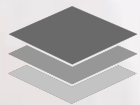




UI Engineering Studio. Day 5



Bootcamp: JS

Javascript

JavaScript (JS) is a lightweight, interpreted or JIT compiled programming language with **first-class functions**. Most well-known as the scripting language for Web pages

-MDN



History

Env

Basics

UI Boot Camp: JS

History

- Brendan Eich
- 1995
- Mocha
- Netscape

Nothing related to java, it was popular and they've used it.



UI Boot Camp: JS

History

ECMA 1,2,3 - > 1999 (Make it consistent with the existing implementation)

ECMA 4 - > 2000

ECMA 5 -> 2009 First post harmony release. (json, object, strict)

ECMA 6 -> 2015

Year based release.

The logo for ECMAScript, featuring the text "ECMAScript" in a bold, black, sans-serif font, centered within a solid yellow rectangular background.

ECMAScript

UI Boot Camp: JS

Env

Ecma

ActionScript

Jscript

Javascript

Browser.

Node

ApacheCouch DB



UI Boot Camp: Basics JS

Coercion

Type coercion is the process of converting value from one type to another (such as string to number, object to boolean, and so on). Any type, be it primitive or an object, is a valid subject for type coercion.

-MDN

```
true + false
```

```
12 / "6"
```

```
"number" + 15 + 3
```

```
15 + 3 + "number"
```

```
[1] > null
```

```
"foo" + + "bar"
```

UI Boot Camp: Basics JS

Scope and context

Every function invocation has both a scope and a context associated with it. Fundamentally, scope is function-based while context is object-based. In other words, scope pertains to the variable access of a function when it is invoked and is unique to each invocation. Context is always the value of the **this** keyword which is a reference to the object that “owns” the currently executing code.

- **Ryan Morr**

```
function exampleFunction() {  
    var x = "declared inside scope";  
    this.x= "context of the function";  
    console.log(x);  
  
    console.log(this.x);  
}  
  
console.log(x);
```


UI Boot Camp: Basics JS

Hoisting

Hoisting is JavaScript's default behavior of “*moving*” all declarations to the top of the current scope.

```
num = 6;  
console.log(num); // returns ?  
var num;
```

UI Boot Camp: Basics JS

Equal

=== !== == !=

UI Boot Camp: HTML

Homework: Calculator



