# COMP30680
# Web Application Development

## CSS part 2

David Coyle
d.coyle@ucd.ie

# CSS Properties

CSS provides a large number of properties for you to use.

## CSS Property Groups

- Color
- Background and Borders
- Basic Box
- Flexible Box
- Text
- Text Decoration
- Fonts
- Writing Modes
- Table
- Lists and Counters
- Animation
- Transform
- Transition
- Basic User Interface
- Multi-column
- Paged Media
- Generated Content
- Filter Effects
- Image/Replaced Content
- Masking
- Speech
- Marquee

# This week

The aim today is to consider examples of the use of CSS to manage the style and layout of websites. This also provides a context to look at a range of CSS properties.

**Examples are taken from w3schools as follows:**

CSS Box Model: http://www.w3schools.com/css/css_boxmodel.asp

CSS Links: http://www.w3schools.com/css/css_link.asp

CSS Tables: http://www.w3schools.com/css/css_table.asp

CSS Navigation Bar: http://www.w3schools.com/css/css_navbar.asp

Today

CSS positioning

- Display: http://www.w3schools.com/css/css_display_visibility.asp
- Position: http://www.w3schools.com/css/css_positioning.asp
- Float: http://www.w3schools.com/css/css_float.asp
- Inline block: http://www.w3schools.com/css/css_inline-block.asp

# Box Model


In CSS, the term "box model" is used when talking about design and layout.

The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content.

**Content** - The content of the box, where text and images appear

**Padding** - Clears an area around the content. The padding is transparent

**Border** - A border that goes around the padding and content

**Margin** - Clears an area outside the border. The margin is transparent

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

See box_model.html

# Box Model


The **total width** of an element should be calculated like this:

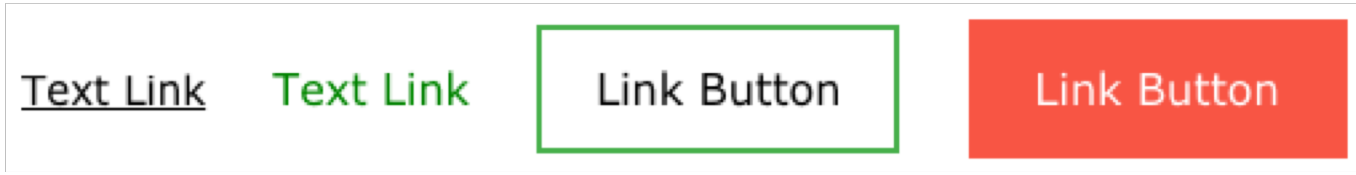
Total element width = width + left padding + right padding + left border + right border + left margin + right margin

The **total height** of an element should be calculated like this:

Total element height = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin

**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.

# Links




Links can be styled with any CSS property (e.g. color, font-family, background, etc.). See css_links.html.

In addition, links can be styled differently depending on what **state** they are in. The four links states are:

- **a:link** - a normal, unvisited link
- **a:visited** - a link the user has visited
- **a:hover** - a link when the user mouses over it
- **a:active** - a link the moment it is clicked

When setting the style for several link states, there are some order rules:

- a:hover MUST come after a:link and a:visited
- a:active MUST come after a:hover

By combining CSS link and box model properties you can create buttons. See button_links.html

# Tables

HTML tables can be greatly improved with CSS.

It can be used to control properties including:

borders, width and height, alignment, padding, color etc.

It also includes nice features to create effects like:

| Language | Percentage |
|---|---|
| Python | 65% |
| Java | 19% |
| HTML | 100% |
| CSS | 100% |
| C++ | 24% |
| Lisp | 1% |

- Hoverable tables, by using the :hover selector on <tr> to highlight table rows on mouse over. hover_table.html
- Striped tables, by using the nth-child() selector and adding a background-color to all even (or odd) table rows. stripe_table.html and e5.html
- Responsive table, by using a container element (like ) with overflow-x:auto around the <table> element to make it responsive. respon_table.html

# Navigation bar


Having easy-to-use navigation is important for any web site.

With CSS you can transform boring HTML menus into good-looking navigation bars.

The starting point: **Navigation Bar = List of Links + some CSS styling**

```html
<ul>
  <li><a href="default.asp">Home</a></li>
  <li><a href="news.asp">News</a></li>
  <li><a href="contact.asp">Contact</a></li>
  <li><a href="about.asp">About</a></li>
</ul>
```

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

See basic_nav.html

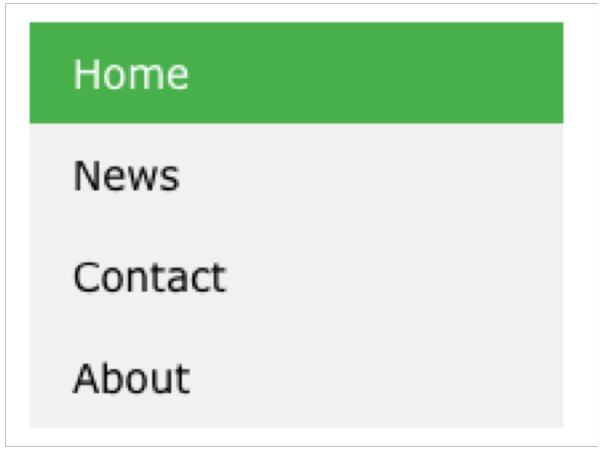
- **list-style-type: none;** - Removes the bullets. A navigation bar does not need list markers
- Set **margin: 0;** and **padding: 0;** to remove browser default settings

# Navigation bar


Add further properties to the list to create a more complex navigation bar: e.g. margin, padding, width, block.

You can also add properties to the links.

**A vertical navigation bar**




This makes use of :

**display: block**; - Displaying the links as block elements makes the whole link area clickable (not just the text), and it allows us to specify the width (and padding, margin, height, etc. if you want)

See vert_nav.html

# Navigation bar


**Horizontal navigation** bars are slightly more tricky.

There are two ways to create a horizontal navigation bar: using **inline** or **floating** list items.

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

Gives you something like this:

Home News Contact About

```
li {
    float: left;
}

a {
    display: block;
    padding: 8px;
    background-color: #dddddd;
}
```

Home News Contact About

# Navigation bar


```
li {
    float: left;
}

a {
    display: block;
    padding: 8px;
    background-color: #dddddd;
}
```


**float: left**; - use float to get block elements to slide next to each other

**display: block**; - Displaying the links as block elements makes the whole link area clickable (not just the text), and it allows us to specify padding (and height, width, margins, etc. if you want)

**padding: 8px**; - Since block elements take up the full width available, specify some padding to make them look good

**background-color: #dddddd**; - Add a gray background-color to each a element

See horz_nav.html

# Navigation bar


Once you begin to combine CSS property on lists and links you can begin build complex navigation bars.

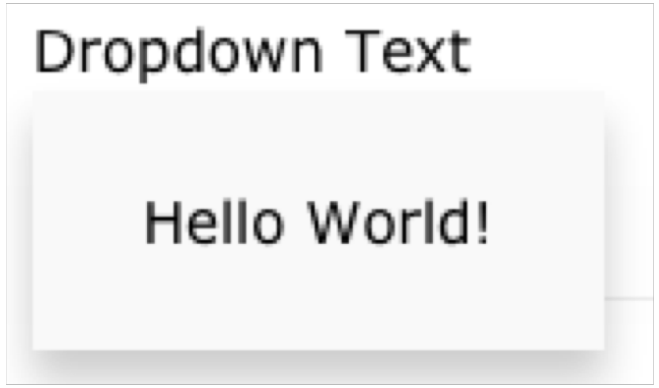

See horz_nav.better.html. This uses a range of properties in both lists and links and also document and inline styles to create a navigation bar with both left and right aligned links.


See also fixed_nav.html. This creates a navigation bar that is fixed to the top of the page as the user scrolls.

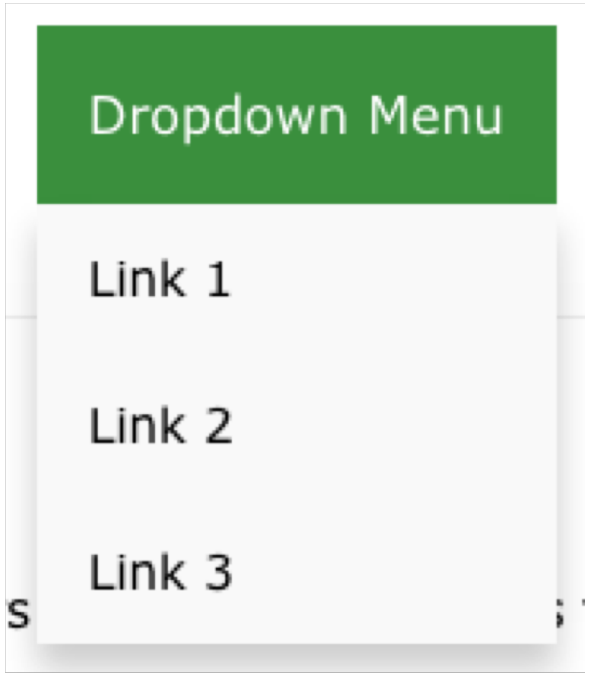
```
ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
    overflow: hidden;
    background-color: #333;
    position: fixed;
    top: 0;
    width: 100%;
}
```

# Dropdown menus

Dropdowns allow you to display text or menus when the user moves the mouse over an element.




basic_dropdown.html




dropdown_menu.html

```
<style>
.dropdown {
    position: relative;
    display: inline-block;
}

.dropdown-content {
    display: none;
    position: absolute;
    background-color: #f9f9f9;
    min-width: 160px;
    box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
    padding: 12px 16px;
    z-index: 1;
}

.dropdown:hover .dropdown-content {
    display: block;
}
</style>
```

See basic_dropdown.html.

The .dropdown class uses position:relative, which is needed when we want the dropdown content to be placed right below the dropdown button.

The .dropdown-content class holds the actual dropdown content and has position:absolute. It is hidden by default, and will be displayed on hover. The :hover selector is used to show the dropdown menu when the user moves the mouse over the dropdown button.

```
<style>
.dropdown {
    position: relative;
    display: inline-block;
}

.dropdown-content {
    display: none;
    position: absolute;
    background-color: #f9f9f9;
    min-width: 160px;
    box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
    padding: 12px 16px;
    z-index: 1;
}

.dropdown:hover .dropdown-content {
    display: block;
}
</style>
```

```
<div class="dropdown">
  <span>Mouse over me</span>
  <div class="dropdown-content">
    <p>Hello World!</p>
  </div>
</div>
```

See basic_dropdown.html.

The .dropdown class uses position:relative, which is needed when we want the dropdown content to be placed right below the dropdown button.

The .dropdown-content class holds the actual dropdown content and has position:absolute. It is hidden by default, and will be displayed on hover. The :hover selector is used to show the dropdown menu when the user moves the mouse over the dropdown button.

In **HTML** Use any element to open the dropdown content, e.g. a , or a <button> element.
Use a container element (like ) to create the dropdown content and add whatever you want inside of it.
Wrap a  element around the elements to position the dropdown content correctly with CSS.

**Tip:** Instead of using a border, this example uses the CSS3 box-shadow property to make the dropdown menu look like a "card".

# CSS Display property

The display property is the most important CSS property for controlling layout. It specifies if/how an element is displayed.

Every HTML element has a default display value depending on what type of element it is. The default display value for most elements is block or inline.

- Examples of block-level elements: , <h1> - <h6>, <p>, <form>, <header>, <footer>, <section>, <li>
- Examples of inline-level elements: , <h1> - <h6>, <p>, <form>, <header>, <footer>, <section>

**Note**: you can override the default, e.g. making inline <li> elements for horizontal menus.

**Display : none**

- Used with dropdown menus
- Also commonly used with JavaScript to hide and show elements without deleting and recreating them. See js.display.html

# **NEXT:** Using CSS to layout elements / pages

CSS can be used in combination with HTML to control the overall layout of elements and pages, and therefore sites.

Common CSS properties used to control layouts include:

Display: http://www.w3schools.com/css/css_display_visibility.asp

Position: http://www.w3schools.com/css/css_positioning.asp

Float: http://www.w3schools.com/css/css_float.asp

Inline block: http://www.w3schools.com/css/css_inline-block.asp

We have seen some examples of this, but in the next class we will examine these in more detail.