



# School of Computing, Maths & Digital Technology

## **6G4Z2101: Introduction to Web Design and Development**

---

## CSS Layout and Position

### Styling lists



# Recap of last week

---

- The syntax for CSS is that styles include a selector, a property and a value
- Properties and values are paired
- A single selector can have many properties and values

```
selector {propertyA:valueA;  
          propertyB:valueB}
```

```
h1 {font-size:20px;  
    color: #FF3399}
```

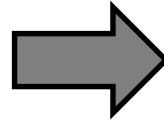
# Recap of last week

---

- Standard html tags are reference in style sheets simply by their tag name (e.g. p, h1, body)
- ids are referenced in style sheets using *#id\_name*
- Classes are reference in style sheets using *.class\_name*

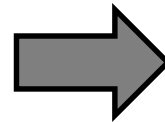
# Contextual selectors

```
#mainContent p {  
    color:#660066;  
    font-size:12px;  
}
```



This would style the paragraph text appearing within the div with the id “mainContent”

```
nav a {  
    color:#666666;  
    text-decoration:none;  
    font-weight:bold;  
}
```



This would style links that appear within the area called “nav”

# Multiple selectors in one statement

- Selectors can be grouped to avoid repetition
  - e.g Centre all headings

```
h1, h2, h3 {  
    text-align: center;  
}  
  
#mainContent, #sidebar {  
    padding: 10px  
}
```

- You can still define other properties for the individual headers, using separate declarations

# CSS Reset

- Browsers can add default margins and/or padding to some html elements
- Good practice to reset them by setting to zero at the beginning of the style sheet

```
html, body, div, span, h1, h2, h3, h4,  
h5, h6, p, ol, ul, li, table, caption, tr,  
th, td, article, aside, canvas, footer,  
header, menu, nav, section {  
    margin: 0;  
    padding: 0;  
    border: 0;  
}
```

# Layout options

- **Fixed** - element specified using pixel values e.g. width: 1000px;
  - Layout tightly controlled/more stable
  - Good for more complex layouts
  - Can create horizontal scroll bars on smaller display screens
- **Fluid/Liquid** – element specified as relative to the browser window, using percentages e.g. width: 70%;
  - Adapts to different viewport sizes
  - Layout can break
  - Images can stretch and lose quality if size is defined in %
- **Elastic** – elements specified in ems e.g. width: 75em; adapts to change in base font size
  - Good for text and text spacing but can be difficult to get right for layout/positioning
- **Responsive** — use HTML, CSS and media queries to enable content to resize, hide, or move, adapting to different size screens



# % vs px values



Navigation and content div widths are specified in % values



Navigation and content div widths are specified in px values



# px and % combined

## BubbleUnder.com

*Diving club for the south-west UK - let's make a splash!*

- [Home](#)
- [About Us](#)
- [Contact Us](#)

### About Us

BubbleUnder is a group of diving enthusiasts based in the south-west UK who meet up for diving trips in the summer months when the weather is good and the bacon rolls are flowing. We arrange weekends away as small groups to cut the costs of accommodation and travel and to ensure that everyone gets a trustworthy dive buddy.

Although we're based in the south-west, our previous weekends away have included...

## BubbleUnder.com

*Diving club for the south-west UK - let's make a splash!*

- [Home](#)
- [About Us](#)
- [Contact Us](#)

### About Us

BubbleUnder is a group of diving enthusiasts based in the south-west UK who meet up for diving trips in the summer months when the weather is good and the...

Navigation width is in px,  
but the content uses %

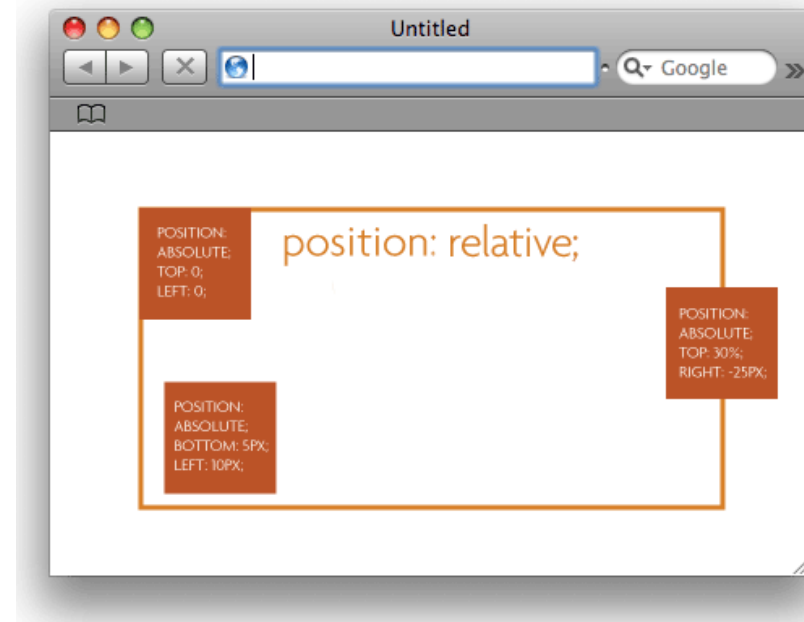
# CSS position

- Last week looked at using float property to position content
  - Can break layout as floated elements are removed from normal document flow
- Alternative is to use position statements
  - Static (default)
  - Fixed
  - Relative
  - Absolute

```
#content {  
    position: absolute;  
    top: 10px;  
    left: 10px;  
}
```

# Relative position

- Normal document flow
- Positions content relative to its container
- Use offsets to position elements (top, left, bottom, right)
- Other element positions not affected



# Absolute Positioning

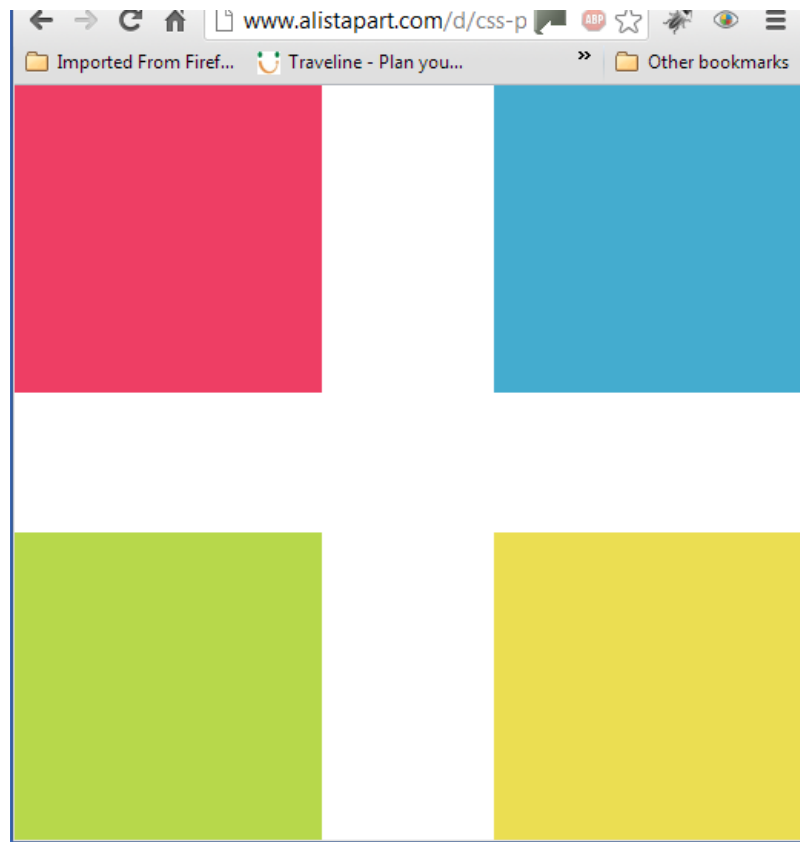
- Breaks normal document flow
- Specify a position using top, bottom, left and right pixel values
- How position is calculated will depend on the position of containing element

```
{  
  position: absolute;  
  top: 20px;  
  left: 20px;  
  width: 100px;  
}
```

You could position another div next to this by using the same top margin and a wider left margin

# Absolute positioning

- All absolute zero positions



How might this code be shortened?

```
#box_1 {  
  position: absolute;  
  top: 0;  
  left: 0;  
  width: 200px;  
  height: 200px;  
  background: #ee3e64;  
}  
#box_2 {  
  position: absolute;  
  top: 0;  
  right: 0;  
  width: 200px;  
  height: 200px;  
  background: #44accf;  
}  
#box_3 {  
  position: absolute;  
  bottom: 0;  
  left: 0;  
  width: 200px;  
  height: 200px;  
  background: #b7d84b;  
}  
#box_4 {  
  position: absolute;  
  bottom: 0;  
  right: 0;  
  width: 200px;  
  height: 200px;  
  background: #ebde52;  
}
```

# Fixed position

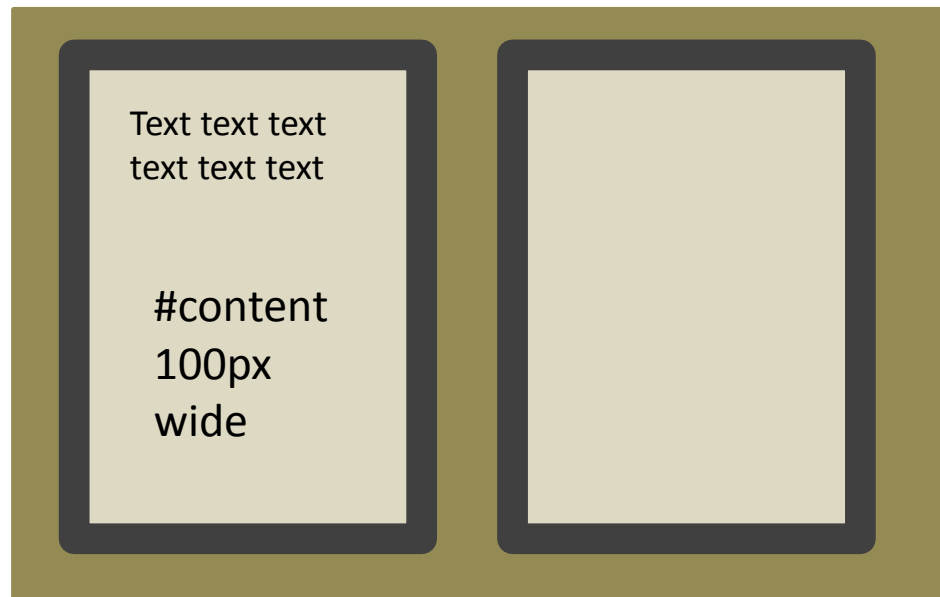
- Elements can be positioned in a fixed location, so they do not move as you scroll
- Can be useful for keeping navigation menu static
- [Example](#) of fixed position

```
footer {  
  position: fixed;  
  bottom: 0;  
  left: 0;  
  height: 70px;  
  background-color: white;  
  width: 100%;  
}
```

# Calculating positions

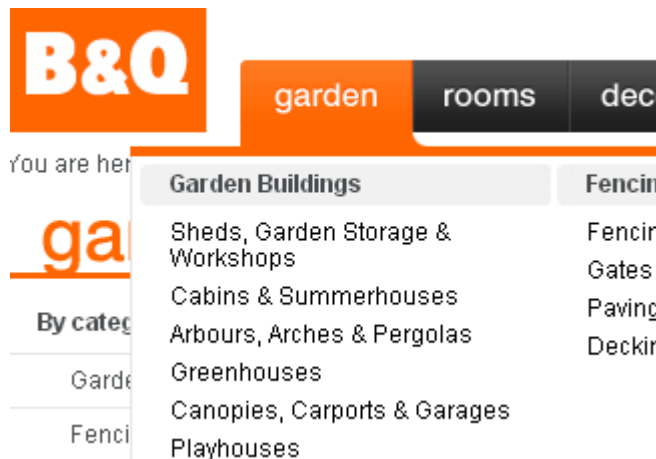
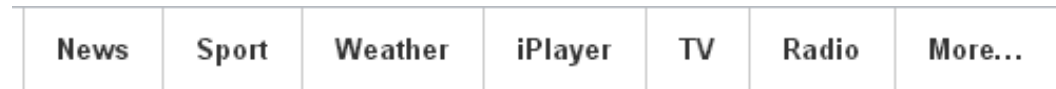
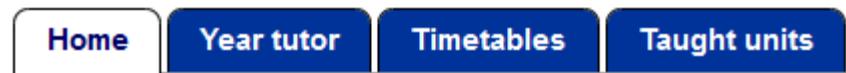
- Not always calculated how you might expect
- Borders and padding are **added to** the dimensions of the element
- Overall width of #content would be 100px + 2 borders of 20px each + padding of 20px on each side

border and  
padding 20px  
wide



# Formatting lists of links

- Navigation menus are often lists
- CSS controls display
- Flexibility



## HTML Tutorials

Learn HTML  
Learn HTML5  
Learn CSS  
Learn CSS3

## Browser Scripting

Learn JavaScript  
Learn HTML DOM  
Learn jQuery  
Learn AJAX  
Learn JSON  
Learn Google Maps

## Server Scripting

## Radio 1

1Xtra

Radio 2

Radio 3

Radio 4

Radio 4  
Extra

5 live

5 live  
sports extra



# The default list

- As seen in week 1, lists have default browser styling

```
<ul>
  <li>Check emails</li>
  <li>Put assignment deadlines in calendar </li>
  <li>Plan Web assignment </li>
  <li>Read book chapter </li>
</ul>
```

- Check emails
- Put assignment deadlines in calendar
- Plan Web assignment
- Read book chapter

```
<ol>
  <li>Check emails</li>
  <li>Put assignment deadlines in calendar </li>
  <li>Plan Web assignment </li>
  <li>Read book chapter </li>
</ol>
```

1. Check emails
2. Put assignment deadlines in calendar
3. Plan Web assignment
4. Read book chapter

# Default styles

- Remove browser defaults for margin & padding
- By default, browsers indent lists with padding
- You can remove the bullet icon using the **list-style-type** property

## Links

- [Webcredible](#)
- [Usability.gov](#)
- [Useit.com](#)
- [W3C Accessibility](#)
- [RNIB](#)
- [WebAIM](#)

default padding

## Links

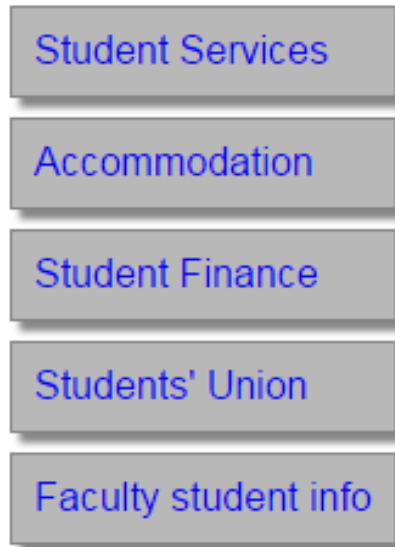
- [Webcredible](#)
- [Usability.gov](#)
- [Useit.com](#)
- [W3C Accessibility](#)
- [RNIB](#)
- [WebAIM](#)

0 padding

```
ul {  
    padding: 0;  
    list-style-type: none;  
}
```

# Vertical navigation bar

- Lists can easily be turned into navigation bars



```
nav ul {  
    padding: 1em;  
    list-style-type: none;  
    width: 90%;  
    background-color: #ECFCFD;  
    border: 1px solid #006;  
}
```

```
nav li {  
    padding-left: 2em;  
    display: inline;  
}
```

```
nav a { text-decoration: none; }
```

Use the 'text-decoration' property to remove the default underline from links

# Horizontal navigation lists

- The same principal as vertical lists, but you need some additional code, and formatting usually needs adjusting
- By default, list items are block level elements, with each list item taking up the full width of its container, but display can be changed to 'inline'



The 'inline' value for the 'display' property makes block level element behave as if it were an inline element

```
nav li { display: inline; }
```

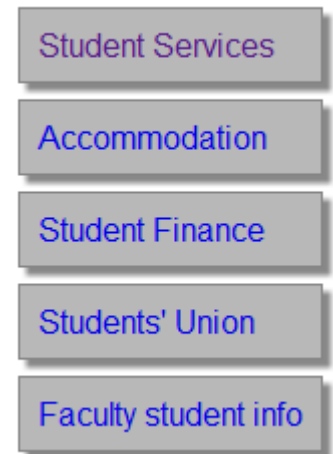
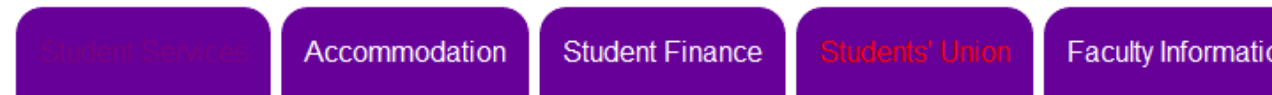
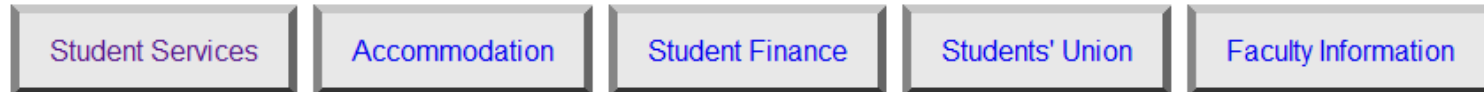
# Block vs Inline elements

---

- Every element in HTML has a default display value, which tends to be either 'block' or 'inline'.
- Block-level elements start on a new line and take up the full width available to them
  - e.g. <div>, <h?>, <p>, <header>, <footer>, <section>
- Inline elements do not start on a new line, and only take up as much width as necessary
  - e.g. <a>, <img>, <span>
- CSS can be used to change the display from the default display for the element

# CSS flexibility

- Changes in the css code can substantially alter the appearance of the same list of links



# Styles to help the user

---

- Show which link a user is hovering over, which link they have visited, and which link is for the current page

```
nav a:link {color:blue;}
```

```
nav a:visited {color:green;}
```

```
nav a:hover {color:red;}
```

```
nav a:active {color:yellow;}
```

The hover action can be more effective if you also change the background colour (depending on how your links are styled)

# CSS position – further info

---

- <http://www.barelyfitz.com/screencast/html-training/css/positioning/>
- <http://alistapart.com/article/css-positioning-101>
- <http://www.brainjar.com/css/positioning/>