Javascript

Dr. Garrett Dancik

Javascript basics

- Javascript is a programming language that can be used to add, remove, change, or modify HTML and CSS elements on a web page.
- Although Javascript can be used as a standalone language, most often is it used for making web pages *dynamic*
- 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 with other programming languages.

Javascript examples

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

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

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.
- Variables declared with *let* have block scope
- Variables in Javascript can be redeclared and the type can be changed.

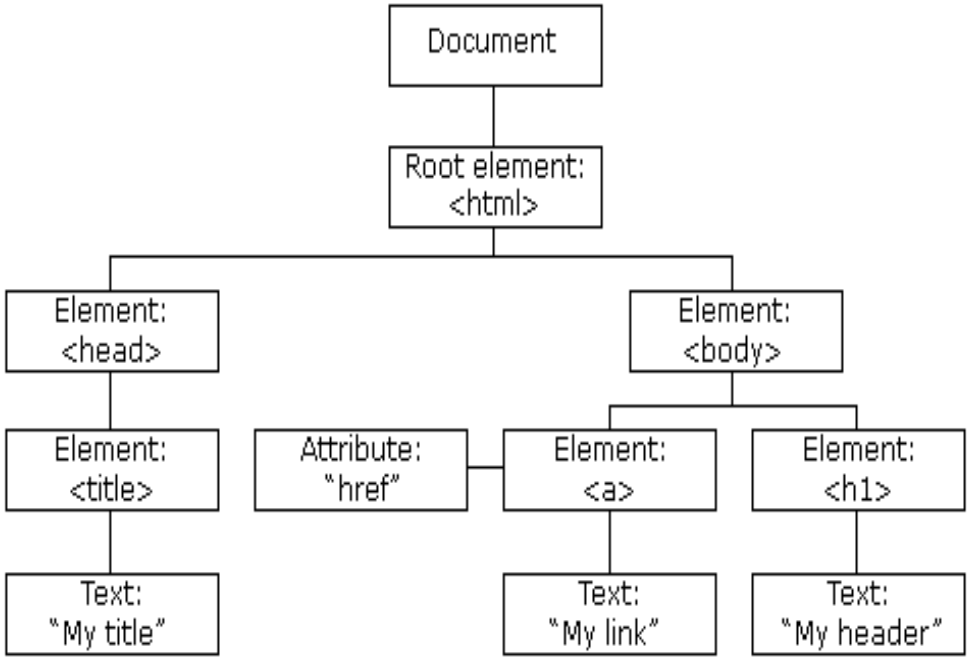
Key difference between Javascript and C++/Java: Variable declaration and scope

- 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.
- 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 element 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* HTML elements	Description
element.innerHTML	The inner HTML of an element
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

*Assignment can be used to change an HTML element, e.g.,
element.innerHTML = “<h2> Changed </h2>”

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