selector to set all elements to obey this box sizing rule.

Margin and Padding

```
main {  
    margin: 20px;  
}
```

```
main {  
    margin-top: 20px;  
    margin-right: 20px;  
    margin-bottom: 20px;  
    margin-left: 20px;  
}
```

```
main {  
    margin: 20px 20px 20px 20px;  
}
```

All three rules on the left are equivalent – all four margins are set to 20px.

It is possible to set the width of padding, border and margin for the top, right, bottom and left of any element independently as in example 2.

Most often this is done using the shorthand notation (example 3) using the following order:

margin: top right bottom left;
begin at the top and move clockwise round the box.

Border

```
main {  
  border: 1px solid #FFFFFF;  
}
```

```
main {  
  border-top: 1px solid  
#FFFFFF;  
  border-bottom: 2px  
solid #EEEEEE;  
}
```

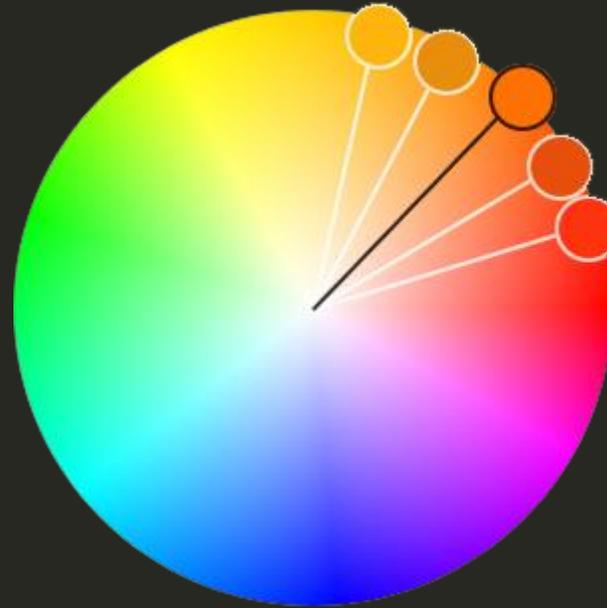
The border property is slightly different because it takes three values, *width*, *style* and *colour*.

Let's have a go. We'll add some margins and borders.

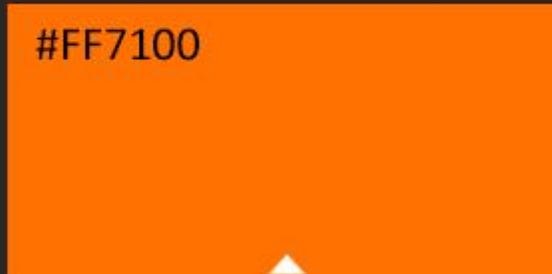
Colour with CSS

How do we change the colour of elements?

How many colours are there?



#FF7100



Specifying colours in CSS

Colours can be specified in many ways, here are the most common:

Colour name	color: <code>red</code> ;
Hexadecimal (Base 16 RGB)	color: <code>#FF0000</code> ;
Hex shorthand	color: <code>#F00</code> ;
RGB (Red, Green, Blue)	color: <code>rgb(255,0,0)</code> ;
RGBA (RGB plus Alpha)	color: <code>rgba(255, 0, 0, 1)</code> ;

Alpha specifies the degree of opacity in a colour in a range from 0 = transparent to 1 = Opaque. All the colours specified above are **Red**. Currently, hexadecimal is most common but RGBA tends to be used when changes to opacity are required.

Let's have a go. We'll specify our colours using hexadecimal.

CSS Inheritance

Do children inherit the styles of their parents?

CSS Inheritance

Some properties are inherited by child elements from their parent and others are not. In general, font and colour properties are inherited while box model properties (e.g. margin, border) are not.

For example, if you wanted the font to be the same on all text elements, you could add a style rule for the <body> and all elements within body would inherit that font.

```
/* all text to be Verdana */  
body {  
    font-family: Verdana, Geneva, sans-serif;  
}
```

On the other hand, you wouldn't want padding inherited from <body>

```
/* only body gets padding */  
body {  
    padding: 20px;  
}
```

Generally, inheritance works as you might expect.

<https://www.sitepoint.com/css3-inheritance-tips-tricks/>

Working with Inheritance

One common situation you may face is that you want all text on a page (paragraphs, lists, tables etc.) to display in one font except for the headings, which you want to display in a different font. Working with inheritance, we can easily achieve this outcome with just 2 style rules:

```
/* all text to be Verdana */
body {
  font-family: Verdana, Geneva, sans-serif;
}
/* except headings */
h1, h2, h3, h4, h5, h6 {
  font-family: Georgia, Times, "Times New Roman", serif;
}
```

The first rule sets all text to Verdana (including headings) but the second rule overrides the first for headings h1-h6. This feature of CSS is known as the cascade and the way it works depends on the order and specificity of the style rules. This can be a little complicated, so for now, let's keep things simple and just accept that it works for the example above.

Validate

 **The W3C CSS Validation Service**
W3C CSS Validator results for <http://websitearchitecture.co.uk> (CSS level 3)

Jump to: [Warnings \(2\)](#) [Validated CSS](#)

W3C CSS Validator results for <http://websitearchitecture.co.uk> (CSS level 3)

Congratulations! No Error Found.

This document validates as [CSS level 3](#) !

To show your readers that you've taken the care to create an interoperable Web page, you may display this icon on any page that validates. Here is the XHTML you could use to add this icon to your Web page:

	<pre><p> </p></pre>
	<pre><p> </p></pre>

CSS can be validated in the same way as HTML using the W3C CSS Validation Service.

<https://jigsaw.w3.org/css-validator/>

```
<p class="end">The End</p>
```