

# CMSC388A

---

## Web Application Development with JavaScript



## CSS

Department of Computer Science  
University of MD, College Park

Slides material developed by Ilchul Yoon, Nelson Padua-Perez

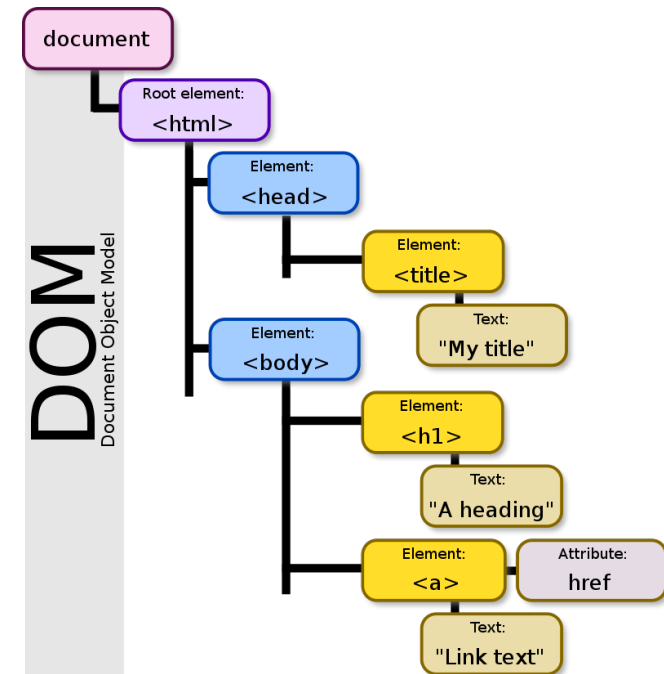
# VS Code Settings

---

- Select wheel (bottom left) and then “Settings”
- Auto Save with **afterDelay** option automatically saves every second
  - Recommended

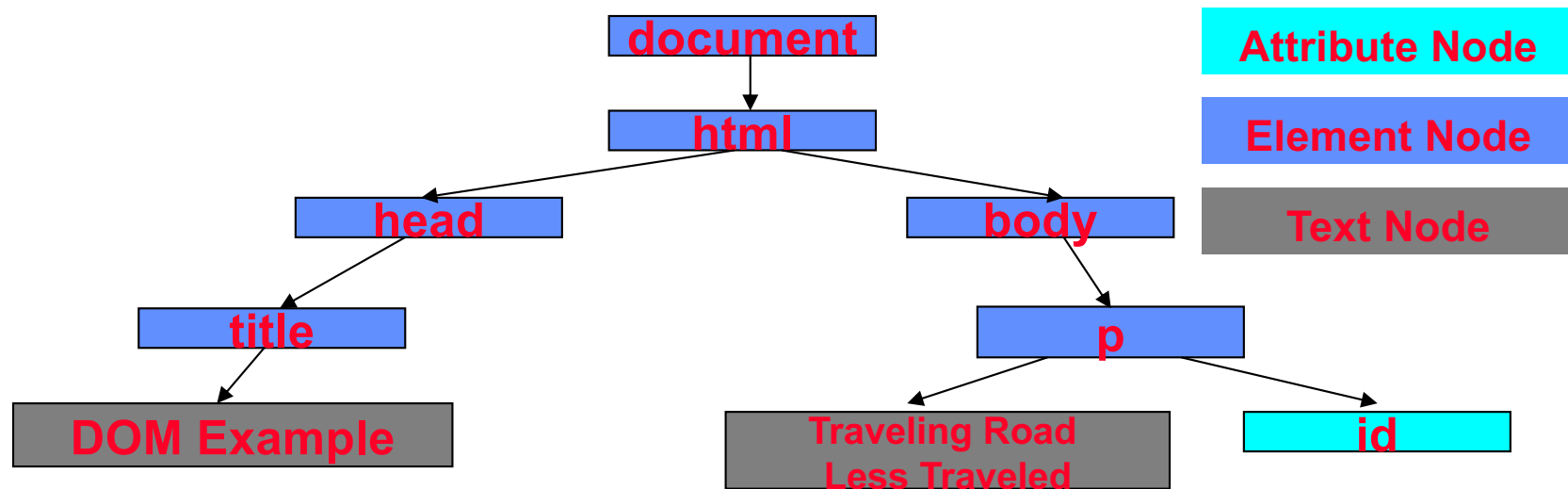
# DOM (Document Object Model)

- A cross-platform and language-independent **programming interface** for HTML and XML documents
- Represents a document with a logical tree wherein each node is an object representing a part of the document
- Each branch of the tree ends in a node, and each node contains objects
- Allow programmatic access to the tree
  - DOM allows changes to the structure, style, or content of a document
  - Nodes can have event handlers attached to them. Once an event is triggered, the event handlers get executed



Source: Wikipedia

# Example DOM for HTML File



```
<html>
  <head><title>DOM Example</title>
</head>
  <body>
    <p id="message">Traveling the road less traveled. </p>
  </body>
</html>
```

- head - child of html
- head and body - siblings

# CSS (Cascading Style Sheets)

---

- Official W3C standard **for controlling presentation**
- Style Sheets
  - Text file with rules. It includes no HTML
  - Style sheets files use a **.css** extension
  - Allows you to apply typographic styles (font size, line spacing, etc.)
  - Allows you to have page layout control
  - Smaller HTML files by avoiding redundancy in style specification
  - Easy update a collection of pages by updating only a single file
- **Example:** ExternalFile.css
- Why CSS?
  - <http://www.csszengarden.com/>
- Clearing CSS cache
  - <https://www.cs.umd.edu/~nelson/classes/resources/web/#clearcsscachechrome>

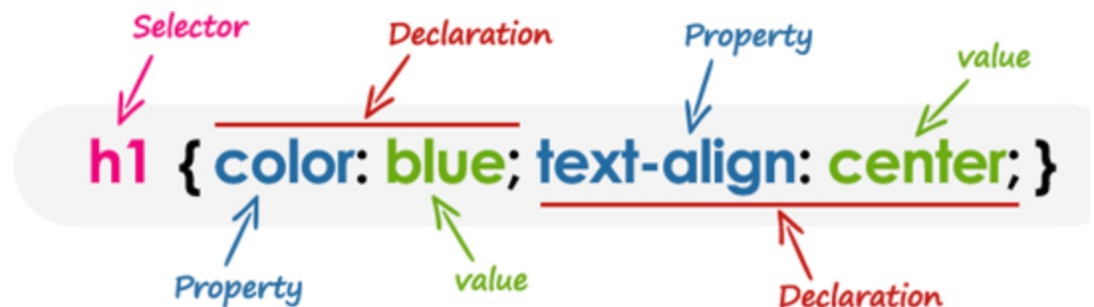
# CSS

---

- Rule
  - Basic element of a style sheet
  - **Describes the formatting associated with a page element**
- Rule format

## selector declarationBlock

- **Selector:** identifies what should be styled in a web document
  - e.g., h1, p
- **Declaration block:** a group of declarations describing styling information (**what** and **how** that portion of the web document should be modified)
- **Declaration:** property/value pair



# CSS

---

- **Declaration:** consists of **property**: **value** pair  
Example: **font-size**: **2em**;
- Popular properties
  - color, font-size, text-decoration, ...
- CSS Properties
  - <http://www.w3schools.com/cssref/>
- **Comments:** Surrounded by /\* \*/
  - You may use them to disable styles
- **Example:** ExternalFile.css

# Types of Style Sheets

---

- **Inline**
  - Style information applied to a specific tag  
(e.g., `<p style="font-size: 2em;">New Semester</p>`)
  - Avoid if possible
- **Internal**
  - Using the `<style>` tag in the header of the HTML document
  - Convenient to provide own style to a specific page
  - **Example:** InternalStyle.html
- **External**
  - External style sheet which web pages link to (see `<link>` tag)
  - Preferred approach
- **Example:** ExternalFile.html, ExternalFile.css



# CSS Validator

---

- <http://jigsaw.w3.org/css-validator/>
- Notice you have three choices
  - By URI
  - By file upload
  - By direct input
- Look at the “More Options” section
- We can also validate using the W3C VS Code extension
  - Let’s see a demo

# Colors

---

- You can specify colors using one of the following predefined colors (e.g., **yellow, white, red, purple, blue**)
- You can specify a color by indicating the red, green, and blue components. For example, all the following are equivalent:
  - red
  - `rgb(255,0,0)`
  - `#ff0000` or `#f00` (shorter, cannot use it for all colors)
- RGBA adds an alpha channel that allows us to specify the opacity for a color
  - RGBA color specification: **`rgba(RED, GREEN, BLUE, ALPHA)`**
  - Alpha value of 0 (transparent), alpha value of 1.0 (opaque)
    - » **Example:** `rgba(125, 125, 125, 0.5);`
- In VS Code, you can define a particular color (including opacity). Hover over a color of a CSS declaration in VS Code to see options
- References:
  - [http://www.w3schools.com/css/css3\\_colors.asp](http://www.w3schools.com/css/css3_colors.asp)

# Setting Size

---

- Length units - centimeters (cm), millimeters (mm), etc.
- Whitespace cannot appear between number and the unit (correct: 3em, incorrect: 3 em)
  - If the value is zero, the unit can be omitted
- Two types of length units:
  - **Relative** (e.g., em, rem)
  - **Absolute** (e.g., cm, pc)
- Relative
  - **em**: Relative to the size of the current font
    - » 3em: 3 times the size of the current font
  - **rem ("root em")**: inherits its value from the base font size of the **root element**
    - » In HTML, the root element is the <html> element
    - » **rem** is constant throughout a document
  - <http://www.sitepoint.com/power-em-units-css/>

# Setting Size

---

- Preferred: relative and percentages
- The **em** and **rem** units are practical in creating a scalable layout
- **Example:** SizeRemVsEm.html, SizeRemVsEm.css
- **Viewport:** browser window size
- [http://www.w3schools.com/cssref/css\\_units.asp](http://www.w3schools.com/cssref/css_units.asp)
- <https://www.w3.org/Style/Examples/007/units.en.html#units>

# Property inheritance

---

- Inheritance of Properties:
  - A child element inherits the same properties of its parent element (unless otherwise indicated by the CSS specification)
- **Examples of inherited properties**
  - color, font-family, font-size, letter-spacing, ...
- **Examples of non-inherited properties**
  - border, margin, padding, ...
- **Example:** PropInheritance.html, PropInheritance.css

# Kinds of Selectors

---

- **Type Selectors:**
  - Based on the name of an HTML tag (e.g., p, table, etc.)
  - **p** { color: red; }
- **Class Selectors:**
  - Allow us to define a style (“type of style”) we can apply to several elements
    - » Allow us to apply the same rules to a set of elements
  - Created with a period . (also known as a full stop)
- **ID Selectors:**
  - Similar to class selectors but appear only once in the document
  - Used when you need to apply a style only once in your document, or you want to capitalize on an id that already exists
  - Created using #
- **Example:** Selectors.html, Selectors.css

# Kinds of Selectors

---

- **Pseudo-class** - “phantom class” that allows assigning style to elements based on the state of certain elements (e.g., a link has been clicked) or markup patterns (e.g., odd rows of a table)
  - Attached to selectors (e.g., p:hover)
  - Four popular pseudo-classes uses are
    - » **a:link**: the initial color of a link
    - » **a:visited**: color for a visited link
    - » **a:hover**: color when the mouse hovers over the link
    - » **a:active**: color during the clicking of the link
- **Example:** Selectors.html, Selectors.css