DBI-012
Cascading Style Sheets (CSS)

COMP1048: Databases and Interfaces

Matthew Pike (matthew.pike@nottingham.edu.cn)

Dylan Shen (linlin.shen@nottingham.edu.cn)

Today

- ❓ What is CSS?
- 🏭 How do apply CSS to HTML?
- 🏗️ How should we organise our code?
- ⚙️ What tools do we need?
- 🖍️ By the end of this lecture, you should be able to apply simple style to a HTML page (and understand how it works!)s

Recap: What makes a webpage?

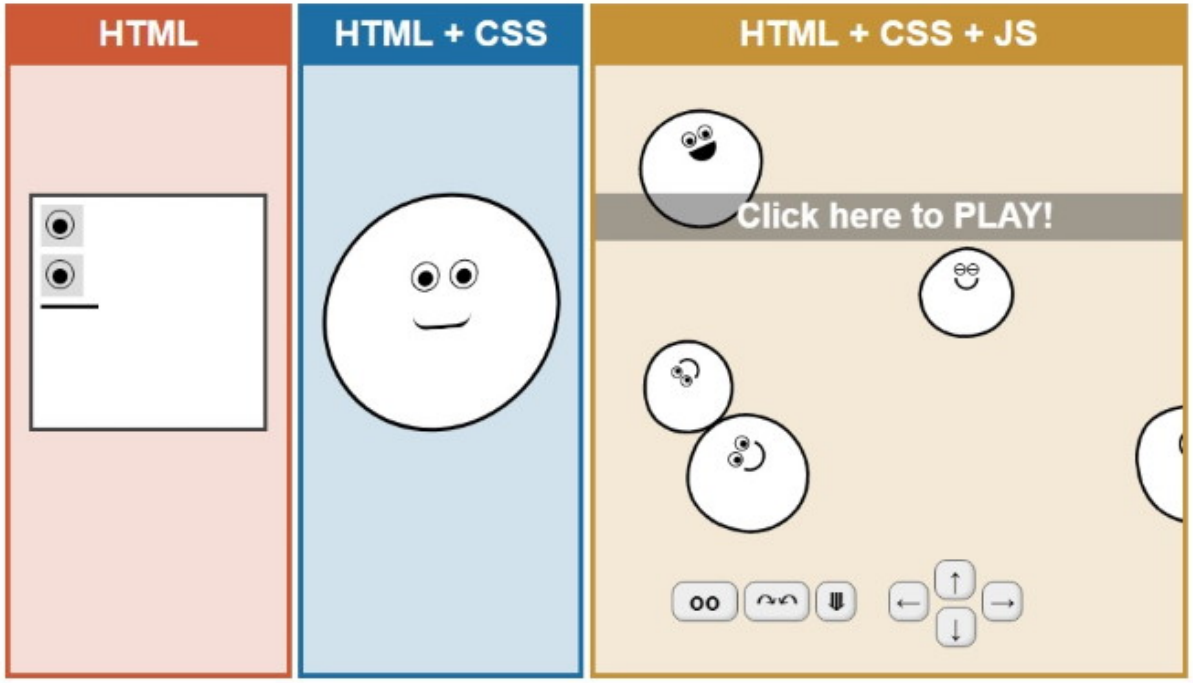

Image Source: https://html-css-js.com/images/og.jpg

What is CSS?

- Using CSS we can control exactly how HTML elements look in the browser, instead of relying on our browser's default styling
- CSS is a language for specifying how documents are presented to users — how they are styled, laid out, etc.
    - **HTML** - Structure and Content
    - **CSS** - Style and Presentation
- CSS can be used for a variety of tasks:
    - Basic: simple text styling
    - Medium: page layout
    - Advanced: animations

CSS Syntax

- CSS is a rule-based language
  - We define style rules that should be applied to particular HTML elements on a web page
- For example, if we wanted to state, all level 1 headings (`h1`) should be bold text and blue in colour, we'd use the following rules:

```
h1 {
  color: blue;
  font-weight: bold;
}
```

- Note:
  - American English spelling
  - Semi-colons to terminate rules
  - Curly-braces to group rules

CSS Syntax (cont)

- Graphical overview of a CSS rule
- Another nice example, written in CSS itself!
  - https://codepen.io/marcobiedermann/pen/osurh

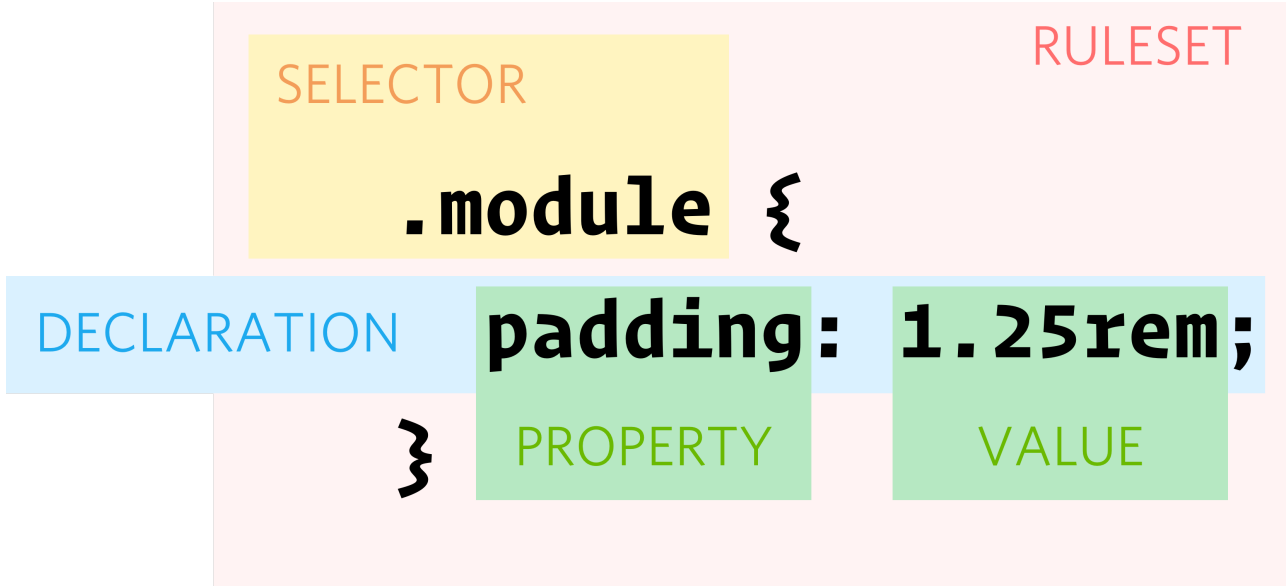

Image Source: https://css-tricks.com/css-ruleset-terminology/

A Quick Example

```html
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="utf-8">
        <title>Hello Stylish World</title>
        <link rel="stylesheet" href="css/DBI_012.Example_001.HelloWorld.css">
</head>
<body>
        <p> Hello Stylish World! </p>
</body>
</html>
```

```css
/* css/DBI_012.Example_001.HelloWorld.css */

p {
        color: orangered;
        background-color: lightgray;

        font-size: 200%; /* relative to the element's font size */
        font-weight: bold;
        text-align: center;

        border: 5px solid darkgrey;
        width: 50%;
        margin: auto;
}
```

Source available on Moodle - examples/css/DBI_012.Example_001.HelloWorld.css

Adding CSS to our document

External Stylesheet

- CSS rules in a separate file with a `.css` extension
- You can link a single CSS file to multiple web pages
- `<link rel="stylesheet" href="styles.css">`
- You are advised to use this method
    - Separates Style from Content (completely)
    - Easier to track changes

```
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="utf-8">
        <title>Hello Stylish World</title>
        <link rel="stylesheet" href="css/DBI_012.Example_001.HelloWorld.css">
</head>
<body>
        <p> Hello Stylish World! </p>
</body>
</html>
```

Adding CSS to our document

Internal Stylesheet & Inline Styles

- Internal Stylesheet
  - **Avoid this approach (generally)**
  - An internal stylesheet resides within an HTML document.
  - Place CSS inside a `<style>` element contained inside the HTML `<head>`
- Inline Styles
  - **Avoid this approach (always)**
  - CSS declarations contained within an element's `style` attribute

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Internal + Inline CSS</title>
    <style>
      p {
        color: red;
      }
    </style>
  </head>
  <body>
    <h1 style="color: blue;">Hello World!</h1>
    <p>This is my first CSS example</p>
  </body>
</html>
```

Cascade Hierarchy

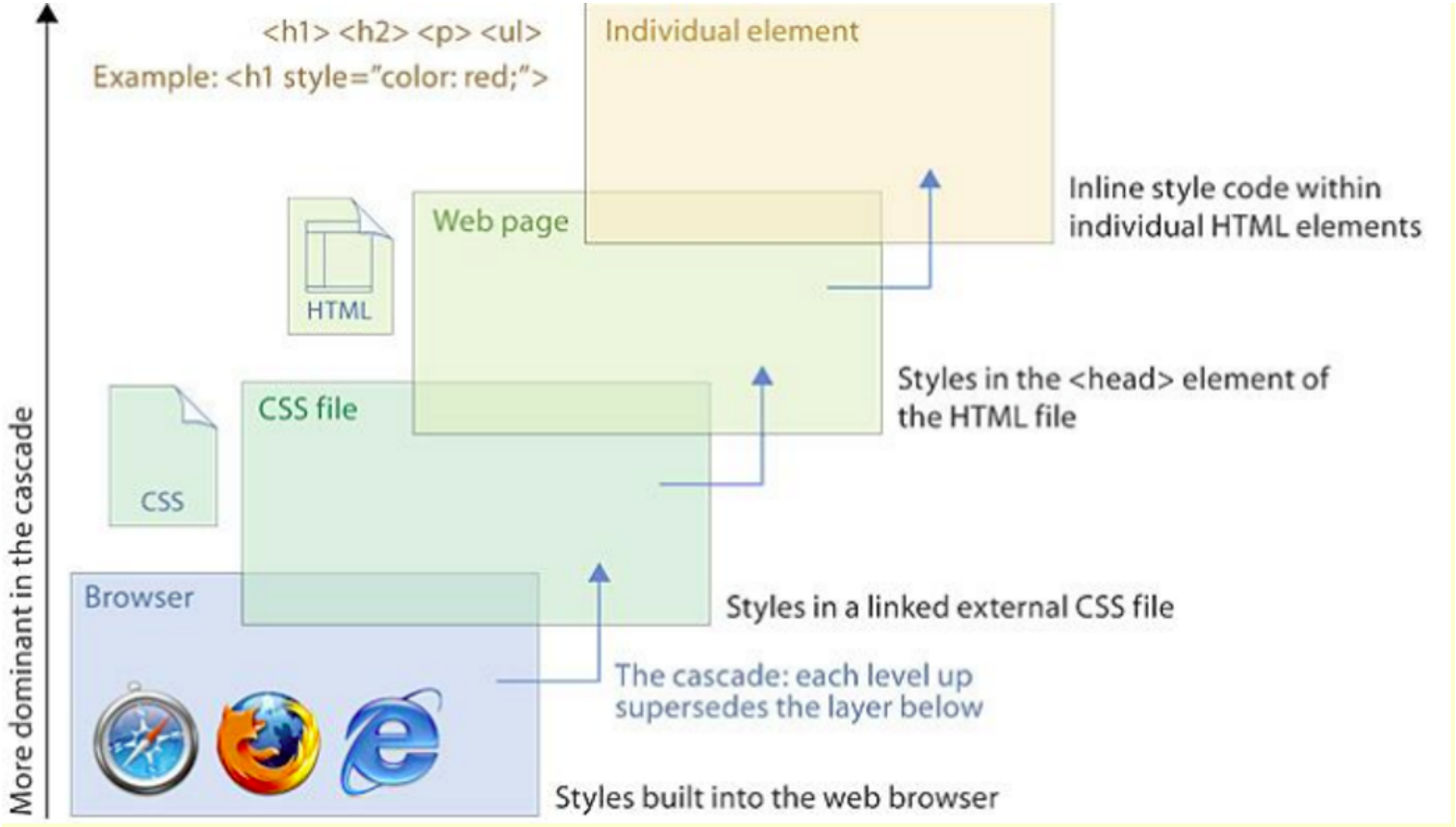

Image Source: From Web Style Guide: Basic Design Principles for Creating Web Sites, by Patrick J. Lynch and Sarah Horton.

Simple Selectors

- The selector is a tag name or a list of tag names, separated by commas
- Examples:

```
h1 {font_size: 24pt;}
h2, h3 {font_size: 20pt;}
```

- Contextual selectors
  - Apply style only to elements in specified position in body of document

```
body p b {font_size: 30pt}
```

  - I.e. Set the font size to 30pt for any bold element which appears within a paragraph element, which itself appears within the document's body (!).

Class Selectors

- Used to allow different occurrences of the same tag to have different style specifications
- Used when you want a style to apply to **more than one** kind/instance of tag
- A style class has a name, which is specified using the `.` specifier

```
.redBoldText{
  font-color: red;
  font-weight: bold;
}
```

- Any HTML element with that class will inherit these rules

```
<p class='redBoldText'>
  My bold red text.
</p>
```

ID Selectors

- An `id` selector allow the application of a style to one specific element
- Used when you want a style to apply to **exactly one** element
- A style `id` has a name, which is specified using the `#` specifier

```
#section3 {font-size: 20pt;}
```

- In HTML

```
<h2 id = “section3”>
        3. Properties for sale
</h2>
```

Pseudo Classes

- Pseudo classes are styles that apply when something happens, rather than because the target element simply exists
- Names begin with colons:
  - `hover` class applies when the mouse cursor is over the element
  - `focus` class applies when an element has focus

```
input:hover {color: red;}
input:focus {color: green;}
```

Cascade and Specificity Rules

- Later styles replace conflicting styles that appear earlier in the stylesheet. This is the **cascade** rule.
- A class is rated as being more specific, as in having more **specificity** than the element selector, so it cancels the other conflicting style declarations.
- **Note**: The `<div>` (division) tag is used as a container for HTML elements

```
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="utf-8">
        <title>Examples of Cascade and Specificity Rules</title>
        <style type="text/css">
                * {
                        font-size: 200%;
                        text-align: center;
                }
                div {
                        color: red;
                }

                div{
                        color: blue;
                }

                .green{
                        color: green;
                }
        </style>
</head>
<body>

        <div> What colour should I be? </div>
        <div class="green"> What colour should I be? </div>
</body>
</html>
```

Properties and Values

- **Properties**: human-readable identifiers that indicate which stylistic features you want to modify.
- **Values**: A property is assigned a value. This value indicates how to style the property.

> **Caution**: properties and values are case-sensitive.

```
selector {
  property: value;
}
```

- There are different properties in 12 categories (CSS 2.1) and 28 (CSS3):
  - Background, Border and outline, Dimension, Font
  - List, Margin, Padding, Positioning, Print, Table, Text

Property Values

- **Keywords** - left, small, ...
- **Length** - numbers, maybe with decimal points
  - **Units**: pixels (**px**), inches (**in**), centimeters (**cm**), millimeters (**mm**), points (**pt**)
- No space is allowed between the number and the unit specification, e.g. `1.5 in` is invalid!
- **Percentage** - just a number followed immediately by a percent sign. Defines the width in percent of the containing block
- **Colors**
  - Color name, e.g. `white`
  - Hex form: `#XXXXXX`, e.g. `#FFFFFF` - white
  - `rgb(n1, n2, n3)`, e.g. `rgb(255, 255, 255)`

The Box Model

- Everything (HTML Elements) in CSS has a box around it
  - **Content** - text, images, etc
  - **Padding** - area around content, usually transparent.
  - **Border** - goes around padding and content
  - **Margin** - Transparent. Clears the area around the border.

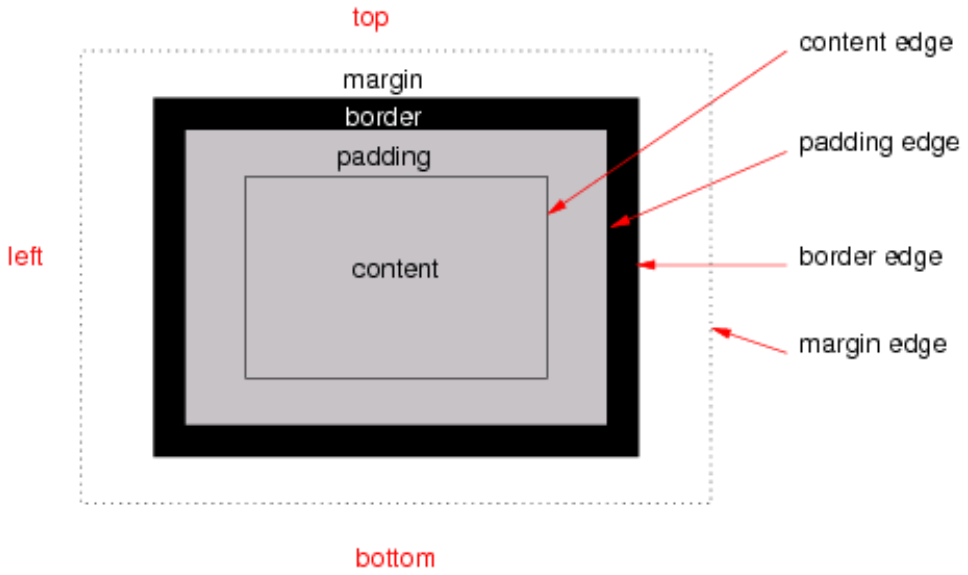

Image Source: https://www.w3.org/TR/css-box-3/

How Does CSS Work?

- Loads HTML and converts into a Document Object Model (DOM)
- Browser fetches other associated resources (CSS, JS, Images, etc)
- Parses the CSS and calculates which rules map to which DOM element

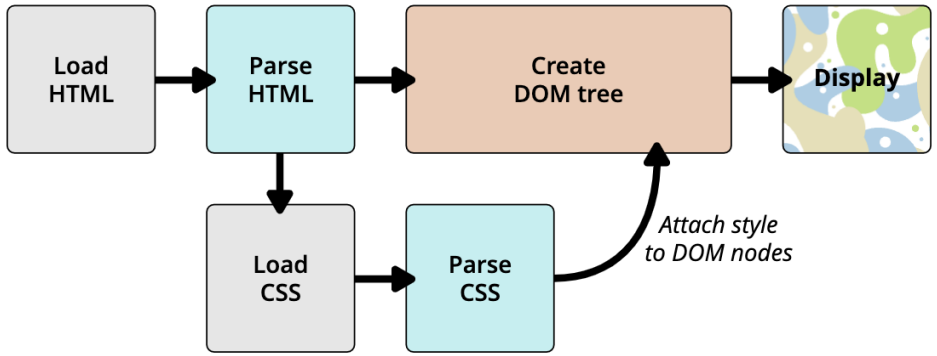

Image Source: https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/How_CSS_works

The Document Object Model (DOM)

- The DOM is a Tree-like structure. HTML elements, attributes and content are nodes
- DOM is an in-memory representation used by browsers when rendering web-pages
- Inspect the DOM using you web-browser's developer tools
- The DOM can be manipulated using JavaScript

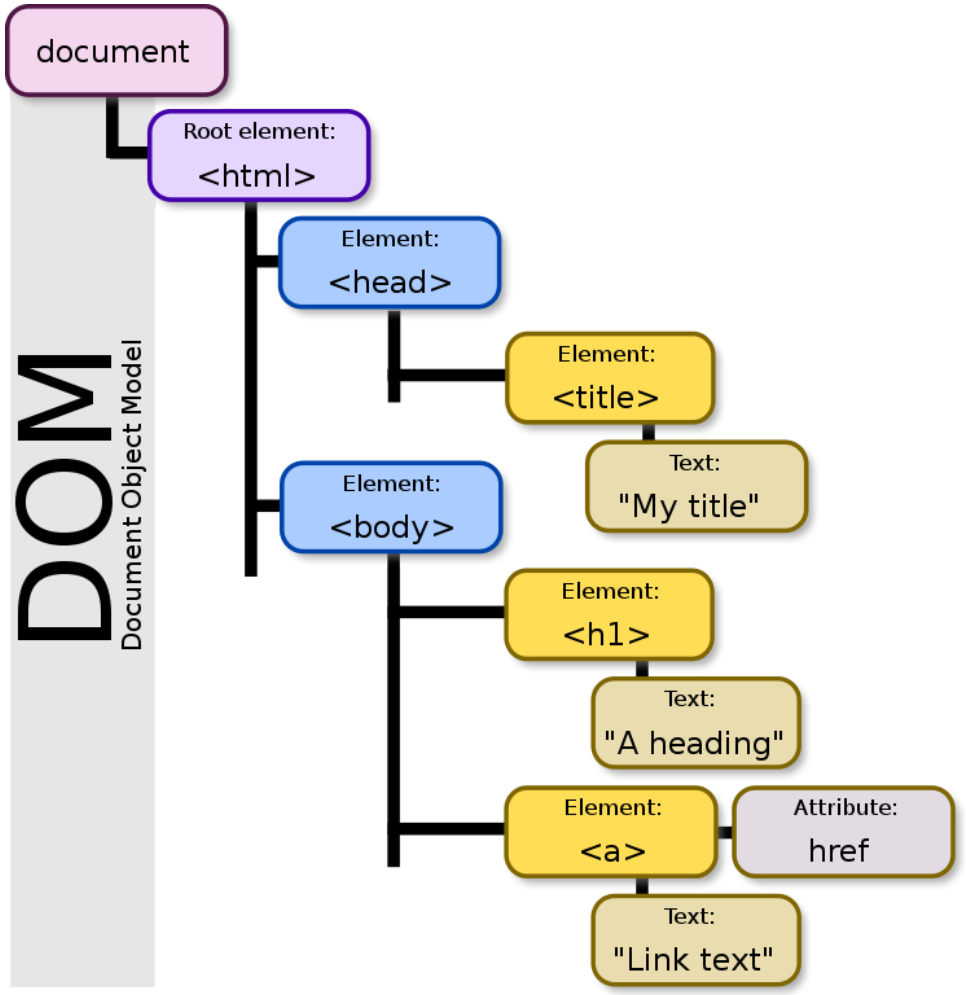

Image Source: https://en.wikipedia.org/wiki/Document_Object_Model

Tips and Tricks

- You will not be expected to remember all the possible styling rules
  - Use the MDN CSS Reference
- You will be expected to interpret and understand CSS rules
- Use your browser's development tools, they are especially useful for debugging CSS

Resources & Further Reading

- Mozilla Developer Network (MDN) - An Excellent Resource
  - MDN CSS Tutorial
  - MDN CSS Reference
- CSS Specification
- The examples presented here are available on Moodle
- Great Examples online
  - http://www.csszengarden.com/