Task

1)What will be the output?

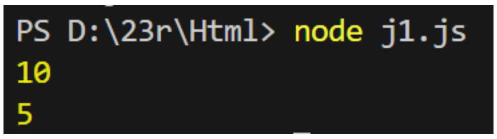
Let x=5;

Let y=x;

x=10;

console.log(x);

console.log(y);

Ans: 

```
PS D:\23r\Html> node j1.js
10
5
```

Explanation:

- let y = x; creates a new variable y and assigns it the value that x currently holds .
- Later, when you change the value of x to 10, it does not affect y because y was already assigned the value 5 independently.

2)What will be the output?

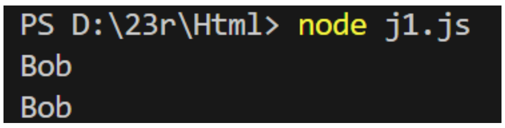
Let obj 1= {name: “Alice”};

Let obj 2=obj 1;

Obj 1.name=” Bob”;

console.log (Obj 1.name);

console.log (Obj 2.name);

Ans: 

```
PS D:\23r\Html> node j1.js
Bob
Bob
```

Explanation:

- let obj2 = obj1; does not create a new object. Instead, it creates a reference to the same object that obj1 points to.
- When you update obj1.name to "Bob", the change is reflected in both obj1 and obj2 since they both reference the same object.

3) Let a=” hello”

Let b=47;

Let c=true;

Let d= {key: “value”};

Let e=null;

Let f=undefined;

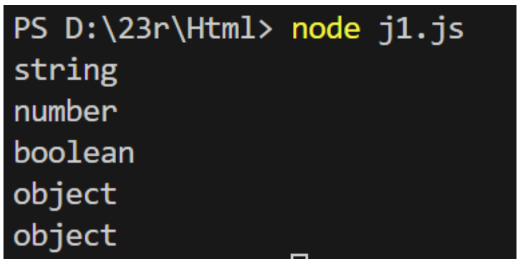
console.log (type of a);

console.log (type of b);

console.log (type of c);

console.log (type of d);

console.log (type of e);

Ans: 

```
PS D:\23r\Html> node j1.js
string
number
boolean
object
object
```

Explanation:

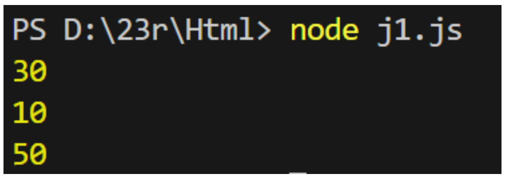
- typeof a returns "string" because a is a string.
- typeof b returns "number" because b is a number.
- typeof c returns "boolean" because c is a boolean.
- typeof d returns "object" because d is an object.
- typeof e returns "object" due to a known peculiarity in JavaScript where null is considered an object, even though it represents the absence of any object.

4)Let numbers= [10,20,30,40,50];

console.log (numbers [2]);

console.log (numbers [0]);

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

Ans: 

```
PS D:\23r\Html> node j1.js
30
10
50
```

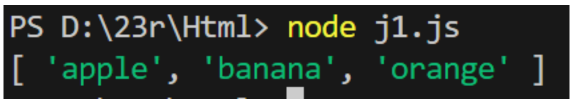
Explanation:

- numbers[2] accesses the third element in the array, which is 30.
- numbers[0] accesses the first element in the array, which is 10.
- numbers[numbers.length - 1] accesses the last element in the array. Since numbers.length is 5, this expression evaluates to numbers[4], which is 50.

5)Let fruits= [“apple”,” banana”,” mango”];

Fruits [2] =” orange”;

Console.log(fruits);

Ans: 

```
PS D:\23r\Html> node j1.js
[ 'apple', 'banana', 'orange' ]
```

Explanation:

- fruits[2] = "orange"; changes the third element of the array from "mango" to "orange".

6) Let matrix= [

[1,2,3],

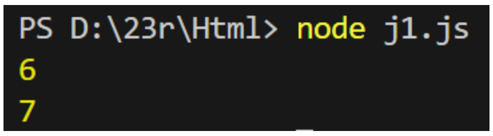
[4,5,6],

[7,8,9]

];

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

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

Ans: 

```
PS D:\23r\Html> node j1.js
6
7
```

Explanation:

- matrix[1][2] refers to the element in the second row and third column of the matrix, which is 6.
- matrix[2][0] refers to the element in the third row and first column of the matrix, which is 7.

7)let person= {

Name:” john”,

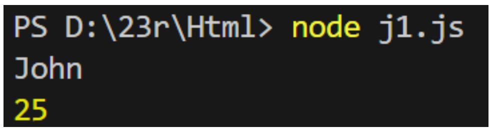
Age:25,

City: “New York”

};

Console.log(person.name);

Console.log (person. age);

Ans: 

```
PS D:\23r\Html> node j1.js
John
25
```

Explanation:

- person.name accesses the name property of the object, which is "John".
- person.age accesses the age property of the object, which is 25.

8)let car= {

make:” Toyota”,

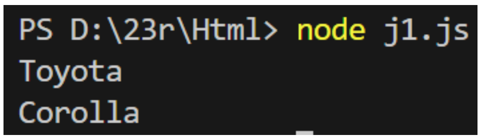
model: “corolla”,

year: 2021

};

Console.log(car[“make”]);

Console.log (car[“model”]);

Ans: 

```
PS D:\23r\Html> node j1.js
Toyota
Corolla
```

Explanation:

- car["make"] accesses the make property of the object, which is "Toyota".
- car["model"] accesses the model property of the object, which is "Corolla".

9)let book= {

t

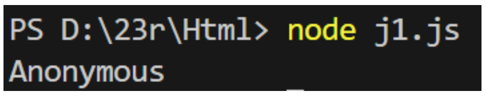
itle:” The Great Gatsby”,

author: “F. Scott Fitzgerald”

};

Book. Author=” Anonymous”;

Console.log (book. Author);

Ans: 

```
PS D:\23r\Html> node j1.js
Anonymous
```

Explanation:

- book.author = "Anonymous"; changes the author property of the object from "F. Scott Fitzgerald" to "Anonymous".

10)let student= {

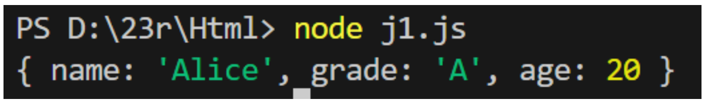
name:” Alice”,

grade: “A”

};

student. age=20;

Console.log (student);

Ans: 

```
PS D:\23r\Html> node j1.js
{ name: 'Alice', grade: 'A', age: 20 }
```

Explanation:

- student.age = 20; adds a new property called age to the student object.

- When you log student, it will output the entire object, including the newly added age property.