Hands-on 1: … Operator

Code for: 2 arrays merged with Spread Operator

<script>

   var array1 = [10, 20, 30, 40, 50];

   var array2 = [60, 70, 80, 90, 100];

   var array3 = [...array1, ...array2];

console.log(array3);

</script>

OUTPUT:

[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

Hands-on 2: … Operator

Code for: Copy object using Spread Operator

<script>

const obj = {

firstname: "Divit",

lastname: "Patidar",

};

const obj2 = { ...obj };

console.log(obj2);

</script>

OUTPUT:

{

firstname: "Divit",

lastname: "Patidar"

}

Hands-on 3: … Operator

Code for: Multiple element as argument using Rest Operator

<script>

function average(...args) {

console.log(args);

var avg =

args.reduce(function (a, b) {

return a + b;

}, 0) / args.length;

return avg;

}

console.log("average of numbers is : "

+ average(1, 2, 3, 4, 5));

console.log("average of numbers is : "

+ average(1, 2, 3));

</script>

OUTPUT:

[1, 2, 3, 4, 5]

"average of numbers is : 3"

[1, 2, 3]

"average of numbers is : 2"

Hands-on 4: For…of with array

// array

const students = ['John', 'Sara', 'Jack'];

// using for...of

for ( let element of students ) {

// display the values

console.log(element);

}

OUTPUT:

John

Sara

Jack

Hands-on 5: For…of with String

// string

const string = 'code';

// using for...of loop

for (let i of string) {

console.log(i);

}

OUTPUT:

Code

Hands-on 6: For…of with Sets

// define Set

const set = new Set([1, 2, 3]);

// looping through Set

for (let i of set) {

console.log(i);

}

OUTPUT:

1

2

3

Hands-on 7: For…of with Maps

// define Map

let map = new Map();

// inserting elements

map.set('name', 'Jack');

map.set('age', '27');

// looping through Map

for (let [key, value] of map) {

console.log(key + '- ' + value);

}

OUTPUT:

Name- Jack

Age-27

Hands-on 8: Create a basic map object

let map1 = new Map([

[1, 2],

[2, 3],

[4, 5]

]);

console.log("Map1");

console.log(map1);

let map2 = new Map([

["firstname", "sumit"],

["lastname", "ghosh"],

["website", "EDUNETFOUNDATION"]

]);

console.log("Map2");

console.log(map2);

OUTPUT:

Map1

Map(3) {1 => 2, 2 => 3, 4 => 5}

Map2

Map(3) {'firstname' => 'sumit', 'lastname' => 'ghosh', 'website' => 'EDUNETFOUNDATION'}

Hands-on 9: Create a new set Object

let set1 = new Set();

set1.add(10);

set1.add(20);

// As this method returns

// the set object hence chaining

// of add method can be done.

set1.add(30).add(40).add(50);

console.log(set1);

OUTPUT:

Set(5) {10, 20, 30, 40, 50}

Hands-on 10: Create a basic promise object

let promise = new Promise(function (resolve, reject) {

const x = "edunetfoundation";

const y = " edunetfoundation"

if (x === y) {

resolve();

} else {

reject();

}

});

promise.

then(function () {

console.log('Success’);

}).

catch(function () {

console.log('Some error has occurred');

});

OUTPUT:

Success

Hands-on 11: Then method handles when a promise is resolved

let promise = new Promise(function (resolve, reject) {

resolve('EDUNETFOUNDATION');

})

promise

.then(function (successMessage) {

//success handler function is invoked

console.log(successMessage);

}, function (errorMessage) {

console.log(errorMessage);

});

OUTPUT:

EDUNETFOUNDATION

Hands-on 12: JS Modules

Index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Document</title>

<style>

body {

font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;

}

.result {

font-size: 18px;

font-weight: 500;

color:blueviolet;

}

</style>

</head>

<body>

<h1>JavaScript Modules</h1>

<button class="Btn">IMPORT</button>

<div class="result"></div>

<h3>Click on the above button to import module</h3>

<script src="script.js" type="module"></script>

<script src="sample.js" type="module"></script>

</body>

</html>

Script.js

import test from './sample.js';

document.querySelector('.Btn').addEventListener('click',()=>{

test();

})

Sample.js

let resultEle = document.querySelector(".result");

export default function testImport(){

resultEle.innerHTML = 'Module testImport has been imported';

}

OUTPUT:

