Introduction to JavaScript

Dylan Lane McDonald

CNM STEMulus Center
Web Development with PHP

November 11, 2015

Outline

History of JavaScript

JavaScript began its life at Netscape in 1995. Originally slated for Netscape 2.0, the language was originally called *LiveScript* and subsequently renamed to *JavaScript* shortly before Netscape 2.0's release. This begs the question: is JavaScript similar to Java?

History of JavaScript

JavaScript began its life at Netscape in 1995. Originally slated for Netscape 2.0, the language was originally called *LiveScript* and subsequently renamed to *JavaScript* shortly before Netscape 2.0's release. This begs the question: is JavaScript similar to Java?

ABSOLUTELY NOT!!!!

History of JavaScript

JavaScript began its life at Netscape in 1995. Originally slated for Netscape 2.0, the language was originally called *LiveScript* and subsequently renamed to *JavaScript* shortly before Netscape 2.0's release. This begs the question: is JavaScript similar to Java?

ABSOLUTELY NOT!!!!

JavaScript's similarities to Java are purely superficial. The syntax is derived from a common ancestor, C, and both use C-like syntax. The similarities stop there.

Features of JavaScript

JavaScript is an **interpreted** language run by a JavaScript engine, normally a component of the end user's web browser. Some of the key features of JavaScript are:

- **Imperative**: Functions can be written to perform specific tasks
- **Object Oriented**: Objects can be modeled around what the system is
- **Dynamically Typed**: The variable type (e.g., String, Integer, ...) are decided at run time

JavaScript is enabled on most end user's web browsers and is used in creating responsive, dynamic, and fun sites.

Javascript Events

An event can roughly be categorized into one or both of the following categories:

- Something the browser does
- Something the user does

Javascript Events

An event can roughly be categorized into one or both of the following categories:

- Something the browser does
- Something the user does

Events are the centerpiece of creating interactive web sites. Whether flat JavaScript or a framework such as Angular or jQuery are used, reacting to events is the key to creating dynamic and interactive web sites.

Javascript Events

An event can roughly be categorized into one or both of the following categories:

- Something the browser does
- Something the user does

Events are the centerpiece of creating interactive web sites. Whether flat JavaScript or a framework such as Angular or jQuery are used, reacting to events is the key to creating dynamic and interactive web sites.

Events map everything from user interaction such as key presses, mouse movements, and scrolls to network events such as loads and going offline.

Example Events

Events are a reaction to a user's action on a web site. Table 1 lists a very small subset of possible JavaScript events.

Event	Comment
click	when a user clicks on an element
change	when a user changes an input field
drag	when a user drags an element
drop	when a user drops an element onto another
scroll	when a user scrolls (desktop) or swipes (mobile)

Table 1: Common JavaScript Events

As always, a more exhaustive list of all the possible events are available at the Mozilla Developer Network. [1]

Closures

Definition

A **closure** *is a JavaScript function that generates another function. The advantage of a closure is to generalize functionality for use cases that only vary slightly.*

Closures

Definition

A **closure** *is a JavaScript function that generates another function. The advantage of a closure is to generalize functionality for use cases that only vary slightly.*

An example use case of a closure is to greet the user in many different languages. Here, we have a constant and a variable:

- **Constant:** the user's name
- **Variable:** the exact words in the user's native language

Closures

Definition

A **closure** *is a JavaScript function that generates another function. The advantage of a closure is to generalize functionality for use cases that only vary slightly.*

An example use case of a closure is to greet the user in many different languages. Here, we have a constant and a variable:

- **Constant:** the user's name
- **Variable:** the exact words in the user's native language

Using a closure, one can boilerplate the words in the particular language and concentrate on the only variable: the user's name. The use of closures is vast in JavaScript frameworks such as Angular and jQuery, where closures are applied to multiple elements at once using class selectors.

Works Cited


Mozilla Development Network.
Event reference.
https://developer.mozilla.org/en-US/docs/Web/Events, 2015.