DATA TYPES IN JS

1.what will be the output?

let x = 5;

let y = x;

x = 10;

console.log(x);

console.log(y);

output:10

5

Explanation: Initially, y is assigned the value of x, which is 5. When x is later changed to 10, it doesn't affect y because y holds a copy of the original value. Therefore, console.log(x) output is 10, while console.log(y) output is 5.

2.what will be the output?

let obj1 = {name:"alice"};

let obj2 = obj1;

obj1.name = "bob";

console.log(obj1.name)

console.log(obj2.name)

output: bob

Bob

Explanation: obj2 is assigned a reference to obj1, so both variables point to the same object in memory. When obj1.name is changed to "bob", it updates the shared object, making obj2.name also reflect this change. Thus, both console.log statements output "bob".

3.what will be the output?

```
let a = "hello";
let b =42;
let c =true;
let d = {key: "value"};
let e = null;
let f = undefined;

console.log(typeof a);
console.log(typeof b);
console.log(typeof c);
console.log(typeof d);
console.log(typeof e);
console.log(typeof f);
```

Output:

string

number

Boolean

object

object

Undefined

Explanation: typeof keyword is used to determine what type of data type is assigned to a variable, string is represented in double quotation. Boolean value may be true or false,typeof null is object type of key value pair is represented as object,typeof undefined is undefined.

4.what will be the output?

let numbers = [10,20,30,40,50];

console.log(numbers[2]);

console.log(numbers[0]);

console.log(numbers[numbers.length - 1]);

Output:

30

10

50

Explanation: In array starting index value is 0, so when we are accessing the value of the reference then the particular value is going to be printed as output. The length of array is determined by number of values presented in it if there are six elements then length is going to be 6.to access the last element we need to subtract length of array with 1.

5.var fruits = ["apple","banana","mango"];

fruits[1] = "orange";

console.log(fruits);

output: ["apple", "orange", "mango"].

Explanation: when initializing the fruits the index of 1 is assigned with banana. In the next step the reference of 1 is reassigned with orange so that output is going to change.

6.what will be the output?

let matrix = [

[1,2,3],

[4,5,6],

[7,8,9]

];

console.log(matrix[1][2]);

console.log(matrix[2][0]);

Output:

6

7

Explanation: In the above matrix the values are stored in index based so when we are trying to access the matrix [1][2] then at second index 3 value is going to print as output which is 6.in the second case the matrix [2][0] at third index first value is going to be printed.

7.what will be the output?

let person = {

**   name:"john",**

age:25,

**   city:"Newyork"**

};

console.log(person.name);

console.log(person. age);

Output:

john

25

Explanation: The above data type is object because the data is stored in key value pair.each value is called property separated by. When we are trying to access the key then the associated value is going to print as output.

8.what will be the output?

let car = {

make: "Toyota",

model: "Carolla",

year : 2021

};

console.log(car["make"]);

console.log(car["model"]);

Output:

Toyota

Carolla

Explanation: In object data type we can access the value by .(dot) form and in brackets mentioned in double quotes so that we are trying to access the key then the associated value is going to print as output.

9.what will be the output?

let book = {

title: "the great gatsby",

author: "f.scott fitzgerald"

};

book.author ="anonymous"

console.log(book.author);

Output:

anonymous

Explanation: we are reassigning author value with Anonymous so that ,when we perform console operation then Anonymous is going to print.

10.what will be the output?

let student = {

name:"alice",

grade: "a"

};

student.age = 20;

console.log(student);

output:

{ name: 'alice', grade: 'a', age: 20 }

Explanation: we are assigning 20 to the age variable then by .(dot) we are assigning to add to which object we have student object we are assigning to it.