CS1115/CS5002

Web Development 1

Dr Derek Bridge

School of Computer Science & Information Technology
University College Cork

Cascading StyleSheets (CSS)

- HTML captures the *structure* of your web page
- But how do you make your web page *look good*?
- Solution: **Cascading StyleSheets** (CSS)
- E.g.

```
p {
    color: red;
}

header {
    color: white;
    background-color: black;
}
```

CSS terminology

- A **CSS stylesheet** contains zero, one or more **rules**
- A **rule** comprises:
  - one or more **selectors** and
  - a block of zero, one or more **declarations** in curly braces
- A **declaration** comprises:
  - a **property** and
  - its **value**, separated by a colon

  and is usually terminated by a semi-colon

```
p {
    color: red;
}

header {
    color: white;
    background-color: black;
}
```

Colours in CSS

- `color` and `background-color` are examples of **properties**
- What **values** are allowed for these two properties?
- Named colours:
  - There are 140 named colours that browsers will understand
  - W3School's list of colour names
- **RGB** values:
  - The amount of red, green and blue light, from 0 to 255
  - E.g.

```
p {
    color: rgb(255, 0, 0);
}
```

  - This allows nearly 17 million colours. **Q**: Why?
  - Computer Scientists prefer to specify RGB values in hexadecimal, e.g.:

```
p {
    color: #FF0000;
}
```

  - Non-Computer Scientists prefer to specify RGB values as percentages, e.g.:

```
p {
    color: rgb(100%, 0%, 0%);
}
```

- Other: RGBA values, HSL values, HSLA values

Class exercise

What colours are these?

1. `rgb(0, 255, 0)`
2. `rgb(255, 255, 255)`
3. `rgb(0, 0, 0)`
4. `rgb(255, 255, 0)`
5. `rgb(255, 250, 0)`
6. `rgb(169, 169, 169)`
7. `rgb(211, 211, 211)`

CSS comments

- A CSS stylesheet can also contain **comments**, e.g.

```
/* This is an example of a comment in CSS */
```

Adding CSS styles to your web pages

- Three ways to add CSS styles to a web page:
  - External stylesheets
  - Embedded stylesheets
  - Inline styles
- We will only use **external stylesheets**

External stylesheets

- Put your CSS rules in a separate file, e.g. `styles.css`
- Link to the file using a `<link>` element in the `head` of your HTML, e.g.

```
<head>
    <meta charset="utf-8" />
    <title>Party Party Party - Cocktails - Mojitos</title>
    <link rel="stylesheet" href="styles.css" />
</head>
```

- **Question**: What kind of element is the `<link>` element
- **Question**: What kind of URL does this example use?
- **Question**: What is the main advantage of putting styles into an external file?
- **Question**: What is the disadvantage of putting styles into an external file?

Embedded stylesheets: the `<style>` element

- You can put the rules into your HTML, inside a `<style>` element, e.g.:

```
<head>
    <meta charset="utf-8" />
    <title>Party Party Party - Cocktails - Mojitos</title>
    <style>
        p {
            color: red;
        }

        header {
            color: white;
            background-color: black;
        }
    </style>
</head>
```

- No need to learn embedded stylesheets: we will use external stylesheets

Inline styles: the `style` attribute

- In HTML, `style` is also a global attribute
  (**Q**: Which other global attributes have we covered?)
- You can put CSS declarations as the value of this attribute, e.g.:

```
<header style="color: white; background-color: black;">
    …
</header>
```

- **Question**: Why do inline styles bloat your web page?
- No need to learn inline styles: we will use external stylesheets

The power of separating structure (HTML) and presentation (CSS)

- Early versions of HTML included some tags and attributes that were *presentational*, rather that *structural*, e.g.
  - Tags: `center, font, big, small, i, b`
  - Attributes: `align, bgcolor, valign,`…
- For HTML5, these tags and attributes have either been deleted or redefined
- So HTML5 is all about *structure*, and we leave *appearance* to CSS
- In the lecture, we will use these three web sites to explore the advantages of this separation:
  - The CSS Zen Garden
  - Derek's web site
  - and a special surprise web site

Inheritance in CSS

- Many CSS properties (but not all) are **inherited**

  inheritance in CSS

  > a property applied to an element is also automatically applied to the element's descendants

- **Class exercise:** What colour will "I like...very much" be? What colour will "Mojitos" be?

```
<p>I like <i>Mojitos</i> very much</p>
```

```
p {
    color: red;
}
```

Blocking inheritance

- The inheritance of a property to a descendant is blocked if some other CSS rule specifies the element's value for that property
- **Class exercise:** What colour will "I like...very much" be? What colour will "Mojitos" be?

```
<p>I like <i>Mojitos</i> very much</p>
```

```
p {
    color: red;
}

i {
    color: blue;
}
```

Not everything is inherited

- Some properties are not inherited
- An intelligent guess usually works
- **Class exercise:** Will the italics be inherited? Will the border be inherited?

```
<p>I like <i>Mojitos</i> very much</p>
```

```
p {
    font-style: italic;
    border: 0.0625em solid black;
}
```

- If you're not sure, you look it up, e.g.
  - Italics: W3C; W3Schools
  - Border: W3C; W3Schools