Learn to Program in JavaScript: Beginner to Pro

Introduction to JS

Intro

- JS is lightweight programming language developed by Brendan Eich.

- JS was first coined as LiveScript but when Netscape partnered with Sun microsystems, they changed it to JavaScript.

- JS makes the website responsive and usable since it is a programming language

- ECMA (European Computer Manufacturers Association) standardize JS.

- JS can be linked to HTML or used with in a HTML file. <script> tag is used in HTML to put JS snippets.

- REPL is a style that browser consoles use, and its full form is Read Evaluate Print Loop. Basically, it loops - reads, evaluate and prints, so we can simultaneously give inputs and checks in the browser.

Data Types

- var is a variable declaration keyword. Unlike other programming languages JS variables doesn’t need to declare its type so it is called DYNAMICALLY TYPED whereas other programming languages like C, C++ and JAVA are statically typed. Ex: var messages = “Hello”; . The browser interpreter/compiler dynamically sets the type of variable when we assign value if not it simply takes undefined.

- JS has categorized its data types into 3 as follows,

— Primitive (Numbers, String, Boolean)

— Object (Apart from primitive types all other types are considered objects even functions are considered as objects)

— Null (is a special type of object in JS that represents “no-objects”)

A screenshot of a cell phone

Description automatically generated

- Undefined is a type of undefined and represent absence of value.

- The below is a REPL concept the chrome console first reads from me and evaluates it and prints it as undefined and gain the console terminates.

A screenshot of a cell phone

Description automatically generated

Flavor of JS

- JS is a scripting language (set of statements which will be interpreted)

- It also supports functional (JS functions are first class citizens that is they are given the first priority, functions is treated as data or var may holds functions) and OOP (not class based but prototype) languages

- It is dynamically typed. Whereas C, Java are statically typed.

A screenshot of a cell phone

Description automatically generated

Ex:

A screenshot of a cell phone

Description automatically generated

String datatype

- Each character of string uses 16 bits of storage. It uses UTF-16 by default.

- We can use either “ ” or ‘ ’ for strings.

- var msg “Hello world”, where Hello world is literally a string, so it is called string literals.

- JS has no inbuilt character data set they only have strings.

- To use “ ” or ‘ ’ in a variable we can use escape sequence (\).

Ex:

A screenshot of a cell phone

Description automatically generated

Number datatype

- JS has no type (int, float) for numbers except number is a datatype and it is stored as 64-bit floating-point number.

Boolean datatype

- Falsy values in JS are undefined, null, 0, NaN (Not a Number) and “” (Empty string).

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

Comments

- //single line comment

- /\* multi line comment \*/

Operators

- Mathematical operators: +, -, \*, / and %.

- Conditional operators: <, > <= and >=. Ex: “aa” < “ab” 🡪prints true

- Equality: == and Inequality: != Ex: “2” == 2 🡪prints true, \*the reason is to throw minimum errors. “2” === 2 -->prints false, this === checks type of the variable or value.

- String operators: typeof and instanceof.

Statement and Expressions

- var x = 5; is a statement and is evaluated as undefined in a browser console because it is not returning any value so it is undefined.

- x = 5; is an expression but is evaluated as 5 that is console prints 5 since we assign variable to a value which is not undefined.

/\*The first is a statement, while the second is an expression. While not quite the same, it is similar to C's rules:

// A statement that has no value.

int x = 5;

// An expression...

x = 10;

// ...that can be passed around.

printf("%d\n", x = 15);\*/

\* literals mean string literals and constant means a number.

Control flow statements (if-else, nested if-else, switch)

- if works with range of values.

- switch works with discrete values.

It’s better to use if than switch because switch consecutively executes the cases after the desired expression. To avoid that break; should be used for each case.

- while executes code block when the expression is true.

- for loop is sugar syntax of while loop. JS has another version of for loop which is called for-in loop.

- break and continue are keywords which we can use inside a looping statement. Just fir our convenience.

Ex: break statement

A screenshot of a cell phone

Description automatically generated

Ex: continue statement

A screenshot of a cell phone

Description automatically generated

Functions

- JS is a functional programming language so JS treats all functions as first class citizens.

- Functions are treated as data or objects.

- We can even pass a function as arguments to a function.

- function keyword declares a function.

- function parameters don’t have type information same like var.

- Functions basically returns something, even JS functions returns same like C++ or Java. But in C++ or Java when a function has nothing to return it returns void whereas JS returns undefined.

Ex: Function returns

A screenshot of text

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Block Scope

- A variable defined or declared in a class or function or in a control flow statement then that variable can only be accessed with in the block. This is called block scope.

Block scope

A screenshot of a cell phone

Description automatically generated

- In JS we have function scope in addition to block scope.

Function scope

A screenshot of a cell phone

Description automatically generated

Type Conversion

\*The basic difference between type conversion and type casting, i.e. type conversion is made “automatically” by compiler whereas, type casting is to be “explicitly done” by the programmer. The two terms “type casting” and “type conversion” occur when there is a need to convert one data type to another.

\*Also, operators do more function apart from their normal nature. like the + operator can add to numbers but also appends two strings.

- JS also use this concept of type conversion, to throw minimum error.

Ex: Below the \* and + are treated differently. Though “2” & “3” are strings they are considered numbers when we use \* but when we use + the interpreter appends because + is defined to do the appending operation.

A screenshot of a cell phone

Description automatically generated

JS in a web page

Always put JS codes in a single place instead of using script tags throughout the HTML. It can be done in that way but it is a bad practise. The below html code has lots of script tags it can be written there but it’s better to put code in a single file and link.

A screenshot of a social media post

Description automatically generated

The above picture also shows that the browser will start to execute from head to tail since the html file has multiple script tags. So, the first script tag is executed later the other.

External JS or linking to a HTML

- Using external libraries like JQuery, angular JS etc

- Code can be better organized and reused

- Intellectual property (not easily access to any one)

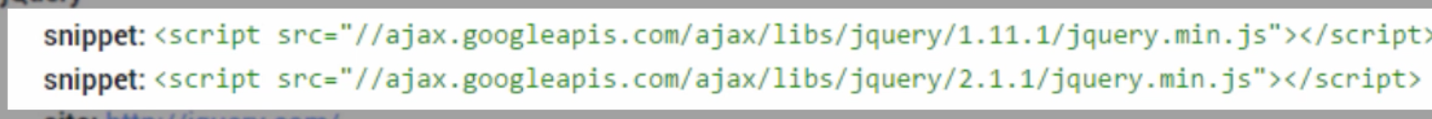
- Cached by browser (download JS from the server and stores in system)

\*Try to use relative path in src attribute of html tag.

A screenshot of a cell phone

Description automatically generated

\*Don’t use the http or https literals in src since there might arise a problem when the html code communicates with the source. If in case the link has http convention but the source as https then warnings or errors may populate. So, it’s better uses the below,



Ex:

A screenshot of a social media post

Description automatically generated

Objects & Array

Objects

- JS provides general object data types basically unordered collection of properties

A screenshot of a cell phone

Description automatically generated

- “color” 🡪 property name and they must be a string but it is not necessary to put in “ “

- red is a value for the property “color”

- A property can hold anything, it could be a primitive value or even objects

\*JSON – Java Script Object Notation

Ex:

A screenshot of a cell phone

Description automatically generated

The object data structure above is called dictionary in python

Empty objects

>var car = {}; //is a object without any properties

<undefined

>car;

<Object {}

- new is a keyword to initialize an empty object i.e without properties. Later properties can be added using (.)dot operator

>var car = new Object(); // var car is initialized with new object

<undefined

>car;// car is empty object

<Object {}

>car.color = “black”; // adding properties to the object

<“red”

>car;

<car = {color : “black”}

\*Difference between instantiation and initialization: Instantiation refers to the allocation of memory to create an instance of a class whereas initialization refers to naming that instance by assigning the variable name to that instance.

- In JS we don’t need not use classes the way we use in C#, Java

- JS supports OOP but not class based, it is prototype based\*

- Another way to create object is to use creaet() method

Ex:

> var car = Object.create(null); // create takes argument so pass null to create a empty object

<undefined

>car;

<{}

Object properties

- Properties can be read and written to an object in some ways one such way is using (.)dot operator and the other way is using associative array.

- Associative array ex,

>var book = new Object();

<undefined

> book[“title”] = “Cracking coding interview”; //Write

< “Cracking coding interview”

>console.log(book[“title”]); //Read

- to delete a property just use delete keyword

Ex:

>delete book.title;

<true

>book;

<{}

- to iterate over object properties in JS, for/in loop is used.

Syntax: for/in loop

for (var x in book) { }

Ex: for/in loop

A screenshot of a cell phone

Description automatically generated

- Normal empty object has additional properties which comes from object prototype.

A screenshot of a cell phone

Description automatically generated

\*Difference between class and prototype : Prototypes vs. Classes. The most important difference between class- and prototype-based inheritance is that a class defines a type which can be instantiated at runtime, whereas a prototype is itself an object instance. ... A class constructor creates an instance of the class.

JSON – Java Script Object Notation

- It is a very popular form of representing data as string.

- JSON is a data format used to send data in strings between client and server.

- There are other data format like XML but JSON is primitively used since the data format is readable.

- JSON data is represented similar to Objects representation in JS but both are different. Where Objects accepts ‘’ while JSON doesn’t accept single quotes ‘’.

\*JSONLint is an open source project which allows you to validate your JSON data. Since when you are working with any programming language and JSON, if your JSON data is not properly formatted, it can cause error, it is better to validate your JSON data beforehand.

A screenshot of a social media post

Description automatically generated

Converting var book\_as\_string – string literals which is in JSON format to Objects using JSON.pasre() method

A screenshot of a social media post

Description automatically generated

\*

- XML are really a big deal to parse data in JS. And this is why exactly JSON’s are prominent.

- JS objects can also be converted to JSON

\* The default conversion from an object to string is "[object Object]"

- What is the output of the following code snippet?

>({title : "The Three Musketeers", author: "Alexandre Dumas", price: "$49"}).toString()

<[object Object]

Global Object

- Like other programming languages JS also links functions but it happens quite different here.

- JS has no linker but has globalThis object, which ties everything together.

- In the browser environment globalThis is the window object, so when the new pages load up JS interpreter starts and creates a new window and put all properties.

A screenshot of a cell phone

Description automatically generated

- JS interpreter always has a globalThis object but in case of a browser globalThis is called Window.

Both are same 🡪

>global === window

<true

- the variable, functions and objects which is created will become part of Window object.

A screenshot of a cell phone

Description automatically generated

Arrays in JS

General definition of an array: Array is basically a data structure or container object that holds a fixed number of items of specific types. Memory layout of an array is contiguous

But in JS Arrays don’t follow their general nature.

- They can be initialized with zero element.

- items can be added to the array later. That is it grows dynamically.

- obviously they don’t have a data type. Hence we can store any type of element.

A screenshot of a cell phone

Description automatically generated

Sparse(scattered) Array

In JS we can load values to any index without error and warnings.

A screenshot of a cell phone

Description automatically generated

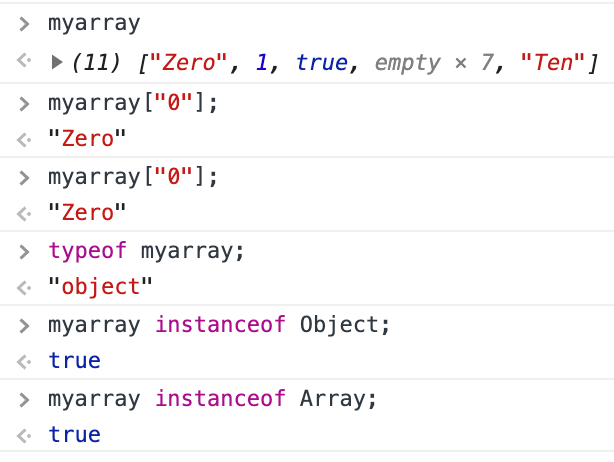
In other programming languages array elements should be adjacent to each other. But JS overrules the traditional rule.

Array as objects

- The reason why arrays are so convenient in JS is because they don’t have any underlying data structure. They use objects has their data structure.

- Arrays work like this🡪 Array indices are converted to strings and these strings will hold values. Exactly like properties in object, where properties are strings. can hold. Hence JS just makes up array but in fundamentally it is an object.

The below code snippet shows array as objects,



Array is a function but remember a function can also be an Object in JS.

A close up of a black background

Description automatically generated

Here the myarray takes a string as indices but still add the values to it.

A close up of a screen

Description automatically generated

Length of n array