

Introduction to JavaScript CS50

Monday, June 13, 2022 7:39 PM

⇒ Browsers only understand Javascript.

⇒ Javascript is Interpreted

i) Each browser has its own Javascript engine, which either interpret the code or compile it in some way

ii) They all follow ECMAScript standard, but may differ for anything which is not standardise by ECMA.

⇒ Type

i) Dynamic Typing

ii) Primitive types (No methods)

iii) Objects

⇒ Type casting? Coercion ✓

var x = 42

var explicit = String(x)

var implicit = x + ""

// explicit = "42"

// implicit = "42"

⇒ == vs ===

→ == covers the types

→ === requires equivalent types

JavaScript is Interpreted

- Each browser has its own JavaScript engine, which either interprets the code, or uses some sort of lazy compilation
 - V8: Chrome and Node.js
 - SpiderMonkey: Firefox
 - JavaScriptCore: Safari
 - Chakra: Microsoft Edge/IE

Coercion, cont.

- Which values are falsy?
 - undefined
 - null
 - False
 - +0, -0, NaN
 - ''
- Which values are truthy?
 - {}
 - []
 - Everything else

A detailed diagram titled "JavaScript Equality Table" showing the results of various comparisons between different types of values in JavaScript. The table highlights that NaN is not equal to itself and that objects are compared by reference.

<https://github.com/dorey/JavaScript-Equality-Table>

HTML script src Attribute Document

127.0.0.1:5500/first.html

Apple Sarkar DBT (59) Complete Java... Placements - Geeks... Studying heating ef... Chayanprash: A Tr...

Hello World

4 messages

1 user mess...

No errors

3 warnings

1 info

No verbose

Filter

Default levels

No Issues

brain.js:9

```
1 var o = new Object();
2 o.firstname = "Dheeraj";
3 o.lastname = "Jadhav";
4 o.isStudent = true;
5 o.age = 20;
6
7 o.greet = function () { console.log("hello I
8 am ", o.firstname); };
9 console.log(o);
```

➤ Different ways to create an object

HTML script src Attribute x Document x + − ×

127.0.0.1:5500/first.html

Apple Sarkar DBT (59) Complete Java... Placements - Geeks... Studying heating ef... Chyawanprash: A Tr...

Hello World

Elements Console Sources Network Performance Memory > | ⚙️ ⚙️

top Filter Default levels No Issues 3 hidden ⚙️

6 messages 3 user mess... No errors 3 warnings 3 info No verbose

```
brain.js:8
{
  firstname: 'Dheeraj',
  lastname: 'Jadhav',
  isStudent: true,
  age: 20,
  greet: function () {
    console.log('Hello ' + this.firstname + ' ' + this.lastname);
  }
}

brain.js:15
{
  firstname: 'Suraj',
  lastname: 'Jadhav',
  isStudent: true,
  age: 20,
  greet: function () {
    console.log('Hello ' + this.firstname + ' ' + this.lastname);
  }
}

brain.js:21
{
  firstname: 'Rushi',
  lastname: 'Jadhav',
  isStudent: true,
  age: 20,
  greet: function () {
    console.log('Hello ' + this.firstname + ' ' + this.lastname);
  }
}
```

File Edit Selection View ... brain.js - Web_Development_CS... Run Testcases 0 0 Ln 8, Col 16 Spaces: 4 UTF-8 CRLF {} JavaScript Port: 5500 8:44 PM 6/13/2022

Prototyping

The screenshot shows a browser window with the URL `127.0.0.1:5500/first.html`. The page content is "Hello World". Below the page, the browser's developer tools are open, specifically the Console tab. The console output is as follows:

```
Console was cleared
VM1263:1
< undefined
> var x = 44;
< undefined
> x
< 44 > → Object
> x.toString()
< '44' >
> 44.toString() → Primitive Datatype
VM1373:1
✖ Uncaught SyntaxError: Invalid or unexpected token
VM1373:1
>
```

A red arrow points from the handwritten note "Primitive Datatype" to the output of `44.toString()`.

⇒ Objects have methods associated with them.

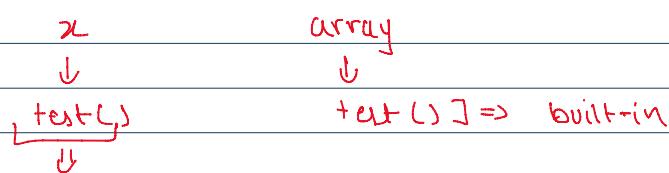
⇒ primitives have no methods associated with them.

\Rightarrow now x is an instance of an object.

\rightarrow If x has its own method
 $\in u$

```
var n = []
n.text = function () { }
```

→ As `x` is an array object,
let's assume that array has
a its own test method



User-defined

⇒ Now if `test()` of array is overridden, then user-defined `test()` will override built-in `test()`.

i.e. if we call `x.test()`, user-defined `test()` will be called.

How Javascript Engine works?

The JavaScript Engine

- Before executing the code, the engine reads the entire file and will throw a syntax error if one is found
 - Any function definitions will be saved in memory
 - Variable initializations will not be run, but variable names will be declared

⇒ Before execution of code Javascript Interpreter, do following things.

- i) If a function is defined it will be saved in the memory.
- ii) If a variable is there, then its name will be declared, value will not be stored.

b will not be stored.

```
Console.log();
Greet();

Function greet() {
    Console.log('hi');
}

Var i=42;
```

Let's say we have this code

⇒ so, before execution

- i) greet will be stored in memory
- ii) i will be declared, not initialized.



Now at the time of execution

⇒ though greet() is called before its defined, it will be executed error free.

⇒ As it was already stored in memory before execution



Now, in case of variable i

⇒ there will be an error



Because, i was declared in memory but its value was not stored, its value was defined later in the code



⇒ So, when i was called in the first line, i was undefined
that's why it will give error

WHAT IF?

⇒ we don't declare i later in the code

⇒ Error will be known, as it was not even declared in the code.

A screenshot of a web-based development environment. On the left, a code editor window titled "script.js" shows the following code:

```
1 console.log(i);
2
3 greet();
4
5
6▼function greet() {
7   console.log('Hello, how are you');
8 }
9
10 var i = 45;
```

The right side features a terminal window titled "Output" with the URL "https://Web-Development.dheerajjadav1.repl.co". The output shows:

```
Hello world
```

Below the terminal is a "Console" tab with the following logs:

```
All Error Warning Info
undefined
Hello, how are you
```

A screenshot of a web-based development environment, similar to the one above, showing the same code in "script.js". The "Output" window shows:

```
Hello world
```

The "Console" tab shows an error message:

```
All Error Warning Info
i is not defined
at https://f21db115-c153-46d9-a925-8b436d3256a9.id.repl.co/script.js:1:13
```



This is called Hoisting in Javascript.



Hoisting :- It allows us to use functions and variables before they are declared.

<https://www.freecodecamp.org/news/what-is-hoisting-in-javascript/>

=> Hoisting in
Détail



The Global Object

- All variables and functions are actually parameters and methods on the global object
 - Browser global object is the 'window' object
 - Node.js global object is the 'global' object



Execution context

- Equivalent to a "stack frame" in C
- Wrapper of variables and functions local to a function's execution
- Collection of execution contexts is known as the execution stack