JavaScript Program

Software Development

1. Desktop Applications
2. Mobile Applications
3. Wed Applications

Script

The are two Script

* ECMA Script
* Java Script

JavaScript add <script> in used to embed a client-side Script(JavaScript).

Src => indicates the URL.

|  |  |
| --- | --- |
| *URL* | The URL of the external script file. |

**Variables**

Variables are Containers for Storing Data

JavaScript Variables can be declared in 4 ways:

1. Var
2. Let
3. Const

**Values**

1. Number
2. Boolean
3. Symbol

**Identifiers**

Usually starts with a letter, underscore(\_), or doler sign($).

**KeyWords**

  A set of reserved words that cannot be used as names of functions, labels, or variables as they are already a part of the syntax of JavaScript. Each of the keywords has its own meaning.

**Data Types**

There are six data types in JavaScript.

**Primitive data types**

1. String
2. Number
3. Boolean
4. Null
5. Undefined

|  |  |  |
| --- | --- | --- |
| **Value** | **Type** | **Comment** |
| "Hello" | string | "Hello" is always "Hello" |
| 3.14 | number | 3.14 is always 3.14 |
| true | boolean | true is always true |
| false | boolean | false is always false |
| null | null (object) | null is always null |
| undefined | undefined | undefined is always undefined |

**Reference Data Type**

* Object

**Operators:**

* **Arthimatic Operators (+, -, \*, /, %)**

|  |  |
| --- | --- |
| **Operator** | **Description** |
| + | Addition |
| - | Subtraction |
| \* | Multiplication |
| \*\* | Exponentiation ([ES2016](https://www.w3schools.com/js/js_2016.asp)) |
| / | Division |
| % | Modulus (Division Remainder) |
| ++ | Increment |
| -- | Decrement |

* **Relational operators (>, <, >=, <=, == , !=, ===, !==)**

**Or comparison Operators**

|  |  |
| --- | --- |
| **Operator** | **Description** |
| == | equal to |
| === | equal value and equal type |
| != | not equal |
| !== | not equal value or not equal type |
| > | greater than |
| < | less than |
| >= | greater than or equal to |
| <= | less than or equal to |
| ? | ternary operator |

* **Logical operators (! (Not) , || (or), &&(And) )**

|  |  |
| --- | --- |
| **Operator** | **Description** |
| && | logical and |
| || | logical or |
| ! | logical not |

* **Conditional operators or ternary**
* **Assignment operations**

|  |  |  |
| --- | --- | --- |
| **Operator** | **Example** | **Same As** |
| = | x = y | x = y |
| += | x += y | x = x + y |
| -= | x -= y | x = x - y |
| \*= | x \*= y | x = x \* y |
| /= | x /= y | x = x / y |
| %= | x %= y | x = x % y |
| \*\*= | x \*\*= y | x = x \*\* y |

* **Type operations**

|  |  |
| --- | --- |
| **Operator** | **Description** |
| typeof | Returns the type of a variable |
| instanceof | Returns true if an object is an instance of an object type |

* **Bitwise operators**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Operator** | **Description** | **Example** | **Same as** | **Result** | **Decimal** |
| & | AND | 5 & 1 | 0101 & 0001 | 0001 | 1 |
| | | OR | 5 | 1 | 0101 | 0001 | 0101 | 5 |
| ~ | NOT | ~ 5 | ~0101 | 1010 | 10 |
| ^ | XOR | 5 ^ 1 | 0101 ^ 0001 | 0100 | 4 |
| << | left shift | 5 << 1 | 0101 << 1 | 1010 | 10 |
| >> | right shift | 5 >> 1 | 0101 >> 1 | 0010 | 2 |
| >>> | unsigned right shift | 5 >>> 1 | 0101 >>> 1 | 0010 | 2 |

**Function:**

**Data types**

**Name =>string**

**Age=>number**

**Question =>Boolean**

**Not question =>undefined**

**null**

## **Types of JavaScript Operators**

There are different types of JavaScript operators:

* Arithmetic Operators
* Assignment Operators
* Comparison Operators
* String Operators
* Logical Operators
* Bitwise Operators
* Ternary Operators
* Type Operators

Strings property =>{length}

const name ="prasanth"

//length property

console.log(name.length)

output

8

String methods => {charAt}

"charAt" retrieves character at specific index in a string.

Top of Form

//strings

const name ="prasanth"

//String methods

console.log(name.charAt(0))

output

p

**Index of**

Top of Form

**indexOf**: Method to find position of substring in a string.

//strings

const name ="prasanth"

//String methods

console.log(name.indexOf("as"))

output

2

**LastIndexof**

Top of Form

JavaScript method: Returns index of last occurrence in string/array.

//strings

const name ="prasanth"

//String methods

console.log(name.lastIndexOf("a"))

output

4

**Slice**

JavaScript slice: Extracts a section of an array. Syntax: array.slice(start, end).

Top of Form

//strings

const name ="prasanth"

//String methods

console.log(name.slice(3,5))

output

sa

**toUpperCase**

JavaScript **toUpperCase** converts a string to uppercase letters. Example: **"hello".toUpperCase()** returns **"HELLO"**. It's case-insensitive.

//strings

const name ="prasanth"

//String methods

console.log(name.toUpperCase())

Output

PRASANTH

**toLowerCase**

JavaScript **toLowerCase** converts a string to lowercase letters. Example: **let str = "Hello"; console.log(str.toLowerCase()); // Output: "hello"**.

//strings

const name ="prasanth"

//String methods

console.log(name.toLowerCase())

output

prasanth

**includes => Boolean**

JavaScript **includes** checks if an array contains a certain element, returning **true** or **false**. Syntax: **array.includes(element)**.

//strings

const name ="prasanth"

//String methods

console.log(name.includes())

**output**

**fasle**

**Split**

JavaScript's **split()** method divides a string into an array of substrings based on a specified separator and returns it.

//strings

const name ="prasanth"

//String methods

console.log(name.split("s"))

**output**

*(2) ['pra', 'anth']*

//strings

const name ="prasanth"

//String methods

console.log(name.split(""))

**output**

*(8) ['p', 'r', 'a', 's', 'a', 'n', 't', 'h']*

**Number**

**isInteger =>(Boolean)**

**isInteger** is a JavaScript method checking if a value is an integer. Returns true if it's an integer; false otherwise.

//number

const number ="456"

const age =34;

//number methods

console.log(Number.isInteger(number))

console.log(Number.isInteger(age))

**output:**

**false**

**true**

**parseFloat**

JavaScript parseFloat: Convert string to floating-point number.

Top of Form

//number

const number ="456.09prasanth"

const age =34;

//number methods

console.log(Number.parseFloat(number))

console.log(Number.parseFloat(age))

**output:**

**456.09**

**34**

**parseInt  
 parseInt** converts strings to integers in JavaScript. It takes a string and returns an integer, parsing from left to right.

//number

const number ="456.09prasanth"

const age =34.9087;

//number methods

console.log(Number.parseInt(number))

console.log(Number.parseInt(age))

**output:**

**456**

**34**

**Use in for toFixed**   
 JavaScript method **toFixed** converts a number to a string, rounding it to a specified number of decimal places.

//number

const number ="456.09prasanth"

const age =34.0987656;

//number methods

console.log(Number.parseFloat(number).toFixed(2))

console.log(Number.parseFloat(age).toFixed(2));

**output:**

**456.09**

**34.10**

**Chaning**

JavaScript chaining means invoking multiple methods on an object in a single statement, passing the result of one method to the next.

//number

const age =34.0987656;

//number methods

console.log(parseFloat(age).toFixed(2).toString());

**output**

**34.10**

**isNaN**

**The are two type of isNaN**

1. **Number.isNaN**

**Number.isNaN** is a JavaScript method to determine if a value is "Not-a-Number" (**NaN**). It returns **true** if the value is **NaN**, else **false.**

1. **isNaN**

JavaScript **isNaN** function checks if a value is not a number. Returns true if not a number, false if number.

console.log(Number.isNaN("prasanth"))

console.log(isNaN("prasanth"))

**output**

**false**

**true**

**Math Properties**

**Math.PI =>**

JavaScript constant representing the rat+io of a circle's circumference to its diameter.

console.log(Math.PI)

**output**

3.141592653589793

**Math.trunc**

**Math.trunc: Removes decimal part, returns integer.**

console.log(Math.trunc(Math.PI))

**output**

**3**

**Math.Round**

JavaScript Math.round() rounds a number to the nearest integer. Halfway values round to the nearest even integer.

console.log(Math.round(4.4))

console.log(Math.round(4.6))

**output**

**4**

**5**

**Math.Floor**

**Math.floor** rounds down a number to the nearest integer, returning the largest integer less than or equal to a given number.

console.log(Math.floor(4.6))

**output**

**4**

**Math.Ceil**

Math.Ceil() rounds up a number to the nearest integer, returning the smallest integer greater than or equal to the given number.

console.log(Math.ceil(4.1))

**output**

**5**

**Math.pow**

JavaScript's **Math.pow** function raises a base to an exponent. Example: **Math.pow(2, 3)** returns 8.

console.log(Math.pow(4,3))

**output**

**64**

**Math.min**

JavaScript's **Math.min** returns the smallest of zero or more numbers. Example: **Math.min(10, 5) // Output: 5**.

console.log(Math.min(4,6,8,10,12))

**output**

**4**

**Math.Max**

JavaScript's **Math.max** returns the highest value of given numbers. Syntax: **Math.max(num1, num2, ...)**.

console.log(Math.max(4,6,8,10,12))

**output**

**12**

**Math.random**

JavaScript's **Math.random()** generates a random decimal between 0 (inclusive) and 1 (exclusive), crucial for creating randomness in applications.

console.log(Math.random()\*5)

**If-else**

The **JavaScript if-else statement** is used to execute the code whether condition is true or false. There are three forms of if statement in JavaScript.

1. If Statement
2. If else statement
3. if else if statement
4. switch statement

**if statement**

It evaluates the content only if expression is true. The signature of JavaScript if statement is given below.



**ex:**

let name = "prasanth js class";

 if (name) {

   console.log(`enjay ${name}`);

 }

Output

enjay prasanth js class

JavaScript if-statements check conditions and execute code based on true or false. Syntax: if (condition) {code block}. Optional: else, else if.

**If else statement**

JavaScript if-else statements execute code if a condition is true; otherwise, they execute alternative code. Syntax: if (condition) {} else {}.

Top of Form

 let name ;

 if (name) {

   console.log(`enjay ${name}`);

 }else{

    console.log("sorry sir")

 }

Output

Sorry sir



**if else if statement**

**JavaScript if-else if statements execute different code blocks based on multiple conditions. Syntax: if (condition1) {code1} else if (condition2) {code2}.**

let Mark = 68;

 if (Mark >= 90) {

   console.log("A");

 }

 else if(Mark >= 80){

    console.log("B")

 }

 else if(Mark>=70){

    console.log("C")

 }

else if (Mark>=60){

    console.log("D")

}

 else{

    console.log("F")

 }

Output

D

Switch Statement

JavaScript switch statement evaluates an expression and executes code blocks based on matching cases. Syntax: switch (expression) {case: statements}.

let grade = "D";

let result;

switch (grade){

    case 'A':

        result = "A grade";

        break;

    case 'B':

        result ="B grade";

        break;

    case 'C':

        result ="C grade";

        break;

    case 'D':

        result ="D grade"

        break;

    default:

        result = "No grade"

}

console.log(result)

output

D grade

Ternary => ?

JavaScript ternary operator (? :) is a concise conditional statement. Syntax: condition ? expression1 : expression2. Used for simple if-else logic.

let grade ="B";

let result = grade  === 'A' ? "A grade max mark 90 to 100": grade === 'B'?"B grade max mark 70 to 90":grade ==='C'?"C grade max mark 50 to 70":"D grade min mark Fail";

console.log(result);

output

B grade max mark 70 to 90

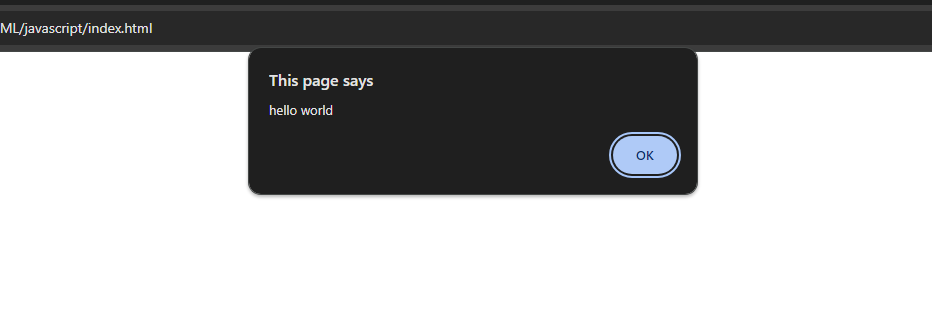
**User input**

**alert**

JavaScript alert displays a message in a dialog box. Syntax: `alert("message")`. Useful for notifications and simple interactions.

let my = alert("hello world")

console.log(my)

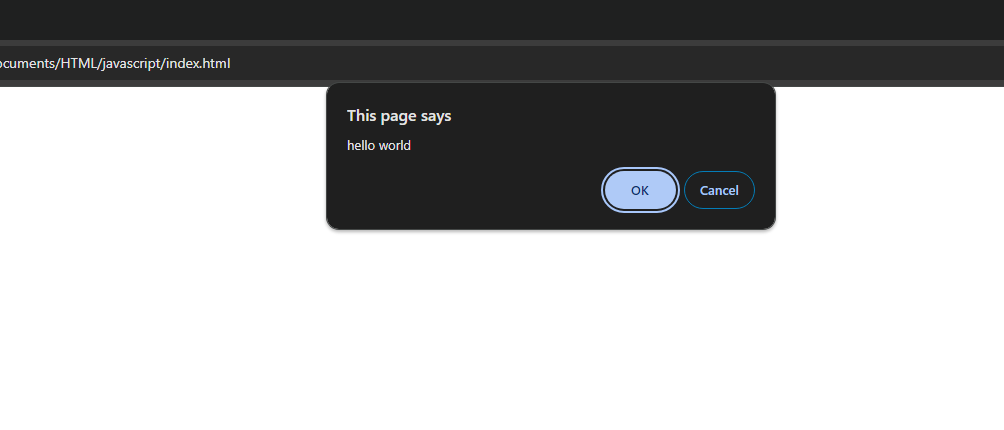


**Confirm**

**JavaScript `confirm` displays a dialog box with OK/Cancel buttons. Returns true if OK is clicked, false for Cancel.**

let my = confirm("hello world");

console.log(my)



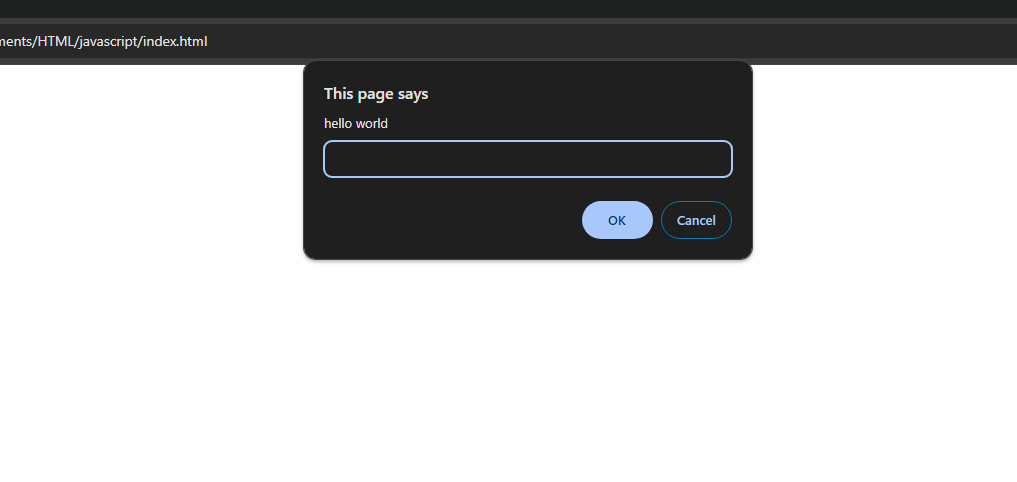
Prompt

JavaScript prompt() displays a dialog box for user input. Syntax: prompt("message", "default"). Returns user input or null.

Ex 1:

let my = prompt("hello world");

console.log(my)



Output:

Not user name ok button=> no ans

Ex 2:

let my = prompt("Enter your name:");

if(my){

console.log(my ?? "you didn't Enter your name");

}else{

    console.log("you didn't Enter your name");

}

Output:

Not user name ok button=> Null you didn’t enter your name

cancel button =>you didn’t enter your name

Ex 3:

let my = prompt("Enter your name:");

if (my) {

  console.log(my ?? "you didn't Enter your name");

} else {

  console.log("you didn't Enter your name");

}

console.log(my.length)

Output:

Not user name ok button=> prasanth

Name after space : 32

cancel button =>you didn’t enter your name

Ex 4:

let my = prompt("Enter your name:");

if (my) {

  console.log(my ?? "you didn't Enter your name");

} else {

  console.log("you didn't Enter your name");

}

console.log(my.trim().length)

Output:

Not user name ok button=> prasanth

Name not space : 8

cancel button =>you didn’t enter your name

Loops

While loop:

Loop executes while condition is true; may cause infinite iteration.

Ex :1

const name = ["HTML","CSS","javascript"]

let i = 0;

while(name[i]){

    console.log(name[i]);

    i=i+1;

}

Output

HTML  
CSS

javascript

EX :2

let i = 1;

while(i <= 10){

    console.log(i);

    i=i+1;

}

Output

1

2

3

4

5

6

7

8

9

10

Do while loop

JavaScript `do-while` loop executes code block once, then repeats if condition is true, ensuring at least one execution.

Ex : 1 => (string type)

let name = ["HTML","CSS","javascript"]

let i=2;

do{

    console.log(name[i]);

    break;

}while(name[i])

Output

Javascript

Ex : 2 => (number)

let i=50;

do{

    console.log(i);

    i=i+10;

}while(i<50)

Output:

50

For loop