

JAVASCRIPT FOR JAVA DEVELOPERS: AN INTRODUCTION

WRITTEN AND CREATED BY EDUARDO DE LUNA

October, 2014.

OBJECTIVES, SUBJECTS AND COURSE METHODOLOGY

OBJECTIVES

- To familiarize students with the history, uses, and syntax of the Javascript Programming Language
- To convey to students, familiar with the use of Java and/or Python, the differences between it and Javascript, with particular emphasis on prototypical inheritance
- To introduce students to advanced techniques and practical applications of the Javascript Programming Language
- To pique attendant's interest on the potential of Javascript as a capable programming language for their future endeavors

SUBJECTS

A Brief History of JS

The work of Brendan Eich and the beginnings of JS
The browser war with Microsoft
The participation of ECMA International
Douglas Crockford and the creation of JSON
AJAX: The game changer of JS
JS as a result of its history
Further Information: Crockford on Javascript - History of Javascript

How to Use and Code JS

Using the Console as Integrated Development Environment
Using the <script> tag
Ability of concatenating multiple script files
JavaScript Editey for Google Chrome
Scratchpad for Mozilla Aurora
JSFiddle.net, Codepen.io, and Runnable.net
IDEs: Sublime Text and JetBrains WebStorm
A view under the hood: Javascript Engines

Javascript Essentials

Javascript Syntax
Similarities with Java and Python
Dynamic Typing and Variable Hoisting
Departure from Java's Static Typing
Everything is an Object

- Departure from Java's Primitives and Objects
 - Variable Scope
 - Numbers
 - The concept of NaN
 - What stands for false and true
 - Strings and Arrays
 - Control Structures
 - Functions
 - Objects
 - The "this" object
 - Javascript Object Notation
 - The Window/Global Object and Namespaces
 - Departure from Java's Packages
- Further Information: Crockford on Javascript – The JavaScript Programming Language

Prototypical Inheritance: The heart of JS

- The Classical Inheritance Model (recap)
- What is prototypical inheritance?
- Constructor Functions
 - Prototype Chaining
- Object Augmentation
- Parasitic Inheritance
- Alternatives: Typescript and ECMAScript Harmony
- Further Information: Source Decoded - Prototypes, Prototypical Inheritance done right.

Examples of Advanced Uses of JS

- Functional Programming
- Promises
- Modules
 - CommonJS and AMD
- WebGL
- Further Information: nodeschool.io

External Libraries in JS

- jQuery
- Phaser Game Engine
- Three.js
- Angular.js and Backbone.js
- Apache Cordova
- Node.js
 - Socket.io
 - Nodeuino
 - Node-Webkit

Conclusions and Parting Thoughts

CLASS METHODOLOGY

- The conference will be held as a webinar using Google Hangout and Hangouts On Air to broadcast and archive the content on YouTube for later viewing.
- Every subject point during the webinar will be covered with a detail explanation designed to be direct and effective.
- The audience is expected to have previous knowledge in computer programming, as such the seminar will gloss over simple concepts like data-flow structures, the nature of variables and logical operators.
- Every technical point is to be presented with a live coding example that demonstrates the use of the Javascript feature in question. These examples will be hosted on a GitHub repository for download.
- The audience watching live and in the hangout will have the ability to make questions at specific times during the presentation, to clarify any kind of misunderstanding. These questions are to be asked through text only and only made when prompted to by the teacher. This is to facilitate seminar flow and making efficient use of time.
- All subjects and material will have additional material readily available on the web or for free that furthers the discussion on each major topic in order to give the information validity and give students a chance to understand JS at a deeper level.