Java Script Basic

Table of Contents

[Links 3](#_Toc25787599)

[Comment 3](#_Toc25787600)

[Console 3](#_Toc25787601)

[Vaiables 3](#_Toc25787602)

[Let vs Var 3](#_Toc25787603)

[Value Types 3](#_Toc25787604)

[Reference Types 3](#_Toc25787605)

[Arrays 3](#_Toc25787606)

[Clear 4](#_Toc25787607)

[Iter on array 4](#_Toc25787608)

[Sort 4](#_Toc25787609)

[Every – check all element in array for condition 4](#_Toc25787610)

[Some – check if at least one element in array for condition 4](#_Toc25787611)

[Filter 5](#_Toc25787612)

[Functions 5](#_Toc25787613)

[function as object 5](#_Toc25787614)

[rest arguments 5](#_Toc25787615)

[Operators 5](#_Toc25787616)

[Equality 5](#_Toc25787617)

[Switch 5](#_Toc25787618)

[Loops 6](#_Toc25787619)

[for 6](#_Toc25787620)

[while 6](#_Toc25787621)

[do while 6](#_Toc25787622)

[Iter on object 6](#_Toc25787623)

[Iter on array 6](#_Toc25787624)

[for of 6](#_Toc25787625)

[Objects 6](#_Toc25787626)

[Factory function 7](#_Toc25787627)

[Constructor function 7](#_Toc25787628)

[dynamic objects 7](#_Toc25787629)

[Clone object 7](#_Toc25787630)

[Getter and Setter 7](#_Toc25787631)

[Template Literals 8](#_Toc25787632)

[Date 8](#_Toc25787633)

[Try – Catch 8](#_Toc25787634)

# Links

<https://www.w3schools.com/js/default.asp>

<https://developer.mozilla.org/en-US/docs/Web/JavaScript>

# Comment

//

/\*

\*/

# Console

console.log(“print to console”);

# Vaiables

let name = “Ronnie”;

const birthday = “June 15, 1970”;

# Let vs Var

let block scope variable

var

in function, function scope, even out of block

out of function, global scope, added to window object

# Value Types

String

Number

Boolean

Undefined

Null

Symbol

# Reference Types

Object

Function

Array

# Arrays

let selectedColors = [‘red’, ‘blue’];

selectedColors[0];

selectedColors[2] = “green”;

selectedColors[3] = 5;

selectedColors.length;

let numbers = [3, 4]; // [3, 4]

numbers.push(5, 6); // [3, 4, ,5, 6]

numbers.unshift(1, 2); // [1, 2, 3, 4, ,5, 6]

numbers.splice(2, 0, “a”, “b”); // [1, 2, “a”, “b”, 3, 4, ,5, 6]

numbers.pop(); // [1, 2, “a”, “b”, 3, 4, ,5]

numbers.shift(); // [2, “a”, “b”, 3, 4, ,5]

numbers.splice(2, 1); // [2, “a”, 3, 4, ,5]

indexOf(“0”) // -1

indexOf(“3”) // -1

indexOf(“3”) // 4

const circle = circles.find(function(circle) {

return circle.name === “a”

});

## Clear

Numbers = []; // leave reference

numbers.length = 0;

## Iter on array

let colors = [‘red’, ‘blue’, ‘green’];

for (let index in colors) {

console.log(index); // prints 0, 1, 2

console.log(colors[index]); // prints red, blue, green

}

numbers.forEach((number) => {

console.log(number); // prints red, blue, green

}

## Sort

const courses = [

{ id: 1, name: “Node.js” },

{ id: 2, name: “JavaScript” }

];

courses.sort(function(a, b) {

// a < b => -1

// a > b => 1

// a === b => 0

If (a.name < b.name) return -1;

If (a.name > b.name) return 1;

Return 0;

});

## Every – check all element in array for condition

const numbers = [1, 2, 3];

const allPositive = numbers.every(function(value) {

return value >= 0;

});

## Some – check if at least one element in array for condition

const numbers = [1, 2, 3];

const atLeastOnePositive = numbers.some(function(value) {

return value >= 0;

});

## Filter

const numbers = [1, -1, 2, 3];

const filtered = numbers.filter(function(value) {

return value >= 0;

});

# Functions

function greet(name) {

console.log(“do something” + name);

return 4;

}

greet(“Ronnie”);

## function as object

let run = function() {

console.log(“run”);

};

run();

## rest arguments

function sum(…args) {

for (arg of args) {

// iter args

}

}

# Operators

let x = 10;

console.log(x); // 10

console.log(++x); // 10

console.log(x); // 11

console.log(x++); // 12

console.log(x); // 12

x += 5;

x \*= 5;

# Equality

Strict equality – Type and Value

1 === 1 // true

‘1’ === 1 // false

Lose equality - Value

1 == 1 // true

‘1’ == 1 // true

# Switch

Switch (role) {

Case “guest”:

console.log(“Guest”)

break;;

Case “moderator”:

console.log(“Moderator”)

break;;

default:

console.log(“Default”)

}

# Loops

## for

for (let i = 0; i < 5; i++) {

console.log(“Guest”)

}

## while

while (i < 5) {

console.log(“Guest”)

}

## do while

do {

console.log(“Guest”)

} while (i < 5)

## Iter on object

const person = {

name: “Ronnie”,

age: 30

};

For (let key in person) {

console.log(key); // prints name, then age

console.log(person[key]); // prints Ronnie, then 30

}

## Iter on array

let colors = [‘red’, ‘blue’, ‘green’];

for (let index in colors) {

console.log(index); // prints 0, 1, 2

console.log(colors[index]); // prints red, blue, green

}

## for of

let colors = [‘red’, ‘blue’, ‘green’];

for (let color of colors) {

console.log(color); // prints red, blue, green

}

# Objects

const circle = {

radius: 1,

location: {

x: 1,

y: 1

},

isVisible: true,

draw: function() {

console.log(“draw”);

}

};

circle.radius = 2;

circle[“radius”] = 2;

circle.draw();

## Factory function

function createCircle(radius) {

return {

radius,

draw() {

console.log(“draw”);

}

};

}

const circle = createCircle(5);

## Constructor function

function Circle(radius) {

this.radius = radius;

this.draw = function() {

console.log(“draw”);

}

}

const circle = new Circle(5);

## dynamic objects

const circle = {

radius: 1

};

circle.color = “yellow”;

circle.draw = function() { }

delete circle.color;

delete circle.draw;

## Clone object

const circle2 = Object.assign( {}, circle); // add object to existing object

const circle2 = { …circle }; // clone object

## Getter and Setter

const person = {

name: “Ronnnie”,

age: 30,

get Age() {

return age;

}

set Age(value) {

age = value;

}

};

person.Age = 30;

console.log(person.Age);

# Template Literals

const name = “John”;

const another =

`Hi ${name},

This is Ronnie`;

# Date

const now = new Date();

const date1 = new Date(“June 15 1970 12:00”);

const date2 = new Date(1970, 5, 15, 12, 0, 0);

# Try – Catch

throw new Error(“index out of bound”);

try {

} catch (e) {

console.log(e);

}