COMP309
Web-based Technology

Dr. N. B. Gyan

Central University, Miotso. Ghana

CSS (Pt. 1)

Introduction to CSS

- CSS stands for **Cascading Stylesheet**.
- It is used to style and lay out web pages — for example, to alter the font, colour, size and spacing of your content, split it into multiple columns, or add animations and other decorative features.
- CSS is a language for specifying how documents are presented to users – how they are styled, laid out, etc.

What is CSS?

- A *document* is usually a text file structured using a markup language – HTML is the most common markup language, but you will also come across other markup languages such as SVG or XML.
- *Presenting* a document to a user means converting it into a usable form for your audience.

How does CSS affect HTML?

- Web browsers apply *CSS rules* to a document to affect how they are displayed. A CSS rule is formed from:
    - A set of properties, which have values set to update how the HTML content is displayed, for example I want my element's width to be 50% of its parent element, and its background to be red.
    - A **selector**, which *selects* the element(s) you want to apply the updated **property** values to.
    - E.g., *Apply my CSS rule to all the paragraphs in my HTML document.*

A Quick Example

- A set of CSS rules contained within a *stylesheet* determines how a webpage should look.
- Let's take a simple HTML document, containing an <h1> and a <p> (notice that a stylesheet is applied to the HTML using a <link> element):

eg-css-file.html

```
<!DOCTYPE html>
<html>
 <head>
  <meta charset=”utf-8”>
  <title>My CSS experiment</title>
  <link rel=”stylesheet” href=”style.css”>
 </head>
 <body>
  <h1>Hello World!</h1>
  <p>This is my first CSS example</p>
 </body>
</html>
```

A Quick Example: A CSS with two rules

style.css

```css
/* rule 1 */
h1{
   color:blue;
   background-color:yellow;
   border:1px solid black;
}

/* rule 2 */
p {
   color:red;
}
```

A Quick Example. Explanation

- The first rule starts with an h1 selector, which means that it will apply its property values to the <h1> element.
- It contains three properties and their values (each property/value pair is called a *declaration*):
    - The first one sets the text colour to *blue*.
    - The second sets the background colour to *yellow*.
    - The third one puts a border around the header that is *1 pixel* wide, *solid* (not dotted, or dashed, etc.), and coloured *black*.

A Quick Example. Explanation

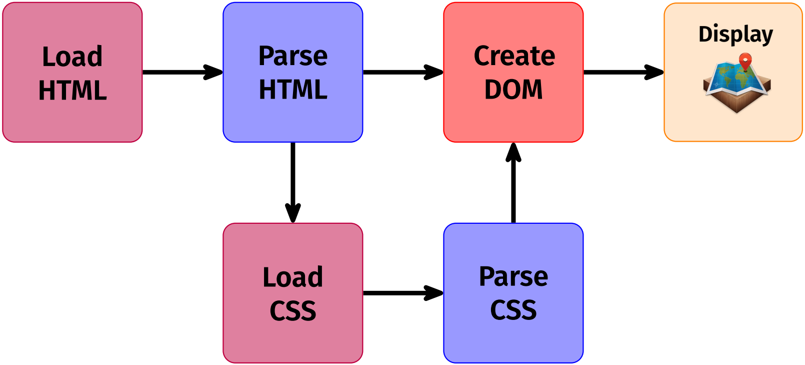
- The second rule starts with a p selector, which means that it will apply its property values to the <p> element. It contains one declaration, which sets the text color to red.

How does CSS actually work?

When a browser displays a document, it must combine the document’s content with its style information. It processes the document in two stages:

1. The browser converts HTML and CSS into the DOM (*Document Object Model*). The DOM represents the document in the computer’s memory. It combines the document’s content with its style.
2. The browser displays the contents of the DOM.

How does CSS actually work?



About the DOM

- A DOM has a *tree*-like structure. Each element, attribute and piece of text in the markup language becomes a DOM **node** in the tree structure.
- The nodes are defined by their relationship to other DOM nodes. Some elements are parents of child nodes, and child nodes have siblings.
- Understanding the DOM helps you design, debug and maintain your CSS because the DOM is where your CSS and the document's content meet up.

DOM representation

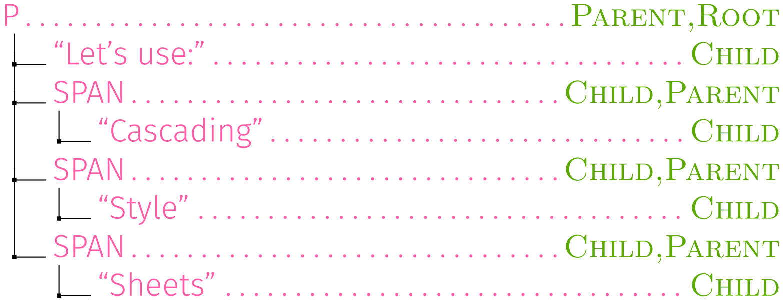
- Rather than a long, boring explanation, let’s take an example to see how the DOM and CSS work together.
- Let’s assume the following HTML code:

```html
<p>
  Let’s use:
  <span>Cascading</span>
  <span>Style</span>
  <span>Sheets</span>
</p>
```

DOM representation

- In the DOM, the node corresponding to our `<p>` element is a parent.
- Its children are a text node and the nodes corresponding to the `<span>` elements.
- The SPAN nodes are also parents, with text nodes as their children:

DOM representation



- P . . . . Parent,Root
  - "Let's use:" . . . . Child
  - SPAN . . . . Child,Parent
    - "Cascading" . . . . Child
  - SPAN . . . . Child,Parent
    - "Style" . . . . Child
  - SPAN . . . . Child,Parent
    - "Sheets" . . . . Child

DOM representation

- This is how a browser interprets the previous HTML snippet - it renders the above DOM tree and then outputs it in the browser like so:

  Let’s use: Cascading Style Sheets

Applying CSS to the DOM

- Let’s say we added some CSS to our document (shown above), to style it.
- Again, the CSS is as follows:

```css
span {
  border:1px solid black;
  background-color:lime;
}
```

Applying CSS to the DOM

- The browser will *parse* the HTML and create a DOM from it, then parse the CSS.
- Since the only rule available in the CSS has a span selector, it will apply that rule to each one of the three spans.
- The updated output is as follows:

Applying CSS to the DOM

Let’s use: Cascading Style Sheets

Providing common styling to table columns

- Back to our tables examples.
- HTML has a method of defining styling information for an entire column of data all in one place – the <col> and <colgroup> elements.
- These exist because it can be a bit annoying and inefficient having to specify styling on columns — you generally have to specify your styling information on every <td> or <th> in the column.
- Take the following simple example:

```
<table>
 <tr>
  <th>Data 1</th>
  <th style=”background-color:yellow;”>
    Data 2
  </th>
 </tr>
 <tr>
  <td>Calcutta</td>
  <td style=”background-color:yellow;”>
   Orange
  </td>
 </tr>
```

```
  <tr>
    <td>Robots</td>
    <td style=”background-color:yellow;”>
      Jazz
    </td>
  </tr>
</table>
```

Data 1	Data 2
Calcutta	Orange
Robots	Jazz

Providing common styling to table columns

- This isn’t ideal, as we have to repeat the styling information across all three cells in the column.
- Instead of doing this, we can specify the information once, on a <col> element.
- <col> elements are specified inside a <colgroup> container just below the opening <table> tag.
- We could create the same effect as we see above by specifying our table as follows:

```
<table>
 <colgroup>
  <col>
  <col style=‘background-color:yellow;’>
 </colgroup>
 <tr>
  <th>Data 1</th>
  <th>Data 2</th>
 </tr>
 <tr>
  <td>Calcutta</td>
  <td>Orange</td>
 </tr>
 <tr>
  <td>Robots</td>
  <td>Jazz</td>
 </tr>
<table>
```

- Effectively we are defining two “style columns”, one specifying styling information for each column.

Providing common styling to table columns

- We are not styling the first column, but we still have to include a blank <col> element – if we didn't, the styling would just be applied to the first column.
- If we wanted to apply the styling information to both columns, we could just include one <col> element with a span attribute on it, like this:

Providing common styling to table columns

```
<colgroup>
  <col
    span=”1”
    style=”background-color:yellow;”>
</colgroup>
```

Providing common styling to table columns

- Just like colspan and rowspan, span takes a *unitless* number value that specifies the number of columns you want the styling to apply to.

Exercise: Creating a Timetable

Do the timetable exercise on vcampus.

See you next week, God willing 🙏🏿