Java script day 2 task

1)What will be the output?

```
let x=5;
let y=x;
x=10;
console.log(x);
console.log(y);
O/P:10
    5
```

The output shows `10` for `x` (after the update) and `5` for `y` (which retains its original value). This demonstrates that `y` was assigned a copy of the value, making it unaffected by subsequent changes to `x`.

2)What will be the output?

```
let obj1={name:"Alice"};
let obj2=obj1;
obj1.name="Bob";
console.log(obj1.name);
console.log(obj2.name);
O/P: Bob
     Bob
```

`obj2` is assigned to `obj1`, meaning both variables reference the same object in memory. When you change `obj1.name` to "Bob", it also updates `obj2.name` because they point to the same object.

3)

```
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);
O/P:
String
Number
Boolean
Object
Object
undefined
```

The only unexpected result is `typeof e`, which returns `"object"` instead of `"null"`.

4)

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

console.log(numbers[2]);
console.log(numbers[0]);
console.log(numbers[numbers.length-1]);
O/P:
30
10
50
```

It returns index value in the given input.

5)

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

fruits[1]="orange";
console.log(fruits);
O/P: [ 'Apple', 'orange', 'mango' ]
```

It replaces orange in the place of banana because we have given index value 1.

6)

```
let matrix=[[1,2,3],[4,5,6],[7,8,9]];
console.log(matrix[1][2]);
console.log(matrix[2][0]);
O/P:
6
7
```

`matrix[1][2]` accesses the element in the second row and third column, which is `6`.
`matrix[2][0]` accesses the element in the third row and first column, which is `7`.

7)

```
let person = {name:"john",age:25,city:"New York"};
console.log(person.name);
console.log(person.age);
O/P:
John
25
```

`console.log(person.name);` logs the value of the `name` property, which is `"john"`.
`console.log(person.age);` logs the value of the `age` property, which is `25`.

8)

```
let car={
    make:"Toyota",
    model:"Corolla",
    year:2021
};
console.log(car["make"]);
console.log(car["model"]);
O/P:
Toyota
```

```
Corolla
```

`console.log(car["make"]);` retrieves the value of the `make` property, which is `"Toyota"`.
`console.log(car["model"]);` retrieves the value of the `model` property, which is `"Corolla"`.

9)

```
let book={
    title:"The Great Gatsby",
    author:"F.Scott Fitzgerald"
};
book.author="Anonymous";
console.log(book.author);
O/P: Anonymous
```

Initially, the `author` is set to `"F.Scott Fitzgerald"`. The line `book.author = "Anonymous";` changes the `author` property to `"Anonymous".console.log(book.author);` then logs the updated value.

10)

```
let student={
    name:"Alice",
    grade:"A"
};
student.age=20;
console.log(student);
O/P: { name: 'Alice', grade: 'A', age: 20 }
```

The `student` object is initially defined with properties `name` and `grade`.
The line `student.age = 20;` adds a new property `age` with a value of `20`.
`console.log(student);` outputs the entire object, showing all properties.