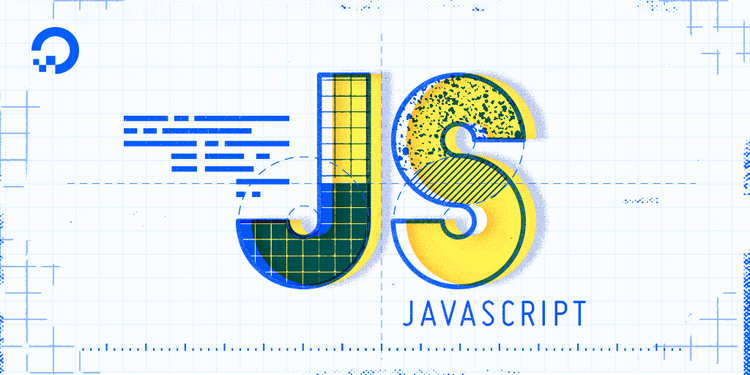
javascript



KiranKumar

Test Yantra Software Solutions Pvt Ltd

Bengaluru.

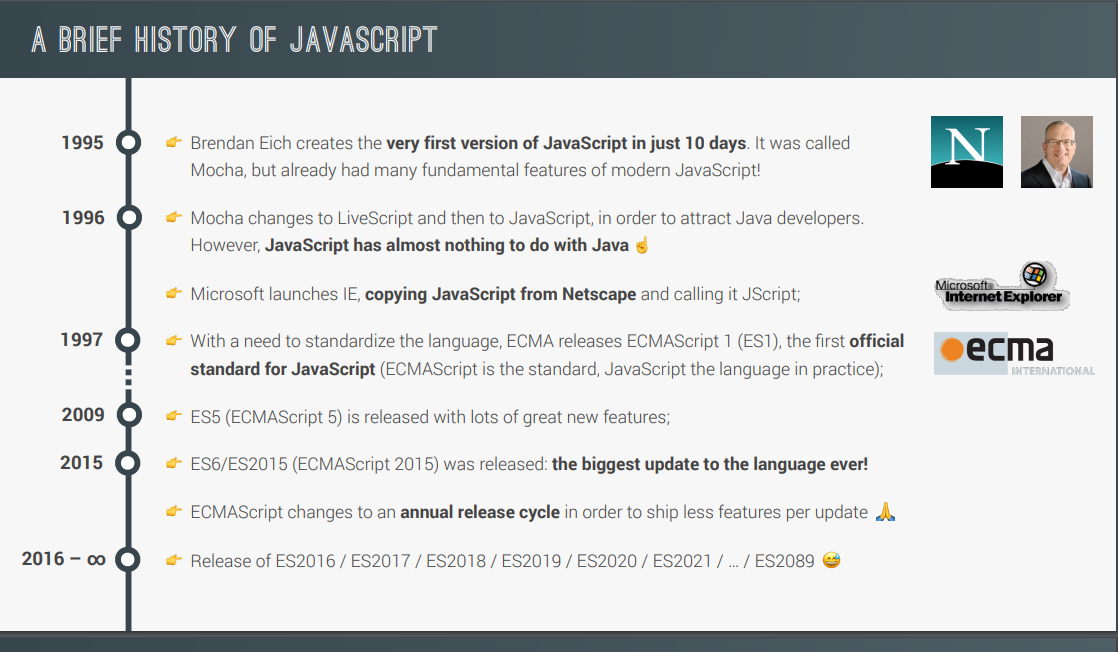
TAble of  
Contents

1. Introduction to JavaScript……………………………………………………………1
2. Conditions and Looping Statement 2

3.

*JavaScript*

JavaScript is a scripting language that enable you to create dynamically updating content, control multimedia, animate images.



*Features of JavaScript:*

1. JavaScript is a scripting language (interpretation and execution occurs at the run time)
2. is also called interpreted programming language (as it uses interpreter for code translations)
3. JavaScript is case sensitive
4. using JavaScript one can create interactive webpages/dynamic web pages
5. It is Loosely Typed Language.

*Why it is called as Full Stack data File?*

JavaScript is the universal language that can be used across all software layers.`

* For client side (browser) we use core Js (vanilla js).
* In webserver side we can use NodeJS
* In database we can use mongo DB and Couch DB.

*Libraries built by using JavaScript*

Libraries are built to ease Programming tasks.

J-Query, Load#, Bootstrap, \_JS Etc.

*Frameworks built using JavaScript*

**Node JS** - for developing web applications.

**React JS** - for developing web applications.

**React Native** - for developing client server applications.

**Angular JS** - for developing single page web applications. (Gmail, google maps)

**Electron JS** - for developing standalone applications.

**Tensor flow** -for developing AI, machine learning applications.

*Softwares Required:*

**Node.Js:** It is Open-Source, Cross-Platform Back-end JavaScript runtime environment. (It Creates Environment to run script).

Version Node.js: 16.14.2

**Visual Studio Code:** It is used write and execute JS code.

Version: 1.71.2 (user setup)

*Steps to launch VS Code By Command Prompt:*

Create a New Folder -> enter cmd in path tab

(Command prompt will be opened)

Type “code .” and press Enter.

(VS code will be Opened)

*JavaScript Tokens:*

Smallest unit of the code is called tokens.

1. Keywords
2. Identifiers
3. Literals
4. Operators
5. Separators
6. **Keywords**

* Keywords are the predefined words with predefined meaning in it (developers has defined their meaning already)

*e.g. break, case, catch, continue, debugger, default, delete, do , else , finally , for , function , if , in , instance of , new , return , switch , this , throw , try , type of , var , void , while , and with*

Rules to write the keywords:

🡪JavaScript is a case sensitive language.

🡪We should use keywords in lowercase letters.

🡪We cannot use keywords as identifiers.

1. **Identifiers**

* JavaScript Identifiers are names given to variables, functions, etc

Rules to write the Identifiers:

🡪 You should not use any of the JavaScript reserved keywords as a variable name. For example, break or Boolean variable names are not valid.

🡪 JavaScript variable names should not start with a numeral (0-9). They must begin with a letter or an underscore or dollar character. For example, 5demo is an invalid variable name but \_5demo is a valid one.

🡪 JavaScript variable names are case-sensitive. For example, Name and name are two different variables.

1. **Literals**

* Values what we store inside the memory allocated.

eg. numeric literals –7, 5, 8

string literals – “7”, “5”, “8”, “hello”

Boolean literals – true, false

null

undefined

1. **Operators:**

Used to perform some mathematical operations e.g. (+, \_, \*, \*\*, /, %) etc.

1. **Separators:**

Used to separate the statements in the program e.g. (, (), {}, [],) etc.

*JavaScript Engines:*

It is basically an Interpreter and it is a software component that executes JavaScript code. The first JavaScript engines were mere interpreters, but all browsers having Built-in JS Engines.

Eg: Chrome browser 🡪 v8.

Mozilla Firefox 🡪 SpiderMonkey.

IE 🡪 Chakra.

Safari 🡪 JavaScriptCore.

Declaring a variable in JavaScript:

Variables are the containers that you can store the values in it

Syntax:

var(keyword) name(identifier) = value

Followed by

var a // declaration

a = 10 // Initialization

console.log(a) // Utilization

*DataTypes:*

There are 2 types of datatypes

1. Primitive datatypes
2. Non-primitive datatypes (object references)

1.Primitive datatypes/immutable:

* + - * Undefined
      * [Boolean](https://developer.mozilla.org/en-US/docs/Glossary/Boolean)
      * [Number](https://developer.mozilla.org/en-US/docs/Glossary/Number)
      * [String](https://developer.mozilla.org/en-US/docs/Glossary/String)
      * null
    1. Non-primitive datatypes (object references)/mutable:
* Objects
* Functions
* Arrays
* Date
* Math

*Flow Control Statements:*

1. **Conditional Statements**

* If
* If else
* Else if
* Switch

1. **Looping Statements**

* While
* Do while
* For
* For in
* For of
* For each(Method)

*1.Conidtional Statement:*

Conditional statements control behavior in JavaScript and determine whether or not pieces of code can run.

“If” statements: where if a condition is true it is used to specify execution for a block of code.

var a='kiran'

var b='kumar'

if(a==b)

{

    console.log('a and b are same')

}

“Else” statements: where if the same condition is false it specifies the execution for a block of code.

var a='kiran'

var b='kumar'

if(a==b)

{

    console.log('a and b are same')

}

else

{

    console.log('a and b r not same')

}

“Else if” statements: this specifies a new test if the first condition is false.

var a=1;

var b=6;

var c =1;

if(a>b && a>c)

{

    console.log ('a is greater')

}

else if(b>a && b>c)

{

    console.log ('b is greater')

}

else

{

    console.log ('c is greater')

}

Switch Statement : This will switch to particular mentioned case.

var names="kiran"

switch (names) {

    case 'kiran':

        console.log('im kiran')

        break;

        case 'kumar':

            console.log('kumar')

        break;

    default:

        console.log('im neither kumar nor kiran')

        break;

}

*1.Looping Statement:*

***While Loop:*** Loops through a block of code while a condition is true.

***Do-While Loop***: The do...while statement will execute a block of code once, and then it will repeat the loop while a condition is true.

***For Loop:*** The for statement will execute a block of code a specified number of times.

1)Programme execution in memory:

2) Function execution in memory

**DataTypes(Var let const):**

|  |  |  |
| --- | --- | --- |
| **DataType** | **Re-Initialization** | **Re-Declaration** |
| Var | Is possible | Is Possible |
| let | Is possible | Is not possible |
| const | Is not possible | Is not possible. |

**Array Methods:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Method**  **Name** | **Its Function** | **Arguments** | **Returns** | **Modifies O. value or no** |
| 1 | concat() | Combines 2 arrays | Array or String | Array | No |
| 2 | push() | It will add the element at the end of the array | Element | Length of original array | Yes |
| 3 | pop() | It will remove the last element in the array | Element | Element | Yes |
| 4 | Unshift() | It will add the element in the beginning of the array | Element | Length of modified array | Yes |
| 5 | Shift() | It will remove the first element from the array | Element | Element | Yes |
| 6 | Splice() | It will delete and as well add the given element to the specific index of an array. | Start index, Delete count, new Element | Deleted Element | Yes |
| 7 | Slice() | It will take out part of array | Start index , end index | Part of array | No |
| 8 | Every() | Compares the each and every element of array with specified condition (returns true only if all conditions are specified) | \*Function | Boolean | No |
| 9 | some() | Compares all value and returns true id any one of the element satisfying condition | \*Function | Boolean | No |
| 10 | indexOf() | Returns the index of an element (left to right) | (element, Fromindex) | Index no | No |
| 11 | lastIndexof() | Returns the index of an element (right to left) | (element,index) | Index | No |
| 12 | Reverse() | Reverse the array | ------ | Reversed array | Yes |
| 13 | Includes() | Search the element in an array | Element | Boolean | No |
| 14 | Join() | Joins the element to all the elements present in array. | Element | Concated elements with array | No |
| 15 | Foreach() | Perform operation mentioned in callback function with all the element in array | Callback function  (element, index) | Modified array | No |
| 16 | Map() |  |  |  |  |
| 17 | Filter() |  |  |  |  |
| 18 | Sort() | Sort the array in ascending , descending order based on condition in callback function. | (a,b) | Ascending array if Return(a-b)  Descending array if Return (b-a) . | yes |
| 19 | Reduce() | Reduce the array element from total no. |  |  |  |
| 20 | Reduceright() |  |  |  |  |

**String Methods:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Method**  **Name** | **Its Function** | **Arguments** | **Returns** | **Modifies O. value or no** |
|  | Length() | Returns the length of the String | No args | Length | No |
| 2 | Split() | Separates string | Separators | Separated string in an array | no |
| 3 | charAt() | Returns the character of specified index no | Index no | Character | No |
| 4 | Concat() | Conacats element the string | Element | Concated string | No |
| 5 | endsWith() | Checks string with last character | Character | Boolean | No |
| 6 | Includes() | Checks whether a string having specified string or no | String or character | Boolean | No |
| 7 | indexOf() | Checks the index of specified String | Character | Index no | No |
| 8 | lastIndexof() |  | character | Index no | No |
| 9 | Replace() | Replces a string with given string | (old string ,required string) | Complete changed String | No |
| 10 | Repeat () | Repeats the string by specified number of times | Number | Repeated strings | No |
| 11 | Trim() | Trims the space | No args | String | No |

**Math Methods:**

1. Math.abs (): It will ignore last zeros of decimal and takes absolute value

/ absolute functions

//C It will ignore last zeros of decimal and takes absolute value

console.log (Math.abs (31.123000)) //Output: 31.123

console.log (Math.abs (31300)) //Output:31300

1. Math.round(): It will round off the number according to decimal value.

// round function

// It will round off the number according to decimal value.

console.log(Math.round(3.67))  // Op = 4

console.log(Math.round(3.49)) // Op =3

1. Math.min(int1,int2,int3): It will compare the values and gives the minimum value among them.

//min function

//It will give the min value among mentioned values

console.log(Math.min(20.9,36)); //Op =20.9

1. Math.max(int1,int2,int3): It will compare the values and gives the maximum value among them.

//max function

//It will give the max value among mentioned values

console.log(Math.max(20.9,36)); //Op =36

1. Math.floor(int) :It will neglect the decimal value and gives the whole no.

//floor function

// it will give the value neglecting the decimal value

console.log(Math.floor(50.1));  //50

console.log(Math.floor(50.9));  //50

1. Math.ceil() : It will give the next value.

//ceil function

// It will give the next value

console.log(Math.ceil(50.1));  //  Op=51

console.log(Math.ceil(50.9))   // Op=51

1. Math.random() :It Will generate random number.

// random function

// It Will generate random number

console.log(Math.random());