

Javascript

A scripting programming language



January 26, 2022

Programming Hero

Level-4, 34, Awal Center, Banani, Dhaka

# What is JavaScript?

JavaScript is a scripting or programming language that allows you to implement complex features on web pages.

# Data Types of JavaScript

1. Numeric data type
2. String data type
3. Boolean data type

## Numeric data type

Integer numbers like: -1, -123, 0, 1, 54, 2022.

Float numbers like: -72.43, 3.1416.

## String data type

All kind of characters (A-Z) including symbols like: Hasib, B, Ma, Father, baby-samira, halsey.

All characters are quoted through single quotation (‘’) or double quotation (“”) like: ‘father’, “Mother”.

## Boolean data type

It simply means a straight forward result which can be either true or false which might be represented with straight keyword `true` or `false`.

# JavaScript reserved keywords



# JavaScript variable naming style

* A variable never uses reserved keywords
* A long variable can use underscore (\_) like: var\_name, producer\_\_name e.t.c
* A long variable can use camel case letter like: varName, whoAmI, JavaScript e.t.c.
* Although it can use a dollar ($) sign like: $varName, $var\_name e.t.c.
* Without these all styles are invalid.

# Push and Pop elements from an array

## What is `push`?

‘Push’ used to add an element at the last of an array.

Prototype: var\_name.push(item);

## What is `pop`?

‘Pop’ used to remove an element from the last of an array but it returns the removed element which can be store to a variable.

Prototype: var\_name.pop(); [returns the popped item]

## What is `unshift`?

‘Unshift’ used to add an element at the first of an array.

Prototype: var\_name.unshift(item);

## What is `shift`?

‘Shift’ used to remove an element from the first of an array but it returns the removed element which can be stored to a variable.

Prototype: var\_name.shift(); [returns the popped item]

## What is `splice` while adding?

‘splice’ used to add an element at a specific index of an array. In adding approach it’s not return any value.

Prototype: var\_name.splice(target\_index, delete\_count, item);

### What is `delete\_count` while adding?

Give access the new item of an element is that get replaced or removing other item/items means if ‘delete\_count’ is “0” then the new item won’t delete other items/items or replace, if ‘delete\_count’ is “1” then the new item replace the current item which exist on that mentioned index before or if ‘delete\_count’ is more than “1” then the new item get replaced with the current and start deleting other items after the new item based the number of ‘delete\_count’.

## What is `splice` while removing?

‘splice’ used to remove an element at a specific index of an array. In removing approach, it returns popped item which can be stored to a variable.

Prototype: var\_name.splice(target\_index, delete\_count); [returns the popped item]

### What is `delete\_count` while removing?

Definition & properties are as similar as [this.](#_What_is_`delete_count`)

# Show number with `leading zero` less than 10

Sometimes we may face some numbers which are less than 10 as they are single numbers, we need to add a leading zero before them. For that instance, we approach following scheme.

[i.e.: 01, 02, …, 09]

Prototype: (‘0’ + var\_name).slice(-2);

# Print without `trailing new line`

Most of the time we face a common situation which is print something which won’t need any new line for trailing new line we use following prototype.

Prototype: process.stdout.write(printing\_statement);

# `Clear Console` using JS command

Just type the following:

Prototype: console.clear();

# Declare `objects` with properties

An object contains a pair of properties.

Prototype: var var\_name = {

object\_name1: object\_value1 [if string use ‘’ or ””],

object\_name2: object\_value2 [if string use ‘’ or ””],

object\_name3: object\_value3 [if string use ‘’ or ””],

object\_name4: object\_value4 [if string use ‘’ or ””],

object\_name5: object\_value5 [if string use ‘’ or ””],

object\_nameN: object\_valueN [if string use ‘’ or ””]

};

## Get object and object properties

Simply type object name then use dot (.) so that access object property and following is the procedure.

Prototype:

console.log(object\_name);

console.log(object\_name.object\_property);

## Set object and object properties

Set in this case generally means update object name or update property’s value.

### Rename `object`

Prototype: new\_object\_name = old\_object\_name;

### [Update](Module%2019-Core%20Concepts%20functions%20and%20objects/basic__object.js) object’s `property`

#### Method 1:

Prototype: object\_name.object\_property = value/new\_value; [if string use ‘’ or “”]

#### Method 2:

Prototype:

new\_object\_name = old\_object\_name.old\_object\_property;

new\_object\_name = value/new\_value;

#### Method 3:

Prototype:

new\_object\_name = old\_object\_name[“object\_property”];

new\_object\_name = valu/new\_value;

#### Method 4:

Prototype: object\_name[object\_property] = value/new\_value;

#### Method 5:

Prototype:

new\_object\_name = ‘object\_property’;

old\_object\_name[new\_object\_name] = value/new\_value;

# 3 difference kinds of variable types

## Variable type: `var`

* Default variable declaring type
* ‘Function’ scoped and ‘Block’ scoped
* ‘Globally’ scoped and ‘Locally’ scoped
* No conflict in ‘Redeclaration’
* By default, initialized with ‘undefined’

## Variable type: `let`

* Custom variable declaring type
* ‘Block’ scoped only
* ‘Locally’ scoped only
* Must create conflicts while ‘Redeclaration’
* By default, not initialized

## Variable type: `const`

* Constant value like, PIE: 3.1416, g: 9.8 etc.
* Not changeable
* Once declare with value means fixed declaration
* ‘Globally’ scoped and ‘Locally’ scoped

## Difference between `var` and `let`

Click [here](Module%2020-Apply%20JavaScript%20Concepts/var__let__const.js) for the proper illustration.

# About `array`

## What is an `array`?

Array is a set of elements.

Prototype: data\_type array\_name = [‘set\_of\_strings’ or ‘set\_of\_numbers’]; [comma separation applicable]

## Passing `array` to a function

There is two ways to pass an array to a function.

### First define then pass the defined `array variable`:

Prototype: data\_type array\_name = [set\_of\_elements];

function\_name(array\_name);

## `Directly Pass Array` to a function:

Just use ‘[set\_of\_elements]’ within the parenthesis.

Prototype: function\_name([set\_of\_elements]);