CSS Selectors

Part one of a three-parter on CSS

```
#header {
    color: blue;
    font-weight: bold;
}
```

Selector

Style attributes

Russell Newman
rn2@ecs.soton.ac.uk

Today

- A plug on why you should use CSS.
- How to get started with CSS.
- How to build HTML that works well with CSS.
- How to build CSS for your HTML.

What's CSS?

It's a file that contains all the style and appearance settings for your site.

By using CSS, you can eliminate style information from your HTML.

- Write *content* in HTML.
- Write *style and appearance* in CSS.

So why use it?

- Eliminates style "noise" from HTML.
- Apply one style to all your pages.
- One file: easy to update.

Find the style “noise” in this HTML

```
<html>
  <head>...</head>
  <body>
      <center><font style="font-size: 20px; color: blue;">A
name you can surely trust!</font></center><br/>
      <hr color="red" /><br/><br/>
      <div style="float: right; width: 350px">
          <img src="img/sales-boyd.png" width="100%">
          <hr color="red" />
          <img src='img/car-blue.png' id="car"
height="256px" /><br/>
      </div>
  </body>
</html>
```

(there are six misdemeanors to find)

Without CSS

```
<html>
    <head>...</head>
    <body>
        <center><font style="font-size: 20px; color: blue;">A name you can
surely trust!</font></center><br/>
        <hr color="red" /><br/><br/>
        <div style="float: right; width: 350px">
            <img src="img/sales-boyd.png" width="100%" />
            <hr color="red" />
            <img src='img/car-blue.png' id="car" height="256px" /><br/>
        </div>
    </body>
</html>
```

With CSS, no “noise”

```
<html>
    <head>...</head>
    <body>
        <p id="strapline">A name you can surely trust!</p>
        <div id="sidebar">
            <img src="img/sales-boyd.png" />
            <img src='img/car-blue.png' />
        </div>
    </body>
```

Getting Started

1. Create an empty CSS file.
2. Link your HTML to the CSS file like this:

```
<html>
    <head>
        <link rel="stylesheet" type="text/css"
href="css/style.css" />
    </head>
    <body>
        ...
    </body>
</html>
```

Classes vs IDs

	Class	*ID*
Usable on separate pages	✓	✓
Use on several elements	✓	✗
Use on unique elements	✗	✓

HTML Genealogy

```
<html>
    <head>...</head>
    <body>
        <p id="strapline">A name you can surely trust!</p>
        <div id="sidebar">
            <img src="img/sales-boyd.png" />
            <img src='img/car-blue.png' />
        </div>
    </body>
</html>
```

Child

HTML Genealogy

```
<html>
    <head>...</head>
    <body>
        <p id="strapline">A name you can surely trust!</p>
        <div id="sidebar">
            <img src="img/sales-boyd.png" />
            <img src='img/car-blue.png' />
        </div>
    </body>
</html>
```

Parent

HTML Genealogy

```
<html>
    <head>...</head>
    <body>
        <p id="strapline">A name you can surely trust!</p>
        <div id="sidebar">
            <img src="img/sales-boyd.png" />
            <img src='img/car-blue.png' />
        </div>
    </body>
</html>
```

Descendant

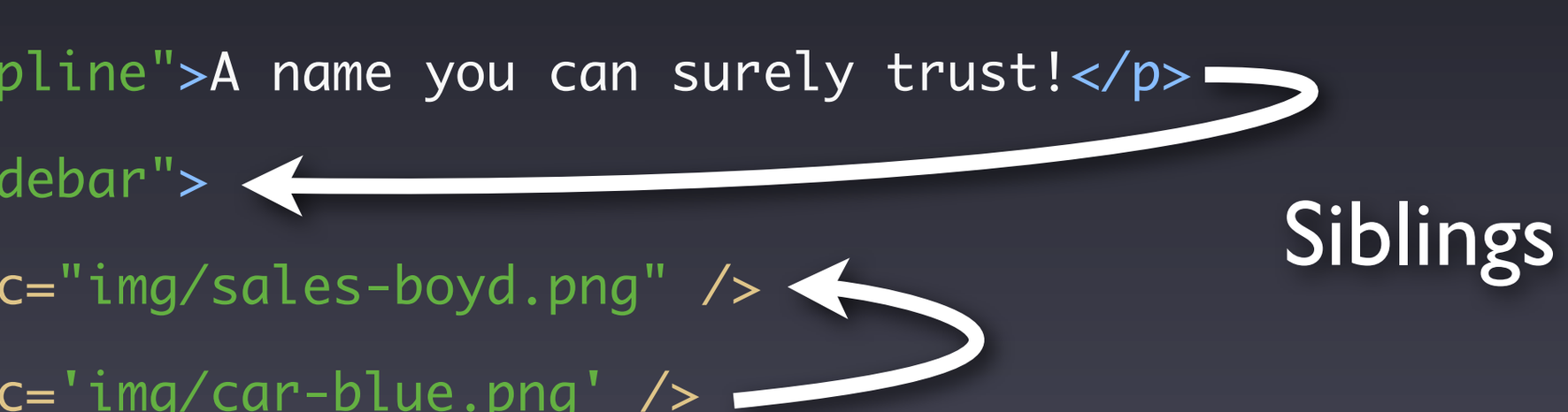
HTML Genealogy

```
<html>
    <head>...</head>
    <body>
        <p id="strapline">A name you can surely trust!</p>
        <div id="sidebar">
            <img src="img/sales-boyd.png" />
            <img src='img/car-blue.png' />
        </div>
    </body>
</html>
```

Ancestor

HTML Genealogy

```
<html>
    <head>...</head>
    <body>
        <p id="strapline">A name you can surely trust!</p>
        <div id="sidebar">
            <img src="img/sales-boyd.png" />
            <img src='img/car-blue.png' />
        </div>
    </body>
</html>
```



This selector...	*Selects...*
#header	The element with `id="header"`
`.article`	All elements with `class="article"`
`img`	All `<img>` elements.

```
<html>
  <head>...</head>
  <body>
      <h1>Site Header</h1>
      <div id="lead">This is the lead article.</div>
      <div id="menu">
          <a class="menu-item">Link 1</a>
          <a class="menu-item">Link 2</a>
          <a class="menu-item">Link 3</a>
      </div>
  </body>
</html>
```

```
<html>
  <head>...</head>
  <body>
      <h1>Site Header</h1>
      <div id="lead">This is the lead article.</div>
      <div id="menu">
          <a class="menu-item">Link 1</a>
          <a class="menu-item">Link 2</a>
          <a class="menu-item">Link 3</a>
      </div>
  </body>
</html>
```

Q: Does the <h1> describe *style* and/or *content*?

Combining Selectors

This selector...	*Selects...*
`h1#header`	The `<h1>` element with `id="header"`
`div.article`	All `<div>` elements with `class="article"`
`div.article.lead`	All `<div>` elements with `class="framed" and class="lead"`

Chaining Selectors

Combining selectors to...

- Target children and descendants.
- Reduce our use of class and id in the HTML.

```
<html>
  <head>...</head>
  <body>
      ...
      <div id="menu">
          <a class="menu-item">Link 1</a>
          <a class="menu-item">Link 2</a>
          <a class="menu-item">Link 3</a>
      </div>
  </body>
</html>
```

```
<html>
  <head>...</head>
  <body>
      ...
      <div id="menu">
          <a class="menu-item">Link 1</a>
          <a class="menu-item">Link 2</a>
          <a class="menu-item">Link 3</a>
      </div>
  </body>
</html>
```

The class attributes on these menu items are repeated, so they are probably candidates for chained CSS.

How can I target them in CSS, without the class attributes?

```
<html>
  <head>...</head>
  <body>
      ...
      <div id="menu">
          <a class="menu-item">Link 1</a>
          <a class="menu-item">Link 2</a>
          <a class="menu-item">Link 3</a>
      </div>
  </body>
</html>
```

The class attributes on these menu items are repeated, so they are probably candidates for chained CSS.

How can I target them in CSS, without the class attributes?

Yes, it's `#menu a`

aka all <a> elements that are children of `id="menu"`

Chaining Selectors

This selector...	*Selects...*
`#content p`	All <p> elements that are children of `id="content"`
`ul#menu li`	All elements that are children of `<ul id="menu">`
`ul#menu li a`	All <a> children of children of `<ul id="menu">`

Attribute Selectors

Attribute Selectors

We can select elements based on other attributes besides id and class.

Attribute Selectors

We can select elements based on other attributes besides id and class.

This selector...	*Selects...*
`input[type="submit"]`	All `<input>` elements with `type="submit"`

Attribute Selectors

We can select elements based on other attributes besides id and class.

```
<form>
    <input type="text" name="email_address" />
    <input type="submit" value="Confirm" />
</form>
```

This selector...	*Selects...*
`input[type="submit"]`	All `<input>` elements with `type="submit"`

Pseudo-selectors

...activate when some operational property of the element is fulfilled.

This psuedo-selector...	*Selects...*
`:hover`	When the element is hovered over.
`:first-child` `:last-child`	The first or last elements inside a parent.
`:first-letter`	The first letter of a block of text.

Pseudo Selectors

Pseudo Selectors

A hover example...

Pseudo Selectors

A hover example...

This selector...	*Selects...*
`a`	All <a> elements.
`a:hover`	An <a> element when hovered over.

Controlling the Ancestors

```
<html>
    <head>...</head>
    <body>
        <div id="header">
            <div>
                <div>Tel: 01234 123456</div>
                <div>Fax: 01234 123457</div>
            </div>
        </div>
    </body>
</html>
```

How can we select *just* the directly inside ?

Controlling the Ancestors

```
<html>
    <head>...</head>
    <body>
        <div id="header">
            <div>
                <div>Tel: 01234 123456</div>
                <div>Fax: 01234 123457</div>
            </div>
        </div>
    </body>
</html>
```

`#header > div` selects `div` children of `#header`.

`#header div` selects `div` descendants of `#header`.

CSS Priorities

Your browser will process CSS styles in this order:

1. Inline styles (e.g. `<a style="color: red">`).
2. Specificity (e.g. `#header img` beats `img`).
3. Order of rules (i.e. last defined rule wins).
4. Browser built-in styles (next lecture!)

   (this is a short list; Wikipedia CSS has a good reference)

CSS Versions

- There are three major versions of CSS.
- They have differing features.
- All backward-compatible.
- Browser support:
  - Most support versions 1 and 2.
  - Some support some features of version 3.

For Reference

1. Google “456 css selectors”
    a. Choose “CSS 2.1 Selectors” for everyday stuff.
    b. Choose “CSS 3 selectors” for fancy stuff.