Data types in JS

1. **What will be the output?**

```
let x=5;
let y=x;
x=10;
console.log(x,y);
```

O/P:10 5

Explanation: At first x value is 5 and in the second line the x value is assigned to y and now the y value is 5 and then in the third line the x is reassigned with 10 and hence the output for console.log(x,y) is 10 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: At first obj1 is assigned with name alice and then obj2 is assigned to obj1 and at the third line obj1 is reassigned to name Bob since objects are mutable hence the obj1 and obj2 both changes their name to Bob.

3. **What will be the output?**

```
let a=" hello"
let b=47;
let c=true;
let d= {key: "value"};
let e=null;
let f=undefined;
console.log (type of a);
console.log (type of b);
console.log (type of c);
console.log (type of d);
console.log (type of e);
console.log (type of f);
```

Output:

string

number

Boolean

object

object
undefined
Explanation: The variable a is a string, so its type is string. The variable b is a number, so its type is number. The variable c is a Boolean, so its type is Boolean. The variable d is an object, so its type is object. Note that null is also considered an object in JavaScript. The variable e is null, which is also classified as object, and f is undefined, so its type 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: At first, we defined array with some values 10,20,30,40,50 with a variable numbers and then in next line we tried to display numbers[2] so it takes the index of third element and prints the value 30 and in next line we tried to display numbers[0] so it takes the index of first element and prints the value 10 and in the last line we tried to display numbers[numbers.length-1] so first numbers.length is 5 and numbers.length-1 means 5-1 so it is 4 and then numbers[4] it will consider the 5th element value that is 50.

5. **What will be the output?**
   let fruits= [“apple”,” banana”,” mango”];
   fruits [2] =” orange”;
   console.log(fruits);
   Output: [‘apple’, ‘banana’, ‘orange’]
   Explanation: At first, we defined an array with some values apple banana and mango to a variable fruits. In the next step we changed the value at index 2 with orange and finally we are printing so the output is apple, banana and orange.

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: Here we used nested arrays concept means that arrays inside another arrays first array takes 3 inputs [1,2,3] as index 0, [4,5,6] as index 1, [7,8,9] as index

2. Here in the next line we tried to print matrix[1][2] so the first index is [1] so it goes to [4,5,6] and then the next index is [2] so inside [4,5,6] it will choose 6 so the output is 6. Here in the next line we tried to print matrix[2][0] so the first index is [2] so it goes to [7,8,9] and then the next index is [0] so inside [7,8,9] it will choose 7 so the output is 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: Here first we consider one object with some keys and values { name: “john”, age:25, City: “New York” }; in the next line we tried to print person.name so the output is john in the last line we tried to print person.age so the output is 25.

8. **What will be the output?**

```
let car= { make:” Toyota”, model: “corolla”, year: 2021 };
console.log(car[“make”]);
console.log (car[“model”]);
```

Output: Toyota, corolla

Explanation: Here first we take an object with keys and values in the next line we used to display car[“make”] so it will print Toyota and in the next line car[“model”] it will print corolla.

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: Here we defined one object with some properties and in the next line we tried to update the author key with anonymous value and in the third line if we want to print the book.author then will print the updated value Anonymous.

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: Here we defined one object with some properties and in the next line we added one more key age with some value of 20 so if we try to print object then it

will consider both old and new properties so the output is : name: 'Alice', grade: 'A', age: 20.