Introduction to Javascript

Definition

JavaScript is most commonly used as a client side scripting language. This means that JavaScript code is written into an HTML page. When a user requests an HTML page with JavaScript in it, the script is sent to the browser and it's up to the browser to do something with it. The fact that the script is in the HTML page means that your scripts can be seen and copied by whoever views your page. This openness is a great advantage, because the flip side is that you can view, study and use any JavaScript you encounter on the WWW.

**<!doctype html>**

**<title>JavaScript Test</title>**

 **<script>**

**alert('Hello Babylon!');**

**</script>**

JavaScript vs. Java

 Although the names are much alike, JavaScript is primarily a scripting language for use within HTML pages, while Java is a real programming language that does quite different things from JavaScript. In addition Java is much harder to learn. It was developed by **Sun** for use in pretty much anything that needs some computing power.

**document.write("<h1>This is a heading</h1>");**

**document.write("<p>This is a paragraph</p>");**

Java and JavaScript both descend from C and C++, but the languages (or rather, their ancestors) have gone in quite different directions. You can see them as distantly related cousins. Both are object oriented (though this is less important in JavaScript than in many other languages) and they share some syntax, but the differences are more important than the similarities.

**<button type="button" onclick="alert('Welcome!')">Click Me ! </button>**

Javascript Core Components

JavaScript is an object oriented dynamic language; it has types and operators, core objects, and methods. Its syntax comes from the Java and C languages, so many structures from those languages apply to JavaScript as well. One of the key differences is that JavaScript does not have classes; instead, the class functionality is accomplished by object prototypes. The other main difference is that functions are objects, giving functions the capacity to hold executable code and be passed around like any other object.

Let's start off by looking at the building block of any language: the types. JavaScript programs manipulate values, and those values all belong to a type. JavaScript's types are:

* Numbers
* Strings
* Booleans
* Functions
* Objects

Undefined and Null, which are slightly odd. And Arrays, which are a special kind of object. And Dates and Regular Expressions, which are objects that you get for free. And to be technically accurate, functions are just a special type of object. So the type diagram looks more like this:

* Number
* String
* Boolean
* Object
* Function
* Array
* Date
* RegExp
* Null
* Undefined