Swipe

BECOME A JAVASCRIPT DEVELOPER

JS

[In 10 Minutes...]

Adil
@adil.codes

Asifa Malik
@tech_malik92

JS Intro

" Used to make web pages interactive. "

⭐ File Extension: (file).js

⭐ Link with Html:

→ 2 ways to link Js with HTML file:

- "External Js".
- "Internal Js" using the <script> tag.

Variable

" Container used to store data (of different types). "

⭐ 3 Ways to create it:

⭐ With var keyword

```
var Name = data-of-any-data-type;
```

⭐ With let keyword

```
let Name = data-of-any-data-type;
```

⭐ With const keyword

```
const Name = data-of-any-data-type;
```

Data Types

" Type of a particular data is called data type. "

⭐ 7 data-types in Js

- ☑ String → var x = "Adil";
- ☑ Number→ var x = 10;
- ☑ Boolean→ var x = true/false;
- ☑ Array→ var x = [1,"a",true];
- ☑ Object→ var x = {prop:"val"};
- ☑ Null→ var x = null;
- ☑ Undefined→ var x;

Operators

" Symbols to perform operations on data. "

⭐ 4 operator types in Js

- ☑ Arithmetic Operators (+,-,*...)
- ☑ Assignment Operators(=,+=,...)
- ☑ Comparison Operators(>,==,...)
- ☑ Logical Operators(&&,||,!)

```
var a = 10;
var b = 20;
var c = a+b;
```

→EXAMPLE

Comments

" Statements that are not executed while execution. "

⭐ 2 ways to write them:

⭐ Single Line Comment

```
// single line
```

⭐ Multi-Line Comment

```
/* multi-line
comment */
```

Print/write

" Printing or writing something on a screen. "

⭐ 2 ways to Print/write:

⭐ On Browser Console

```
console.log("anything");
```

⭐ On document/screen

```
document.write("anything");
```

Functions

" Block of code that performs a specific task. "

⭐ 2 ways to write them

⭐ Regular Function

```
let name = function(parameters)
{/*code block*/}
```

⭐ Arrow Function

```
let name = (parameters) =>
{/*code block*/}
```

Loops

" Used for repeated execution of code until a certain condition. "

⭐ 3 types of loops in Js:

⭐ for loop, ⭐ while loop, ⭐ do while loop.

```
for(init;condition;var++){}
----
while(condition){}
----
do{code}while(condition)
```

Arrays

" Collection of data stored under the same name. "

⭐ Data could be of any type

⭐ Syntax:

```
let arrayName = [item1,item2,...];
```

⭐ location of an item in an array is called index.

⭐ Multi-dimensional array:
" array within an array "

Objects

" Collection of data store in form of name/value pairs. "

⭐ Syntax:

```
let objectName = {name:"value"};
```

Property value

⭐ Access object values:

```
objectName.propertyName;
```

if/else

" executes the code if the condition is true, else another code. "

⭐ Syntax:

```
if(condition == true){
  // execute this code
}else{
  // execute this code
}
```

Switch Statement

It has 1 expression and some cases. cases compare with expression.

Any case match with exp, that case code runs.

```
switch(expression) {
  case x:
    // code block
    break;
  case y:
    // code block
    break;
  default:
    // code block
}
```

Events

"Action occurs when user interact with the web page"

⭐ Example: button clicked

⭐ Syntax:

```
<element onevent='some JavaScript'>
```

oneventname
e.g: onclick

Js code to be executes

Jump Statements

" Statements to transfer the control from one place to another. "

⭐ return value;
return value to the calling code.

⭐ break;
stops the execution.

⭐ continue;
control jumps to beginning of loop

Template Strings

" New way to write strings using back-ticks (` `). "

⭐ Syntax:

```
let str = `Hello`;
```

⭐ Easy Interpolation (insertion of variables in the string).

```
let name = "Adil";
let str = `Hello ${name}`;
//output:Hello Adil
```

this Keyword

" this is a pre-defined word that always points to Js object"

⭐ Globally it represents the window object.

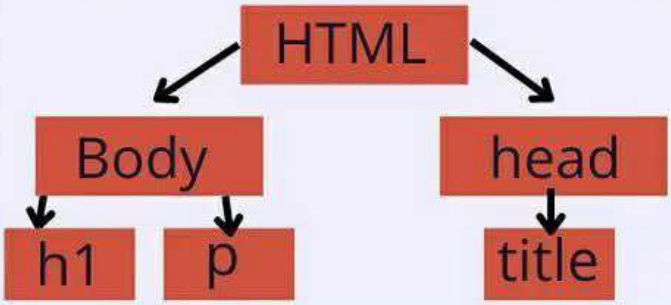
⭐ In function, it represents the window object.

⭐ In object method, it represents the object.

DOM

"Document Object Model"

- ⭐ It looks like a Tree
- ⭐ Tree of all html elements
- ⭐ Everything(element, text, attribute) is called Node.



Dom Methods

" These are the methods to access the DOM node or element"

```
document.method();
```

- ⭐ .getElementById
- ⭐ .getElementsByTagName
- ⭐ .querySelector

BOM

"Browser Object Model"

- ⭐ It looks like a Tree
- ⭐ Tree of all the browser objects
- ⭐ This include the info about the browser.

Bom Methods

" These are the methods to access the BOM elements "

```
window.method();
```

- ⭐ .open()
- ⭐ .resizeTo()
- ⭐ .close()

Closures

"feature in JS where inner function has access to outer function's variables. "

⭐ Example:

```
function hello(){
 var a = 10;
 var b = 20;
 function print(){
     var c = a+b;
     console.log(c);
 }
 print();
}
hello();
```

CallBacks()

" function passed to another function as an argument. "

```
function print1(callback){
   console.log("hello");
   callback();
}
function print2(){
   console.log("World");
}
print1(print2);
```

Modules

" allow you to split code into separate files. "

⭐ Modules rely on import and export statements.

⭐ Send data using export and get data using import

```
export const name = "Jesse";
```

```
import {name,age} from "./file.js";
```

Promises

" Is an object that links the producing code and consuming code. "

```
let myPromise = new
Promise(function(myResolve, myReject) {
// "Producing Code"

  myResolve();//successful
  myReject(); //error
});

// "Consuming Code"
myPromise.then(
  function(value) { /* code if successful
*/ },
  function(error) { /* code if some error
*/ }
);
```

Async

"async is just a simple function returning a promise. "

```
async function mf() {
  return "Hello";
}
```

is same as:

```
function mf() {
  return Promise.resolve("Hello");
}
```

Await

" awaits makes the function wait for a promise. "

```
let value = await promise;
```

await simply puts the statement on wait

Strict Mode

" makes your js coding environment strict. "

- Syntax `"use strict";`
- Code will be executed in strict mode.
- No undeclare variables can be used
- Makes you code more accurate

JS Projects

- Js Stopwatch
- To-do List
- Weather app
- Tic Tac Toe app
- Music Player