

# COMP 1017

Styling Text

Whenever we design something,  
it is with a function or purpose in mind.

In web design, our goal is usually the dissemination of information — that is, **getting a message out**.

Therefore, **typography** is going to be a key component of the overall look and tone of our design.

# Web Safe Fonts

Let's sing a song about safety ...

Many of the fonts that we use in our everyday life are actually optimised for **print**.

And we're, uh ... definitely not printing anything in this course.

So, what about typography  
optimised for display on **screens**?



And, more to the point, will a client be able to actually see the font that we choose?

Web safe fonts are fonts that are pre-installed on many operating systems.

[cssfontstack.com/](https://cssfontstack.com/)

If you want to use fonts other than ones that come pre-installed, you can use **Web Fonts**.

This means pointing to a font file on a server in the `<head>` of our HTML document.

Our browser will then go fetch that file and use it to render the page.

One of the largest and most popular repository of fonts is Google Fonts.

[fonts.google.com/](https://fonts.google.com/)



Once you've chosen your font(s),  
Google Fonts will generate the  
code that you need.

```
<link
```

```
href="https://fonts.googleapis.com/css?family=Open+  
Sans&display=swap" rel="stylesheet">
```

This goes in your `<head>` of your markup.

```
font-family: 'Open Sans', sans-serif;
```

This goes in your CSS.

# Typography in CSS

A hobby for people who hate hobbies.

Now that we've got our fonts,  
what about styling our type?

So far, we've seen the following  
CSS rules for styling type/fonts ...

```
font-family: Arial, sans-serif;  
color: #333333;  
font-weight: bold;  
list-style-type: none;  
line-height: 1.5;
```

But there are many more CSS rules  
that we can use to style our text!



```
font-family: Lato, sans-serif;
```

This defines the font being applied to the  
element.

```
font-size: 36px;
```

This sets the height of a font.

```
font-weight: 700;
```

This sets the weight of a font.

```
/* FONT WEIGHTS */
```

```
100 - thin
```

```
300 - light
```

```
400 - regular
```

```
500 - medium
```

```
600 - semi-bold
```

```
700 - bold
```

```
800 - extra bold
```

```
900 - black
```

```
/* FONT WEIGHTS CONT'D */
```

Some fonts may have more or  
fewer numeric values; it  
depends on the font itself.

```
font-style: normal |  
            italic | oblique;
```

This lets you choose whether  
your font is slanted or not.

```
font-variant: normal |  
                small-caps;
```

This changes the font to small-caps (or not).

```
font-stretch: ultra-condensed | extra-  
condensed | condensed | semi-condensed |  
normal | semi-expanded | expanded | extra-  
expanded | ultra-expanded;
```

This only works if the font you've selected  
has these values.



```
line-height: 1.5;
```

Defines the amount of space  
above and below inline  
elements.

... that's a lot of properties and values. Fortunately, we can **combine** them!

```
font: font-style font-variant font-  
weight font-size/line-height font-  
family;
```

```
font: normal small-caps 700 48px/1.5 'Lato', sans-serif;
```

Finally, letter and word spacing allows us to put spaces between our letters and our words.

```
p { letter-spacing: 2px; }
```

```
p { word-spacing: 32px; }
```

```
text-transform: uppercase | lowercase |  
capitalize;
```

And text can be aligned to the left,  
middle, or the right.

```
p { text-align: left; }
```

```
p { text-align: center; }
```

```
p { text-align: right; }
```



Finally, we can change the appearance of some of our text if we like...

`<span>` is very much like a `<div>` element, but `<div>` is a block-level element whereas a `<span>` is an inline element.