

Welcome

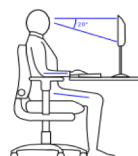
Web Authoring: CSS for Beginners (Level 2)

Simon Meaker – UIS Training Services

ssm38@uis.cam.ac.uk

1

Health and Safety



2

Course Information

Objectives

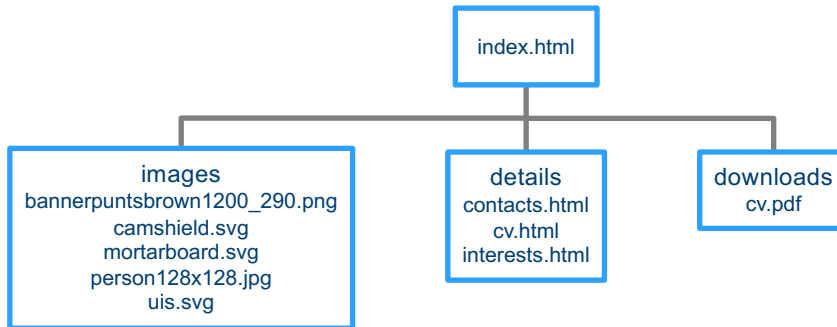
- To style a pre-written website by:
 - editing the HTML
 - creating a CSS (Cascading Style Sheet)
- To understand the structure and syntax of CSS
- Pre-requisites, attendance on HTML5 L1
- Duration, Breaks & Refreshments
- Delivery Style
- Course Material

Course Information

- Please sign the attendance sheet
- Please fill in the online feedback at the end of the course (shortcut on the Desktop)
<http://feedback.training.cam.ac.uk/uis>
- Let us know if you need assistance; please ask questions
- Related Courses:
 - www.linkedin.com/learning
 - www.w3schools.com
 - www.htmldog.com

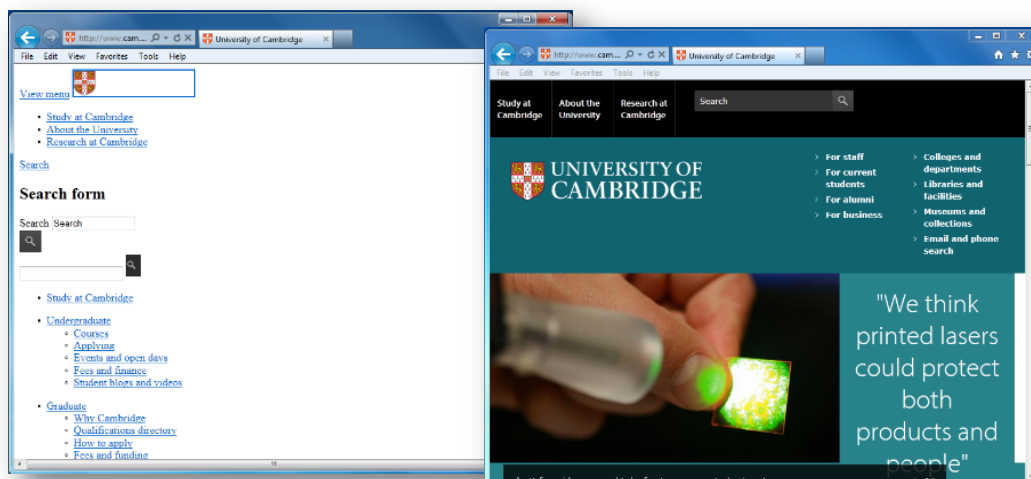
Relative Links

`Curriculum Vitae`



`Home Page`

Content & Style



Content & Style

Name of Person

PERSON TITLE AND / OR POSITION

- [Home](#)
- [Curriculum Vitae](#)
- [My Interests](#)
- [Contact Details](#)

YOUR NAME

Introduction text about who you are and what you do. Lorem ipsum dolor sit amet conset etueter adipiscing elit sed dolor sit amet conset adipiscing elit.

Institutional Information

[UIS Training Services](#)

Lorem ipsum dolor sit amet, conset etueter adipiscing elit.

[University Information Services](#)

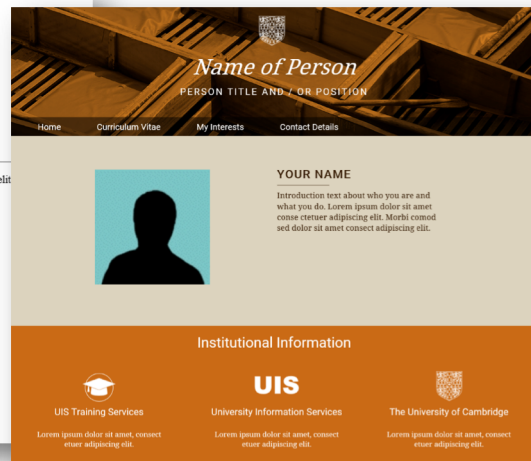
Lorem ipsum dolor sit amet, conset etueter adipiscing elit.

[The University of Cambridge](#)

Lorem ipsum dolor sit amet, conset etueter adipiscing elit.

Last updated April 2018

By Simon Mesker
University Information Services



Content & Style

Web authors should aim to separate **structure and content** from **style and layout** as this brings the following benefits:

- It reduces the need for repeated styling statements (i.e. control style centrally using CSS as opposed to doing it in multiple individual webpages)
- It helps make your webpages machine readable (e.g. better for visually impaired or blind user)
- It produces a more consistent result

Cascading Order

How a webpage is displayed

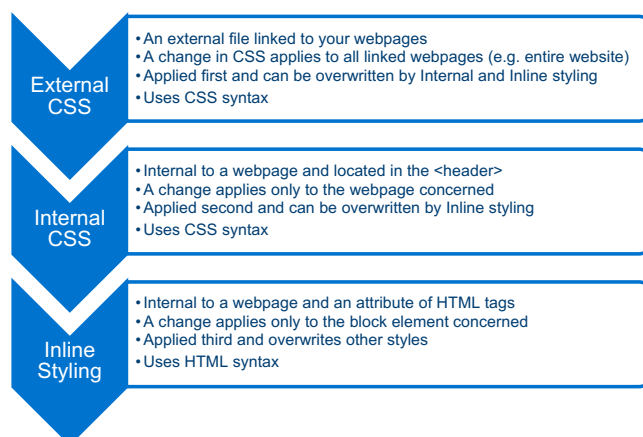
Browser default

External style sheet

Internal style sheet (in <head> section)

Inline style (inside an HTML element)

Three ways to style



Externally Linked CSS – for styling the whole site

You should **<link>** to a style sheet within the **<head>** section of a webpage. The link should contain four attributes:

- rel (relationship) will be “stylesheet”
- type (what is being styled) will be “text/css”
- media (the media/device being targeted) screen, print, speech
- href (location) will be where to locate the stylesheet

<link rel="stylesheet" type="text/css" media="screen" href="css/screen.css" >

- The style sheet should not contain any html tags
- The style sheet file must be saved with a .css extension

Internal CSS – for styling a webpage

Included within the **<head>** section of the html document and used to target a specific element within a specific webpage.

```
<head>
<title>My Interests</title>
<style type="text/css">
  h1 {font-size: 1em; color: black;}
  h2 {font-style: italic; color: red;}
</style>
</head>
```

Overrides External CSS stylesheet

CSS Syntax

https://www.w3schools.com/css/css_syntax.asp

```
selector {property: value; property2: value;}
```

Selectors can be tags

```
h1 {color: red; background: black;}
```

Or a tag with an ID

```
h3.titletext {  
  color: #fff;  
  letter-spacing: 3px;  
  font-size: 2em;  
  font-weight: bold; }
```

You can use multiple selectors or nested selectors

CSS Syntax

https://www.w3schools.com/css/css_syntax.asp

CSS Comments

```
/* This is a comment */
```

Comments can be added to the CSS document and are rendered green by Notepad++.

Handy if you want to maintain a record of what each piece of code does, either for you or for somebody else who may need to make changes later.

Common CSS Properties – Text & Font

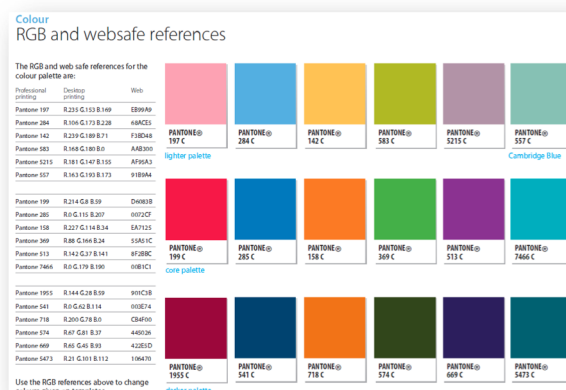
Text color `h1 {color: red;}`
 `h1 {color: #ff0000;}`

A hexadecimal color is specified with: #RRGGBB, where the RR (red), GG (green) and BB (blue) hexadecimal integers specify the components of the color. All values must be between 0-9 and A-F.

Text alignment `h1 {text-align: center;}`
Text decoration `h2 {text-decoration: underline;}`
Text transform `h3 {text-transform: lowercase;}`

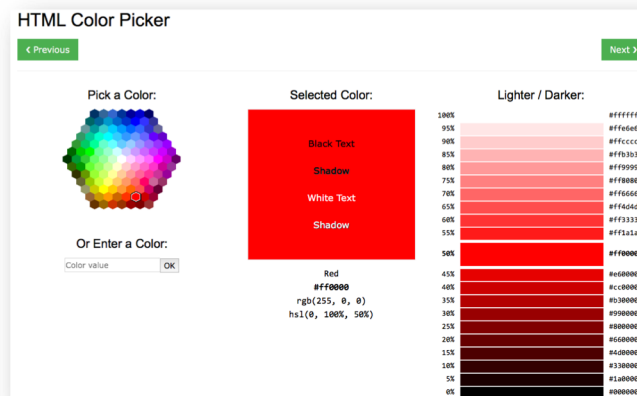
Common CSS Properties – Hexadecimal Colours

Guidelines for using the Cambridge University Colour palette can be found at
<https://www.cam.ac.uk/brand-resources/>



Common CSS Properties – Hexadecimal Colours

https://www.w3schools.com/colors/colors_picker.asp



Common CSS Properties – RGB Colours

RGB short for Red, Green and Blue

<https://www.w3schools.com>

```
<!DOCTYPE html>
<html>
<style>
div {
  background-color: rgb(0, 191, 255);
  color: rgb(255, 255, 255);
  padding: 20px;
}
</style>
<body>

<div>

<h1>London is the capital city of England.</h1>
<p>It is the most populous city in the United Kingdom,
with a metropolitan area of over 13 million inhabitants.</p>

</div>
</body>
</html>
```

London is the capital city of England.

It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.

Common CSS Properties – The RGB Alpha channel

rgba short for red, green, blue and alpha

Alpha defines the opacity as a number between 0.0 (fully transparent) and 1.0 (fully opaque)

```
<!DOCTYPE html>
<html>
<head>
</head>

<body>
<p>RGB colors with opacity:</p>
<p id="p1">Red</p>
<p id="p2">Green</p>
<p id="p3">Blue</p>
<p id="p4">Grey</p>
<p id="p5">Yellow</p>
<p id="p6">Cerise</p>
</body>
</html>
```

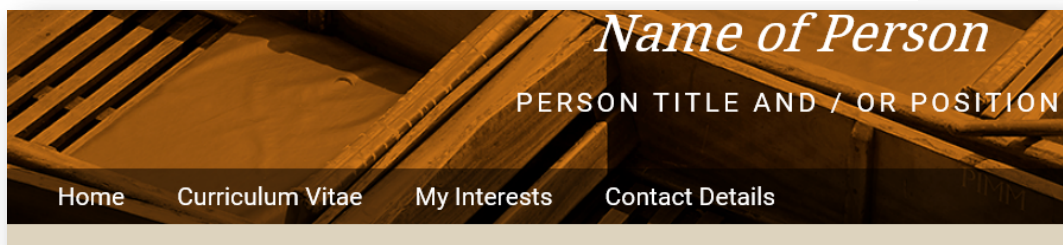
```
<style>
#p1 {background-color:rgba(255,0,0,0.3);}
#p2 {background-color:rgba(0,255,0,0.3);}
#p3 {background-color:rgba(0,0,255,0.3);}
#p4 {background-color:rgba(192,192,192,0.3);}
#p5 {background-color:rgba(255,255,0,0.3);}
#p6 {background-color:rgba(255,0,255,0.3);}
</style>
```

RGB colors with opacity:



Common CSS Properties – The RGB Alpha channel

```
nav {
  background-color: rgba(0,0,0,.50);
  width: 100%;
  position: absolute; top: 247px;
}
```



Common CSS Properties – Class Selectors

Class Selectors

Attribute selector is based on class attributes applied to HTML elements.

HTML	<code><h2 class="subheading">Item two</h2></code>
CSS	<code>.subheading {color:blue;}</code>

Class Selectors can be used to identify more than one element.

ID Selectors

Attribute selector is based on ID attributes applied to HTML elements.

HTML	<code><div id="sidebar">...content...</div></code>
CSS	<code>#sidebar {font-size:80%;}</code>

ID Selectors can only identify one.

Common CSS Properties – Class Selectors

```
<!DOCTYPE html>
<html>
<head>

<style>
p.center {
  text-align: center;
  color: red;
}
p.large {
  font-size: 300%;
}
</style>

</head>

<body>

<h1>This heading will not be affected</h1>
<p class="center">This paragraph will be red and center-aligned.</p>
<p class="center large">This paragraph will be red, center-aligned, and in a large
font-size.</p>

</body>
</html>
```

This heading will not be affected

This paragraph will be red and center-aligned.

**This paragraph will be red, center-aligned,
and in a large font-size.**

Common CSS Properties – ID Selectors

```
<!DOCTYPE html>
<html>
<head>
<style>
#firstname {
  background-color: yellow
}
#center {
  text-align: center
}
#hometown {
  font-size: 20px
}
</style>
</head>
<body>

<h1 id="center">Welcome to My Homepage</h1>

<div class="intro">
  <p id="firstname">My name is Donald.</p>
  <p id="hometown">I live in Duckburg.</p>
</div>

<p>My best friend is Mickey.</p>

</body>
</html>
```

Welcome to My Homepage

My name is Donald.

I live in Duckburg.

My best friend is Mickey.

Common CSS Properties – Font

CSS font properties define the font family, boldness, size and the style of text.

Generic families:



Specific families: Times New Roman (Serif)
Arial (Sans-Serif)

Common CSS Properties – Font

Font Family: `p {font-family: "Times New Roman", verdana, serif;}`

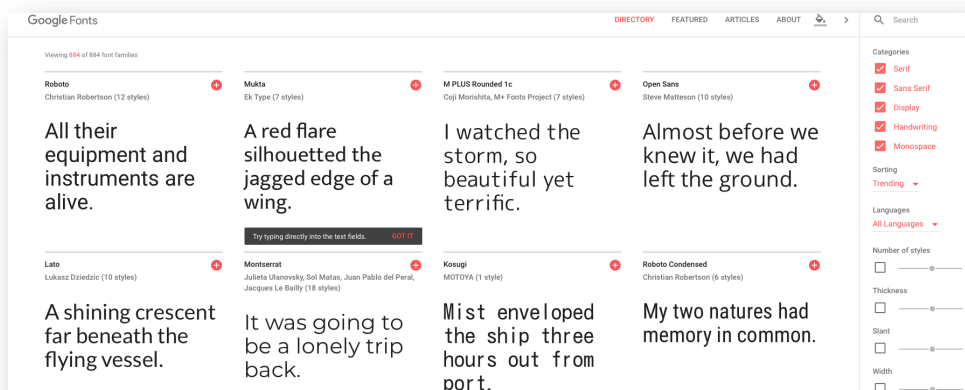
Use several font names as a “fallback” approach

Font Style: `p {font-style: italic;}`

Font Size: `p {font-size: 14px;}`

Common CSS Properties – Font

Google fonts <https://fonts.google.com>



Common CSS Properties – Font

Google fonts <https://fonts.google.com>

Choose fonts then copy and paste given code into the top of CSS document

A screenshot of a code editor with two tabs: 'index.html' and 'screen.css'. The 'screen.css' tab is active, showing the following CSS code:

```
1 @import url('https://fonts.googleapis.com/css?family=Noto+Serif|Roboto');
2
3 body { font-family: 'Noto Serif', serif, 'Roboto', sans-serif; font-size: 16px;
4
```

Common CSS Properties – Font

What is an em?

- An **em** is short for the letter **M**.
- A letter **M** is the biggest sized letter in a font.
- We can use **em's** to scale our font relative to the base font.
- It's a more modern way of scaling fonts and adjusting space around fonts.
- Works better with modern browsers, especially when it come to responsive web design.

Common CSS Properties – Font

What is an em?

Our base font size is 16px

```
index.html | screen.css
1 @import url('https://fonts.googleapis.com/css?family=Noto+Serif|Roboto');
2
3 body { font-family: 'Noto Serif', serif, 'Roboto', sans-serif; font-size: 16px;
4
h1 {margin: 0 0 1em 0; font-family: 'Noto Serif'; font-size: 2.5em; font-weight: 500; line-height: 1.1em;}
h2 {margin: 0 0 .5em 0; font-family: Roboto; font-size: 2em; font-weight: 500; line-height: 1.1em;}
h3 {margin: 0 0 .5em 0; font-family: Roboto; font-size: 1.3em; font-weight: 300;}
h4 {margin: 0 0 1.5em 0; font-family: Roboto; font-size: 1.1em; font-weight: 500; }
p {margin: 0 0 1em 0; font-family: 'Noto Serif'; font-size: 1.1em; }
a {font-family: Roboto; font-size: 1.2em;}
```

1em = 16px

2em = 32px

.5em = 8px

Common CSS Properties – Font

```
<!DOCTYPE html>
<html>
<head>
<style>
p.a {
  font-family: "Times New Roman", Times, serif;
}

p.b {
  font-family: Arial, Helvetica, sans-serif;
}
</style>
</head>
<body>
<h1>The font-family Property</h1>

<p class="a">This is a paragraph, shown in the Times New Roman font.</p>

<p class="b">This is a paragraph, shown in the Arial font.</p>

</body>
</html>
```

The font-family Property

This is a paragraph, shown in the Times New Roman font.

This is a paragraph, shown in the Arial font.

Common CSS Properties – Font

HTML

```
<footer>
  <p>Last updated August 2018
    <br>
    By Simon Meaker
    <br>
    University Information Services
  </p>
</footer>
```

Last updated August 2018
By Simon Meaker
University Information Services

CSS

```
footer {width: 100%; height: 100px; background-color: #000; text-align: center;}

footer p {font-size: 0.8em; color: #fff; padding-top: 20px;}
```

Common CSS Properties – Background

Background colour:

University of Cambridge

HTML

```
<body>
  <h1>University of Cambridge</h1>
</body>
```

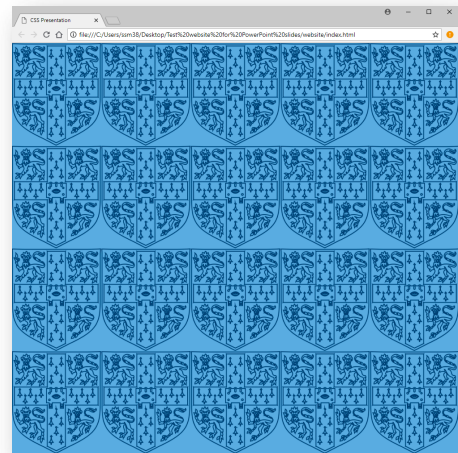
CSS

```
h1 {
  background-color: red;
  color: yellow; }
```

Common CSS Properties – Background

Body background image:

```
body {  
  background-image: url(../images/camshielddtile.png);  
  background-repeat: repeat;  
  background-size: 200px; }
```



Common CSS Properties – Background

Section background image:

HTML

```
<section class="kingspicture"></section>
```

CSS

```
section.kingspicture {  
  width: 100%;  
  height: 500px;  
  background-image: url(../images/kingscollege.jpg);  
  background-size: 800px auto;  
  background-repeat: no-repeat;  
  background-position: center; }
```



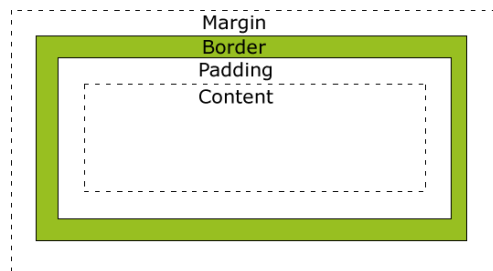
Common CSS Properties – Box Model

Content - where text and images appear

Padding – a transparent area around the content

Border - A border goes around the padding and content

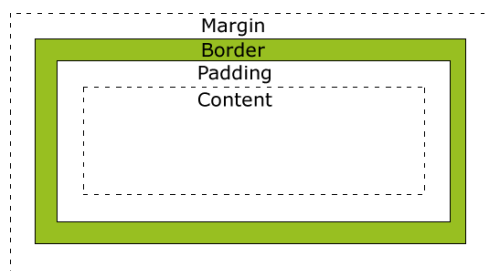
Margin - a transparent area around the border



Common CSS Properties – Box Model

Example:

```
div {  
  width: 500px;  
  padding: 35px;  
  border: 10px solid green;  
  margin: 35px;  
}
```



Common CSS Properties – Box Model

The box model allows us to add a border around elements and to define space between elements.

Important: When you set the width and height properties of an element with CSS, you just set the width and height of the content area. To calculate the full size of an element, you must also add padding, borders and margins.

To style a <div> element to have a total width of 350px:

```
div {  
  width: 320px;  
  padding: 10px;  
  border: 5px solid gray;  
  margin: 0;  
}
```

Common CSS Properties – Box Model

Setting the margin:

Margin: Top Right Bottom Left; `margin: 20px 0 50px 0;}`

Margin: Top & Bottom Right & Left; `margin: 20px 0 ;}`

Margin: All round; `margin: 20px;}`

Common CSS Properties – Positioning

www.w3schools.com/css/css_positioning.asp

Static Positioning – rarely used because not needed.

By default a block element is positioned according to the *normal flow* of the page.

```
<!DOCTYPE html>
<html>
<head>
<style>
div.static {
  position: static;
  border: 3px solid #73AD21;
}
</style>
</head>
<body>

<h2>position: static;</h2>

<p>An element with position: static; is not positioned in any special way; it is
always positioned according to the normal flow of the page:</p>

<div class="static">
This div element has position: static;
</div>

</body>
</html>
```

position: static;

An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page:

This div element has position: static;

Common CSS Properties – Positioning

www.w3schools.com/css/css_positioning.asp

Relative Positioning

A relative positioned block element is positioned relative to its normal position.

```
<!DOCTYPE html>
<html>
<head>
<style>
div.relative {
  position: relative;
  left: 30px;
  border: 3px solid #73AD21;
}
</style>
</head>
<body>

<h2>position: relative;</h2>

<p>An element with position: relative; is positioned relative to its
normal position:</p>

<div class="relative">
This div element has position: relative;
</div>

</body>
</html>
```

position: relative;

An element with position: relative; is positioned relative to its normal position:

This div element has position: relative;

Common CSS Properties – Positioning

www.w3schools.com/css/css_positioning.asp

Fixed Positioning – fairly rare use

A block element with fixed position is positioned relative to the browser window.

It will not move even if the window is scrolled. Top, right, bottom, and left properties are used.

```
<!DOCTYPE html>
<html>
<head>
<style>
div.fixed {
  position: fixed;
  bottom: 0;
  right: 0;
  width: 300px;
  border: 3px solid #73AD21;
}
</style>
</head>
<body>

<h2>position: fixed;</h2>

<p>An element with position: fixed; is positioned relative to the viewport,
which means it always stays in the same place even if the page is scrolled:</p>

<div class="#fixed">
  This div element has position: fixed;
</div>

</body>
</html>
```

position: fixed;

An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled:

This div element has position: fixed;

Common CSS Properties – Positioning

www.w3schools.com/css/css_positioning.asp

Absolute Positioning

An element with position: absolute; is positioned relative to the nearest positioned ancestor (if we have a div within a div).

However; if an absolute positioned element has no positioned ancestors, it uses the document body.

This means it is given an absolute position relative to the top left corner of the webpage and then moves along with page scrolling.

Common CSS Properties – Positioning

www.w3schools.com/css/css_positioning.asp

Absolute Positioning

```
<!DOCTYPE html>
<html>
<head>
<style>
div.relative {
  position: relative;
  width: 400px;
  height: 200px;
  border: 3px solid #73AD21;
}
div.absolute {
  position: absolute;
  top: 80px;
  right: 0;
  width: 200px;
  height: 100px;
  border: 3px solid #73AD21;
}
</style>
</head>
<body>

<h2>position: absolute;</h2>

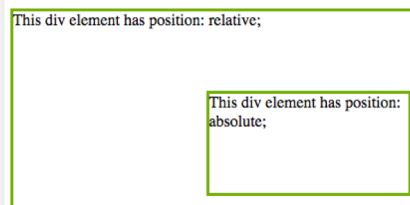
<p>An element with position: absolute; is positioned relative to the nearest positioned
ancestor (instead of positioned relative to the viewport, like fixed):</p>

<div class="relative">This div element has position: relative;
  <div class="absolute">This div element has position: absolute;</div>
</div>

</body>
</html>
```

position: absolute;

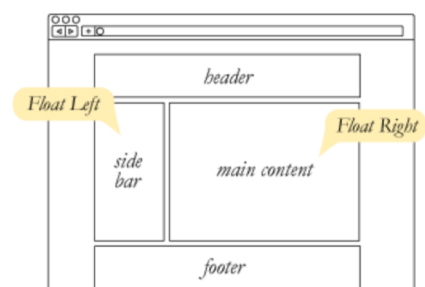
An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed):



Common CSS Properties

float img {float: right;}

- Block elements (e.g. <p>, <h1>, , , <hr /> and <div>) can be floated horizontally (not vertically)
- A floated element will move as far to the left or right as it can
- The elements after the floating element will flow around it.
- The elements before the floating element will not be affected.
- Images can also float.



```
img {
  float: right;
}
```

Common CSS Properties

`float: right;`

```
img {  
  float: right;  
}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac...



```
img {  
  float: left;  
}
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac...

```
img {  
  float: none;  
}
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla

et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac...

Common Image Properties

- Images need to be resized using pixels as the unit of measure
- Images must be resized at a resolution of 72ppi (pixels per inch)
- The image colour mode must be RGB
- Save As Jpeg or PNG
- Photoshop is the industry standard
- Pixlr is a good online option <https://pixlr.com/editor/>

Personal webpages

- University Information Services provides a facility called DS-Web which can be used to host your personal website or webpages. Full details are available on:

<http://www.cam.ac.uk/cs/desktop-services/ds-web/>

- To publish your own pages simply copy your HTML files and associated content to the *public_html* folder in your DS-Files file space.
- To view the web-page navigate to the link below replacing CRSid with your own

<http://people.ds.cam.ac.uk/CRSid/>

Over to you...

The rest of this session will be dedicated to a self-paced practical which will involve styling a basic website using the concepts introduced in this talk.

All instructions are provided in the practical course notes but should you get stuck, find something unclear or have any question about the practical just ask.

Where do you go next? Learn JavaScript:

<http://www.htmldog.com/guides/javascript/>

<https://www.w3schools.com/js/default.asp>