Lecture #17: Introduction to JavaScript

CS106E Spring 2018, Young

In this lecture, we take a look at client-side processing. We consider what our client-side programming options are. We take a look at a simple example of how JavaScript can be used on the web, highlighting some of the key techniques used. We then discuss what broader capabilities are provided by the web browser for client-side processing.

The bulk of the material covered in this lecture is discussed in the “h12 Introduction to JavaScript” and “h13 The JavaScript Language” handouts.

Client-Side Processing Languages
- As I’ve previously mentioned, the vast majority of Client-Side Processing is done with JavaScript
- Java can also run in a web browser, although most security experts recommend that you disable your web browser’s ability to run Java, as this is a known vector for attacks on your computer.
- Adobe Flash is also used on the Client-Side, although it’s no longer in favor, and Adobe is scheduled to end support for it in 2020.
- In addition, as we briefly discussed in the CPU lectures when talking about compilers and interpreters, there are some languages that will compile to JavaScript.
  o Users of these language will use compilers to convert their code to JavaScript, and then their web server will serve out the converted JavaScript code to visitors.
  o These languages include TypeScript, Google Dart, and Scala

About JavaScript
- While JavaScript is primarily used in the web browser, it can be used in other places as well.
  o The Node.js webserver is quite popular and uses JavaScript
  o The MongoDB database uses JavaScript’s object model and supports JavaScript code.
  o JavaScript can also be embedded in other applications as a scripting language.
- The official name for JavaScript is ECMAScript
  o Netscape Corporation passed control of the language to the European Computer Manufacturers’ Association for standardization.
  o You usually will only hear the term ECMAScript in reference to specific versions of the language and the features they added. For example:
    ▪ ECMAScript 6 added a considerable number of features including support for traditional classes.

Frameworks and Libraries Built on JavaScript
- There are a wide number of frameworks and libraries built on top of JavaScript. Some of these are very common. We may look at these in more detail next lecture. Some of the most common include:
  o jQuery, Twitter Bootstrap, Angular, React, and Vue.
- Writing JavaScript without an accompanying framework is sometimes referred to as vanilla JavaScript.
JavaScript Language Characteristics
- Untyped variables, parameters, and return types.
- Limited range of primitive data types
  - Only one Number type which is used for both floats and integers
- Object system based on Prototypes, not traditional classes (see h13 for details)
- Interpreted

Events for Scripting
- There are quite a number of different events we can write scripts for. These include:
  - mouseover and mouseout execute scripts as the mouse moves over an element or moves out of an element.
  - keydown, keyup, keypress execute scripts in response to the user typing at the keyboard.
  - mousemove allows us to execute code as the user moves the mouse around the screen.

HTML Element Properties
- We can change both the HTML properties and the CSS properties of elements on the webpage.
- Properties of particular note include:
  - Changing an <img /> tag’s src property allows us to change an image displayed on our webpage from one png or jpeg to another.
  - Modifying the CSS position properties allow us to move items around on the webpage. This is how something like Google Maps allows us to drag the map around.
  - Modifying the display property to none or setting the visibility property to hidden makes an element on the webpage disappear.

Other Browser Objects
- In addition to the document object, which gives us access to the various HTML elements on a webpage, there are several other browser objects we can access. These include:
  - location allows us to change the current webpage being displayed programmatically from within our JavaScript code.
  - history allows us to move forward and backward in the browser history, going back to a previous webpage the user has visited, or once we’ve gone backwards, returning forwards to the latest webpage.