**Output to Screen**

- window.alert()

- document.write() لطباعة أي شي على الصفحة

- console.log()

--------------------------------------------------------------------------------------------------------------------

**Console Methods**

- log

- error

- table

- group()

- groupEnd()

--------------------------------------------------------------------------------------------------------------------

**Data Types Intro**

- String " "

- Number 52

- Array => Object a[]

- Object {key => value }

- Boolean true – false

- null فاضي

- undefiend غير موجود

console.log("Osama Mohamed");

console.log(typeof "Osama Mohamed");

console.log(typeof 5000);

console.log(typeof 5000.99);

console.log(typeof [10, 15, 17]);

console.log(typeof { name: "Osama", age: 17, country: "Eg" });

console.log(typeof true);

console.log(typeof false);

console.log(typeof undefined);

console.log(typeof null);

**Arithmetic Operators**

+ Addition

- Subtraction

\* Multiplication

/ Division

\*\* Exponentiation (ES7)

% Modulus (Division Remainder)

++ Increment [ Post / Pre ]

-- Decrement [ Post / Pre ]

- + Unary Plus [Return Number If It’s Not Number]

- - Unary Negation [Return Number If It’s Not Number + Negates It]

console.log(+"100");

console.log(-"100");

console.log(Number("100"));

--------------------------------------------------------------------------------------------------------------------

**Type Coercion (Type Casting)**

- +

- -

- "" - 2

- false – true

let a = "100";

let b = 20;

let c = true;

console.log(+a + b + c);

**Number**

- Double Precision

- Syntactic Sugar "\_"

- e

- \*\*

- With Decimal

- Number + BigInt

- Number Min Value MIN\_SAFE\_INTEGER / MIN \_INTEGER

- Number Max Value MAX\_SAFE\_INTEGER / MAX \_INTEGER

--------------------------------------------------------------------------------------------------------------------

Number Methods

- Two Dots To Call A Methods

- toString()

- toFixed()

- parseInt()

- parseFloat()

- isInteger() [ES6]

- isNaN() [ES6]

console.log(Number.isInteger("100"));

console.log(Number.isNaN("Osama" / 20));

console.log(parseFloat("100.500 Osama"));

console.log(parseInt("100 Osama"));

console.log((100.1).toString());

console.log((100.554555).toFixed(2));

**Math Object**

- round()

- ceil()

- floor()

- min()

- max()

- pow()

- random()

- trunc() [Es6] بتكب كل شي بعد الفاصلة

console.log(Math.round(99.2));

console.log(Math.round(99.5));

console.log(Math.ceil(99.2));

console.log(Math.floor(99.9));

console.log(Math.min(10, 20, 100, -100, 90));

console.log(Math.max(10, 20, 100, -100, 90));

console.log(Math.pow(2, 4));

console.log(Math.random());

console.log(Math.trunc(99.5));

**String Methods**

- Access With Index

S.

- Access With charAt() بترجع محرف واحد

- length طول السلسة

- trim() بتشيل المسافات قبل وبعد السلسلة

- toUpperCase()

- toLowerCase()

- Chain Methods

- indexOf(Value [Mand], Start [Opt] 0) بحث عن شي داخل السلسة

- lastIndexOf(Value [Mand], Start [Opt] Length) بحث عن شي داخل السلسة

- slice(Start [Mand], End [Opt] Not Include End) قص من السلسة بتقبل ارقام موجبة وسالبة

- repeat(Times) [ES6]تكرار السلسة

- split(Separator [Opt], Limit [Opt]) قص من السلسة وبترجع العنصر مصوفة بتاخد عن فين قص وكم قصة

- substring(Start [Mand], End [Opt] Not Including End) قص من السلسة ووضع في سلسلة جديدة

--- Start > End Will Swap

--- Start < 0 It Start From 0

--- Use Length To Get Last Character

- substr(Start [Mand], Characters To Extract) قص من السلسة حسب عدد المحارف ووضع في سلسلة جديدة

--- Start >= Length = ""

--- Negative Start From End

- includes(Value [Mand], Start [Opt] Default 0) [ES6] بحث عن ضمن السلسلة

- startsWith(Value [Mand], Start [Opt] Default 0) [ES6]

- endsWith(Value [Mand], Length [Opt] Default Full Length) [ES6]

--------------------------------------------------------------------------------------------------------------------

**Comparison Operators**

- == Equal

- != Not Equal

- === Identical

- !== Not Identical

- > Larger Than

- >= Larger Than Or Equal

- < Smaller Than

- <= Smaller Than Or Equal

--------------------------------------------------------------------------------------------------------------------

**Logical Operators**

- ! Not

- && And

- || Or

--------------------------------------------------------------------------------------------------------------------

**Control Flow**

- if

- else if

- else

  if (Condition) {

     Block Of Code

  }

--------------------------------------------------------------------------------------------------------------------

**Logical Or || إذا ماشفت المتغير حط النص البديل**

-- Null + Undefined + Any Falsy Value

Nullish Coalescing Operator ??

-- Null + Undefined

console.log(Boolean(0));

console.log(`The Price Is ${price || 200}`);

console.log(`The Price Is ${price ?? 200}`);

--------------------------------------------------------------------------------------------------------------------

**Ternary If Syntax**

a < 10 ? console.log(10): a >= 10 && a <= 40? console.log("10 To 40"): a > 40 ? a > 40 : console.log("Unknown");

--------------------------------------------------------------------------------------------------------------------

**Switch Statement**

switch(expression) {

Case 1:

// Code Block

break;

Case 2:

// Code Block

break;

Default:

// Code Block

}

- Default Ordering

- Multiple Match

- ===

**Arrays**

- Create Arrays [Two Methods] new Array() + []

- Access Arrays Elements

- Nested Arrays

- Change Arrays Elements

- Check For Array Array.isArray(arr);

--------------------------------------------------------------------------------------------------------------------

**Array Methods**

- Length

Arrays Methods [Adding And Removing]

- unshift("", "") Add Element To The First

- push("", "") Add Element To The End

- shift() Remove First Element From Array

- pop() Remove Last Element From Array

Arrays Methods [Search]

- indexOf(Search Element, From Index [Opt])

- lastIndexOf(Search Element, From Index [Opt])

- includes(valueToFind, fromIndex [Opt]) [ES7]

Arrays Methods [Sort]

- sort(Function [Opt])

- reverse

Arrays Methods [Slicing]

- slice(Start [Opt], End [Opt] Not Including End)

--- slice() => All Array

--- If Start Is Undefined => 0

--- Negative Count From End

--- If End Is Undefined || > Indexes => Slice To The End Array.length

--- Return New Array

- splice(Start [Mand], DeleteCount [Opt] [0 No Remove], The Items To Add [Opt])

--- If Negative => Start From The End

Arrays Methods [Joining]

- concat(array, array) => Return A New Array

- join(Separator)

--------------------------------------------------------------------------------------------------------------------

**Loop**

- For

for (let i = 0; i < 10; i++) {

  console.log(i);

}

Loop Control

- Break

- Continue

- Label

Loop

- While

while (index < products.length) {

  console.log(products[index]);

  index += 1;

}

Loop

- Do / While

do {

  console.log(i);

  i++;

} while (false);

--------------------------------------------------------------------------------------------------------------------

**Function**

- What Is Function ?

- User-Defined vs Built In

- Syntax + Basic Usage

- Example From Real Life

- Parameter + Argument

- Practical Example

function sayHello(userName) {

  console.log(`Hi ${userName}`);

}

sayHello("Osama");

--------------------------------------------------------------------------------------------------------------------

**Function**

- Return

- Automatic Semicolon Insertion [No Line Terminator Allowed]

- Interrupting

function generate(start, end) {

  for (let i = start; i <= end; i++) {

    if (i === 15) {

      // return;

      // `Interrupting`;

      return `Interrupting`;

    }

    console.log(i);

  }

}

console.log(generate(10, 20));

--------------------------------------------------------------------------------------------------------------------

**- Default Function Parameters**

- Function Parameters Default [Undefined]

- Old Strategies [Condition + Logical Or]

- ES6 Method

function sayHello(username = "Unknown", age = "Unknown") {

  // if (age === undefined) {

  //   age = "Unknown";

  // }

  // age = age || "Unknown";

  return `Hello ${username} Your Age Is ${age}`;

}

console.log(sayHello());

--------------------------------------------------------------------------------------------------------------------

**Function**

- Rest Parameters

- Only One Allowed

- Must Be Last Element

function calc(...numbers) {

  // console.log(Array.isArray(numbers));

  console.log(numbers);

  let result = 0;

  for (let i = 0; i < numbers.length; i++) {

    result += numbers[i]; // result = result + numbers[i]

  }

  return `Final Result Is ${result}`;

}

console.log(calc(10, 20, 10, 30, 50, 30));

--------------------------------------------------------------------------------------------------------------------

**Function Advanced Practice**

- Parameters

- Default

- Rest

- Loop

- Condition

function showInfo(us = "Un", ag = "Un", rt = 0, show = "Yes", ...sk) {

  document.write(`<div>`);

  document.write(`<h2>Welcome, ${us}</h2>`);

  document.write(`<p>Age: ${ag}</p>`);

  document.write(`<p>Hour Rate: $${rt}</p>`);

  if (show === "Yes") {

    if (sk.length > 0) {

      document.write(`<p>Skills: ${sk.join(" | ")}</p>`);

    } else {

      document.write(`<p>Skills: No Skills</p>`);

    }

  } else {

    document.write(`<p>Skills Is Hidden</p>`);

  }

  document.write(`</div>`);

}

showInfo("Osama", 38, 20, "Yes", "Html", "CSS");

--------------------------------------------------------------------------------------------------------------------

**Anonymous Function**

- Calling Named Function vs Anonymous Function

- Argument To Other Function

- Task Without Name

- SetTimeout

let calculator = function (num1, num2) {

  return num1 + num2;

};

setTimeout(function elzero() {

  console.log("Good");

}, 2000);

--------------------------------------------------------------------------------------------------------------------

**Function Inside Function Return Function**

function sayMessage(fName, lName) {

  let message = `Hello`;

  // Nested Function

  function concatMsg() {

    function getFullName() {

      return `${fName} ${lName}`;

    }

    return `${message} ${getFullName()}`;

  }

  return concatMsg();

}

console.log(sayMessage("Osama", "Mohamed"));

--------------------------------------------------------------------------------------------------------------------

**Arrow Function**

-- Regular vs Arrow [Param + No Param]

-- Multiple Lines

let print = function () {

  return 10;

};

let print = () => 10;

let print = (\_) => 10;

let print = \_ => 10;

console.log(print());

let print = function (num) {

  return num;

};

let print = (num) => num;

console.log(print(100))