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:

- `Let x=5;` initializes x with the value 5.
- `Let y=x;`assigns the current value of `x` (which is 5) to `y`. y also holds the value 5.
- `x=10;` changes the value of x to 10. However, this does not affect y, which remains 5
- `console.log(x);` outputs the current value of x, which is 10.
- `console.log(y);` outputs the current value of y, which is still 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:

- `let obj1={ name: "alice" };` creates an object `obj1` with a property `name` set to `"alice"`.

- `let obj2 = obj1;` assigns the reference of `obj1` to `obj2`. This means both `obj1` and `obj2` point to the same object in memory.
- `obj1.name = "bob";` changes the `name` property of the object that both `obj1` and `obj2` reference. Now, the `name` property is `"bob"`.
- `console.log(obj1.name);` outputs the current value of `obj1.name`, which is `"bob"`.
- `console.log(obj2.name);` also outputs the current value of `obj2.name`, which is still `"bob"` since both variables point to the same object.

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:

- `console.log(typeof a);` outputs `"string"` because a is a string.
- `console.log(typeof b);` outputs `"number"` because b is a number.
- `console.log(typeof c);` outputs `"boolean"` because c is a boolean.
- `console.log(typeof d);` outputs `"object"` because d is an object.
- `console.log(typeof e);` outputs `"object"` because `null` is considered an object type.
- `console.log(typeof f);` outputs `"undefined"` because `f` is explicitly set to `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:

- `let numbers = [10, 20, 30, 40, 50];` creates an array named `numbers` containing five elements: `10`, `20`, `30`, `40`, and `50`.
- `console.log(numbers[2]);` accesses the element at index 2. In JavaScript, array indices start at `0`, so:
    - Index `0` corresponds to `10`

    - Index 1 corresponds to `20`
    - Index 2 corresponds to `30`
    - Therefore, this outputs `30`.
- `console.log(numbers[0]);` accesses the element at index `0`, which is `10`. This outputs `10`.
- `console.log(numbers[numbers.length - 1]);` accesses the last element of the array. `numbers.length` returns the total number of elements in the array, which is 5. So, `numbers.length - 1` equals 4, which corresponds to the last index:
    - Index 4 corresponds to `50`
    - Therefore, this outputs `50`.

5.what will be the output?

```
let fruits=["apple","banana","mango"];

fruits[1]="orange";

console.log(fruits);
```

Output: ['apple', 'orange', 'mango']

Explanation:

- The array `fruits` is initialized with three elements: `"apple"`, `"banana"`, and `"mango"`.
- The statement `fruits[1]="orange";` changes the value at index 1 of the array. In JavaScript, the first element is at index 0, the second at index 1, and so on. Therefore, `fruits[1]` refers to `"banana"`.
- After the assignment, the second element of the array is updated to `"orange"`, resulting in the final array being `["apple","orange","mango"]`.
- The `console.log(fruits);` statement then prints the updated array to the console.

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:

- The variable `matrix` is defined as a 2D array (an array of arrays). It has three rows and three columns:

```
[
    [1, 2, 3],  // Row 0
    [4, 5, 6],  // Row 1
    [7, 8, 9]   // Row 2
]
```

- `matrix[1][2]`: This accesses the element in the second row (index 1) and the third column (index 2). In the matrix, this corresponds to the value 6.
- `matrix[2][0]`: This accesses the element in the third row (index 2) and the first column (index 0). In the matrix, this corresponds to the value 7.
- The two `console.log` statements print the values 6 and 7 .

7.what will be the output?

```
let person={name:"john",age:25,city:"new york"};

console.log(person.name);

console.log(person.age);
```

Output: john

25

Explanation:

- The variable `person` is defined as an object with three properties:
  `name`: `"john"`
  `age`: `25`
  `city`: `"new york”`
- `person.name`: This accesses the `name` property of the `person` object, which has the value `"john"`.
- `person.age`: This accesses the `age` property of the `person` object, which has the value `25`.
- The `console.log` statements print the values of `person.name` and `person.age` to the console, resulting in `john` and `25`.

8.what will be the output?

```
let car={make:"toyoto",model:"corolla",year:2021};

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

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

Output: toyoto

Corolla

Explanation:

- The variable `car` is defined as an object with three properties:

  `make`: `"toyoto"`

  `model`: `"corolla"`

  `year`: `2021`

- `car["make"]`: This retrieves the `make` property of the `car` object, returning the value `"toyoto"`.
- `car["model"]`: This retrieves the `model` property of the `car` object, returning the value `"corolla"`.
- The `console.log` statements print the values of `car["make"]` and `car["model"]` to the console, resulting in `toyoto` and `corolla`.

9.what will be the output?

```
let book={tittle:"the great gatsby",author:"f.scott fitzgerald"};

book.author="anonymous"

console.log(book.author);
```

Output: anonymous

Explanation:

- Initially, the `author` of the `book` object is set to `"f. scott fitzgerald"`.
- Then, we overwrite this property by assigning `"anonymous"` to `book.author`.
- When you log `book.author`, it now returns `"anonymous"` because the property has been updated.

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:

- The `student` object initially contains two properties: `name` and `grade`.
- Then, the new property `age` is added with a value of `20`.
- When you log the `student` object, it will show all three properties: `name`, `grade`, and `age`.