JavaScript

Dr. Garrett Dancik

JavaScript basics

- JavaScript is a programming language that can be used to add, remove, change, or modify HTML elements and CSS settings on a web page.
- Although JavaScript can be used as a standalone language, most often is it used for creating *dynamic* web pages
- JavaScript syntax has many similarities with C++ and Java, though it is not related to either
- These notes and examples highlight some of the key concepts and differences between JavaScript and other programming languages.

JavaScript examples

```
// example for loop
var sum = 0;
for (let i = 1; i <= 10; i++) {
		sum += i;
}
document.write('<p>The sum of 1-10 is: ' + sum + '</p>');

// example if..else statement
var sum = 0;
if (sum > 5) {
		document.write('<p>The sum is greater than 5</p>');
} else {
		document.write('<p>The sum is NOT greater than 5</p>');
}
```

Key difference between JavaScript and C++/Java: Variable declaration and scope

- Variables declared (e.g., with *var*) outside a function have *global* scope; variables declared inside a function have *local* scope. Variables do not have *block* scope unless declared with *let*
- Variables can be used (initialized) without being declared, and will have *global* scope
- A variable that is not initialized will have the value *undefined*
- Variables declared with *let* have <u>block</u> scope
- Variables in JavaScript can be re-declared and the type can be changed.

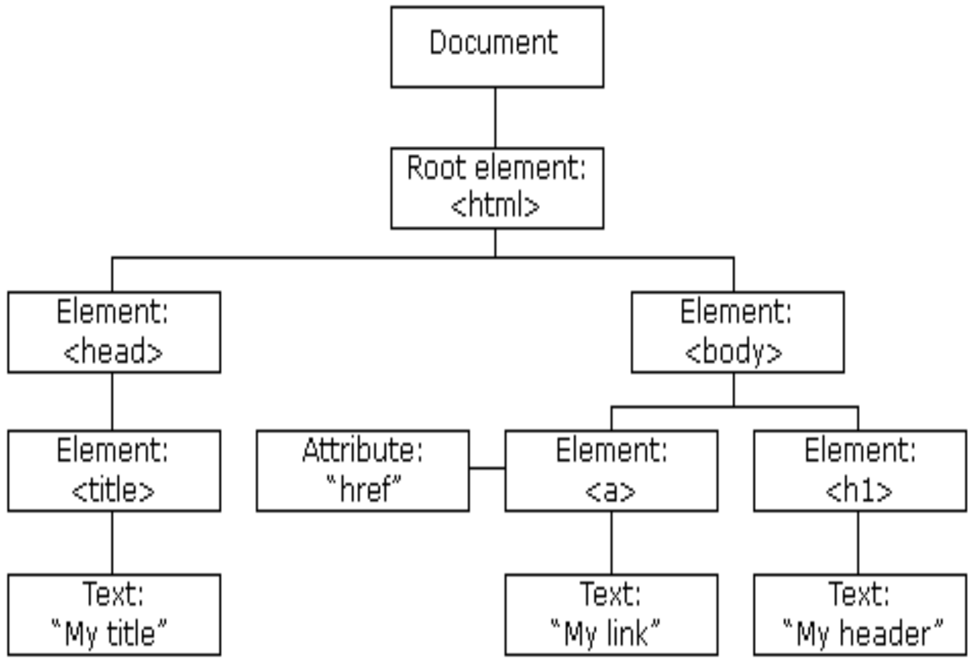
Key difference between JavaScript and C++/Java: Variable declaration and scope

- JavaScript has a bizarre behavior known as “hoisting” where variable declarations (but not assignments) are moved to the top of the current scope.
- This is why, for example, in Javascript you can call functions before defining them. However, for standard variables, it is good practice to declare variables before they are used using a *var* or *let* statement
- More details and examples:
  - https://www.w3schools.com/js/js_scope.asp
  - https://www.w3schools.com/js/js_let.asp

HTML Document Object Model (DOM)

- The HTML DOM provides standards for programmatically accessing, changing, adding, or deleting HTML elements

- Key observation:
  - The DOM defines a tree where HTML elements have *children* and *parents*
  - Each HTML element has attributes and styles and includes its children



source: https://www.w3schools.com/js/js_htmldom.asp

Finding and changing HTML elements

Method For Finding HTML elements	**Description**
document.getElementById(id)	Find an element by its unique id (returns a **single** element)
document.getElementsByTagName(name)	Find elements by tag name (returns an **array** of elements)
document.getElementsByClassName(name)	Find elements by class name (returns an **array** of elements)

Syntax for accessing and/or changing* an element	**Description**
element.innerHTML	The inner HTML of an element (may contain HTML tags)
element.innerText	The inner text of an element (HTML tags are ignored)
element.attribute	The attribute value of an HTML element
element.style.*property*	The style of an HTML element (properties are in camelCase, e.g., 'background-color' is 'backgroundColor')

*Assignment is used to change the corresponding value; for example to change the HTML of an element use, e.g., element.innerHTML = “<h2> Changed </h2>”

Modified from: https://www.w3schools.com/js/js_htmldom_document.asp