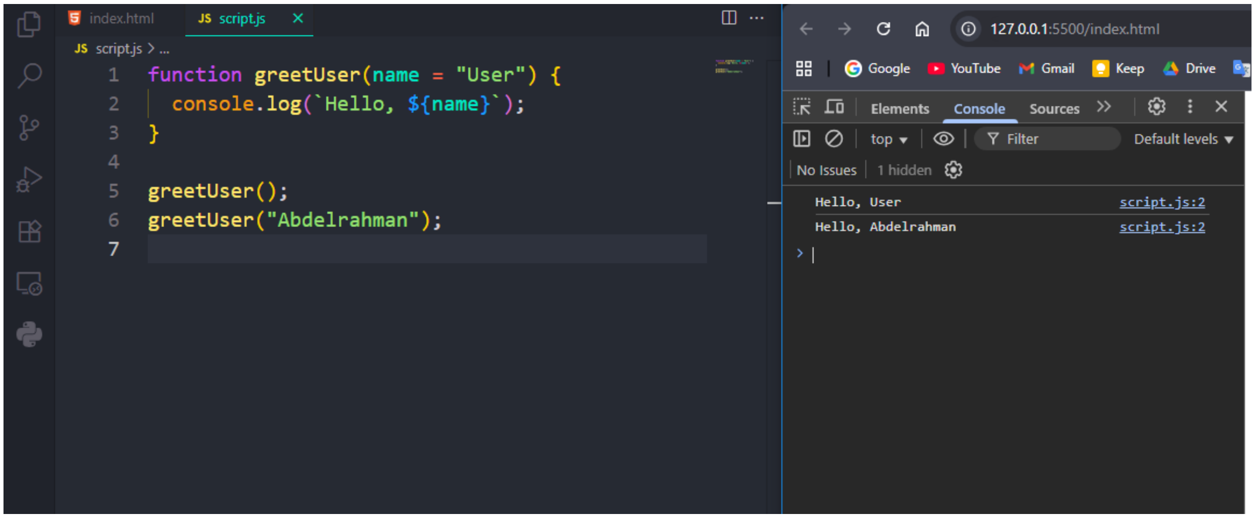
JavaScript ES6 – Day.7 Assignments

1- Write a function that greets a user, using a default parameter for the name.

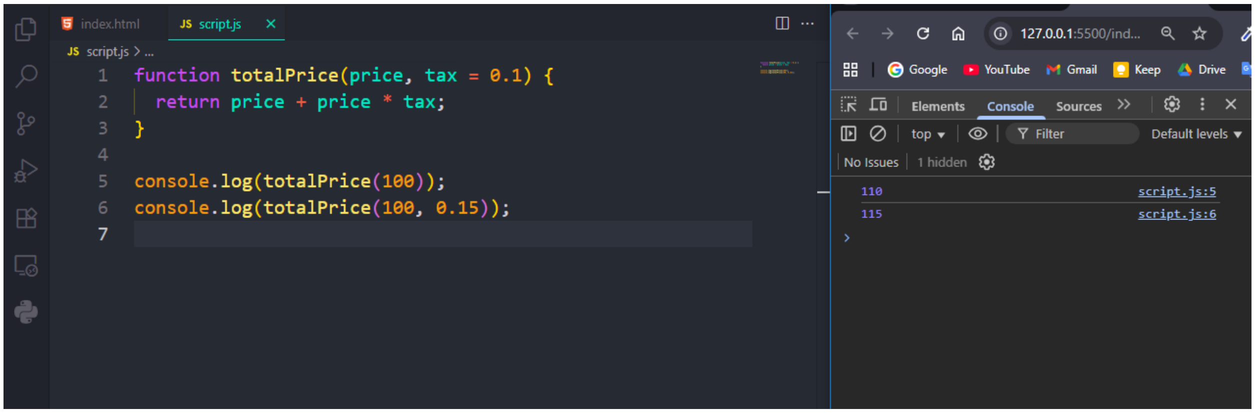


```javascript
function greetUser(name = "User") {
  console.log(`Hello, ${name}`);
}

greetUser();
greetUser("Abdelrahman");
```

```
Hello, User                  script.js:2
Hello, Abdelrahman           script.js:2
```

2- Write a function that calculates the total price with a default tax rate parameter.

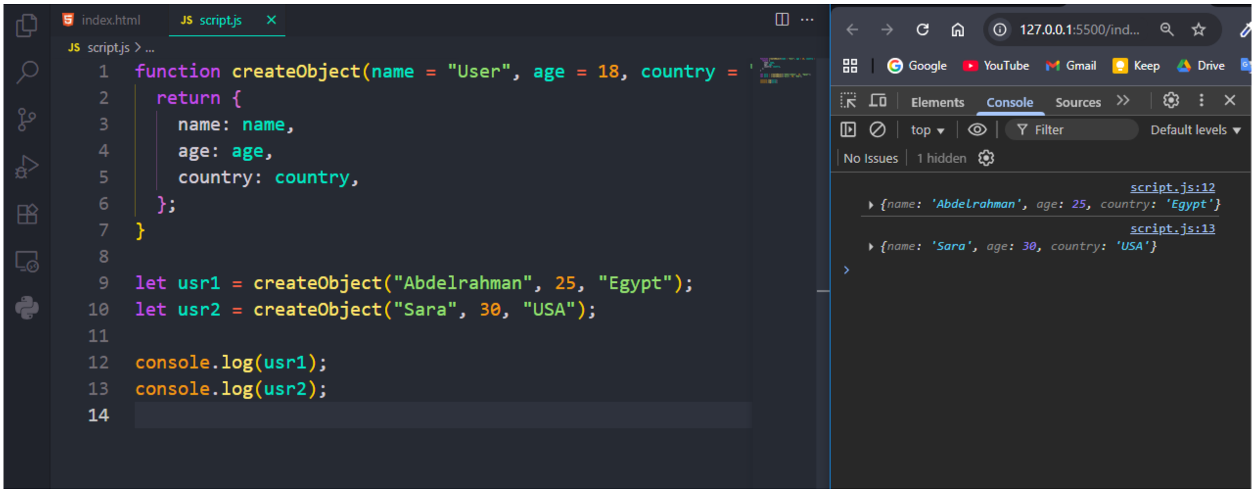


```javascript
function totalPrice(price, tax = 0.1) {
  return price + price * tax;
}

console.log(totalPrice(100));
console.log(totalPrice(100, 0.15));
```

```
110                          script.js:5
115                          script.js:6
```

3- Write a function that creates a user object, using a default role parameter.



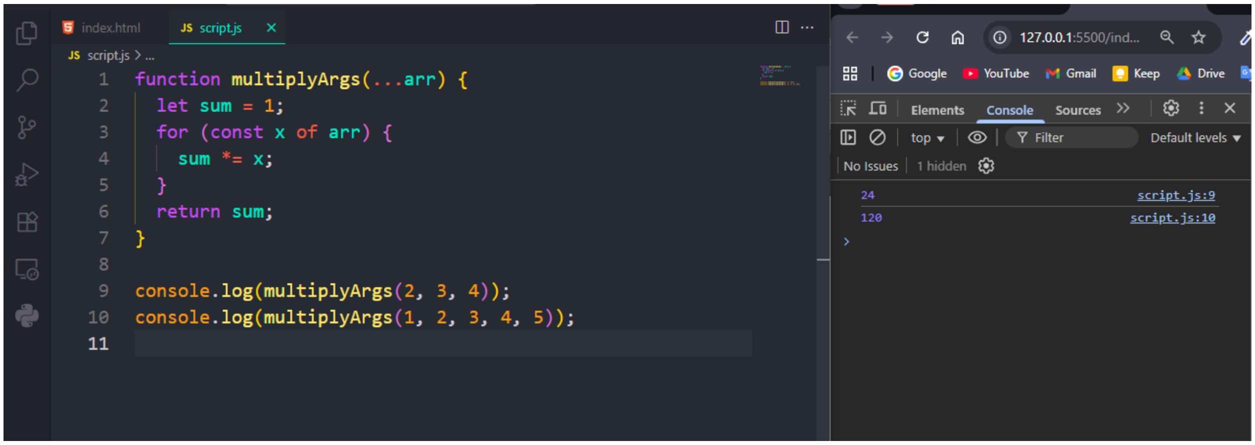
```javascript
function createObject(name = "User", age = 18, country = "
  return {
    name: name,
    age: age,
    country: country,
  };
}

let usr1 = createObject("Abdelrahman", 25, "Egypt");
let usr2 = createObject("Sara", 30, "USA");

console.log(usr1);
console.log(usr2);
```

```
                                              script.js:12
▸ {name: 'Abdelrahman', age: 25, country: 'Egypt'}
                                              script.js:13
▸ {name: 'Sara', age: 30, country: 'USA'}
>
```

4- Write a function that multiplies any number of arguments using the rest operator.

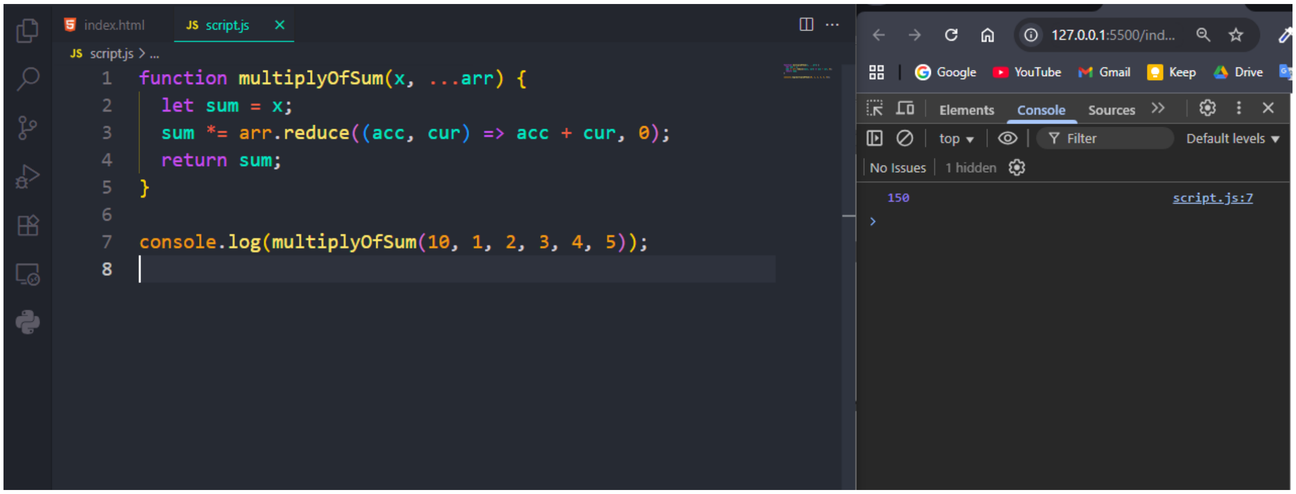


```javascript
function multiplyArgs(...arr) {
  let sum = 1;
  for (const x of arr) {
    sum *= x;
  }
  return sum;
}

console.log(multiplyArgs(2, 3, 4));
console.log(multiplyArgs(1, 2, 3, 4, 5));
```

```
24                                            script.js:9
120                                           script.js:10
>
```

5- Write a function that multiplies the first argument by the sum of all the rest using the rest operator.

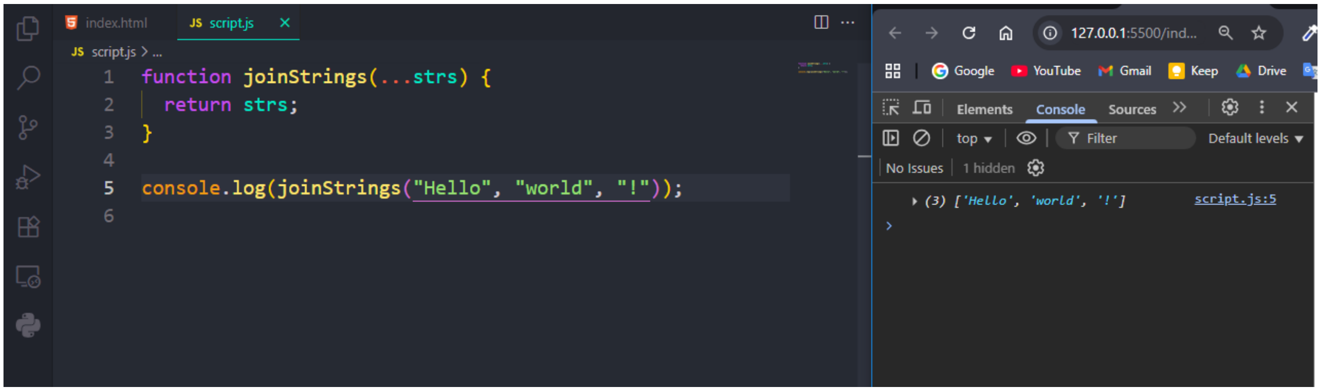


```javascript
function multiplyOfSum(x, ...arr) {
  let sum = x;
  sum *= arr.reduce((acc, cur) => acc + cur, 0);
  return sum;
}

console.log(multiplyOfSum(10, 1, 2, 3, 4, 5));
```

```
150                                          script.js:7
```

6- Write a function that takes a variable number of strings and returns them as a single array using the rest operator.

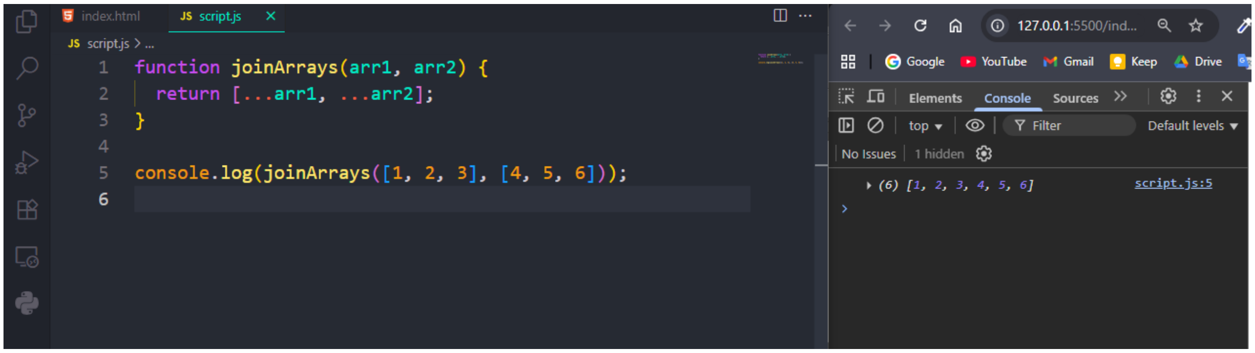


```javascript
function joinStrings(...strs) {
  return strs;
}

console.log(joinStrings("Hello", "world", "!"));
```

```
▸ (3) ['Hello', 'world', '!']                script.js:5
```

7- Create a new array by combining two arrays using the spread operator.

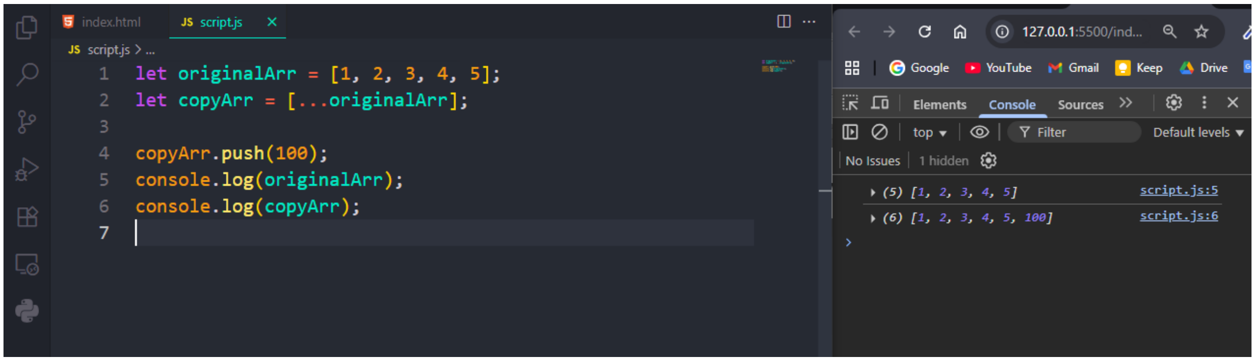


```javascript
function joinArrays(arr1, arr2) {
  return [...arr1, ...arr2];
}

console.log(joinArrays([1, 2, 3], [4, 5, 6]));
```

```
▸ (6) [1, 2, 3, 4, 5, 6]                script.js:5
```

8- Copy an array using the spread operator.

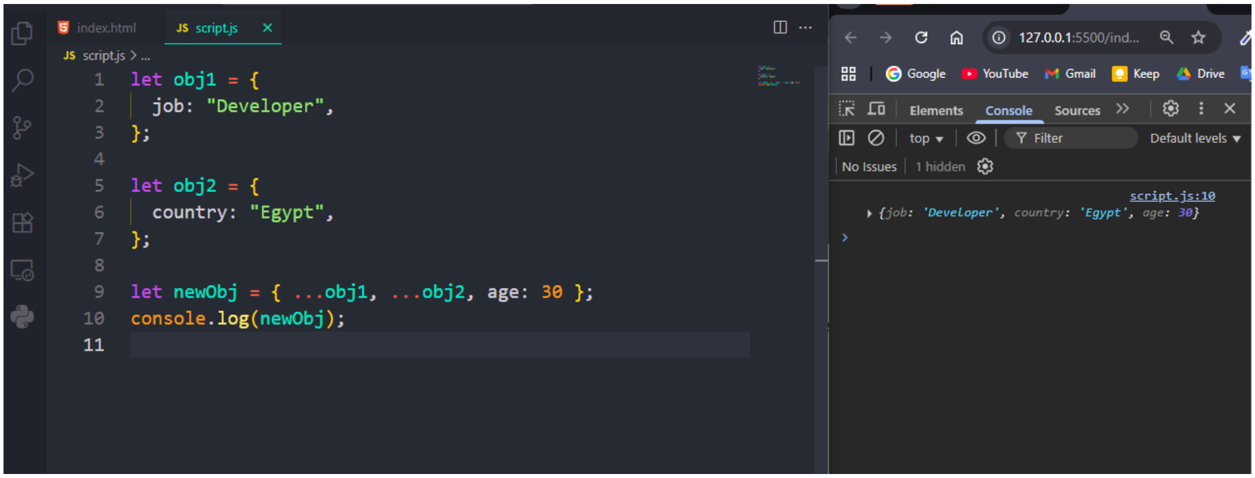


```javascript
let originalArr = [1, 2, 3, 4, 5];
let copyArr = [...originalArr];

copyArr.push(100);
console.log(originalArr);
console.log(copyArr);
```

```
▸ (5) [1, 2, 3, 4, 5]                   script.js:5
▸ (6) [1, 2, 3, 4, 5, 100]              script.js:6
```

9- Merge two objects into one using the spread operator.



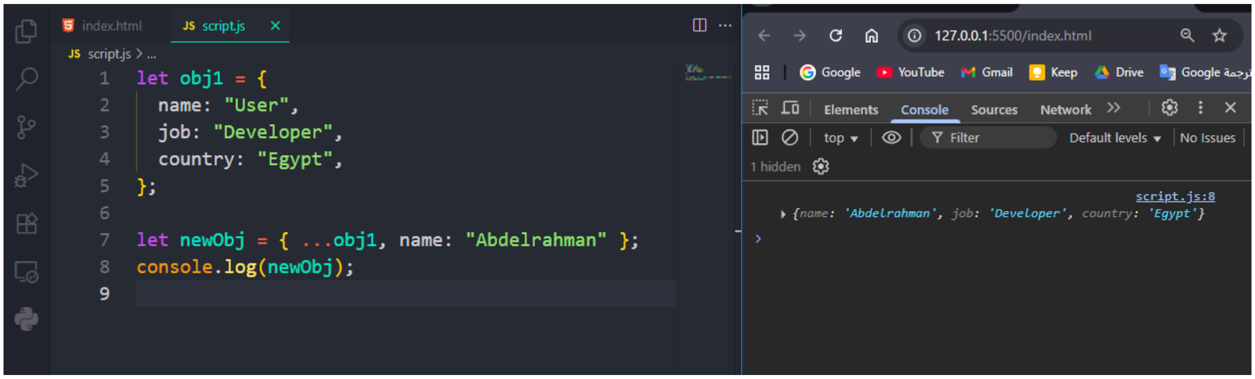
```javascript
let obj1 = {
  job: "Developer",
};

let obj2 = {
  country: "Egypt",
};

let newObj = { ...obj1, ...obj2, age: 30 };
console.log(newObj);
```

```
                                        script.js:10
▸ {job: 'Developer', country: 'Egypt', age: 30}
```

10- Update a property in an object using the spread operator to create a new object.

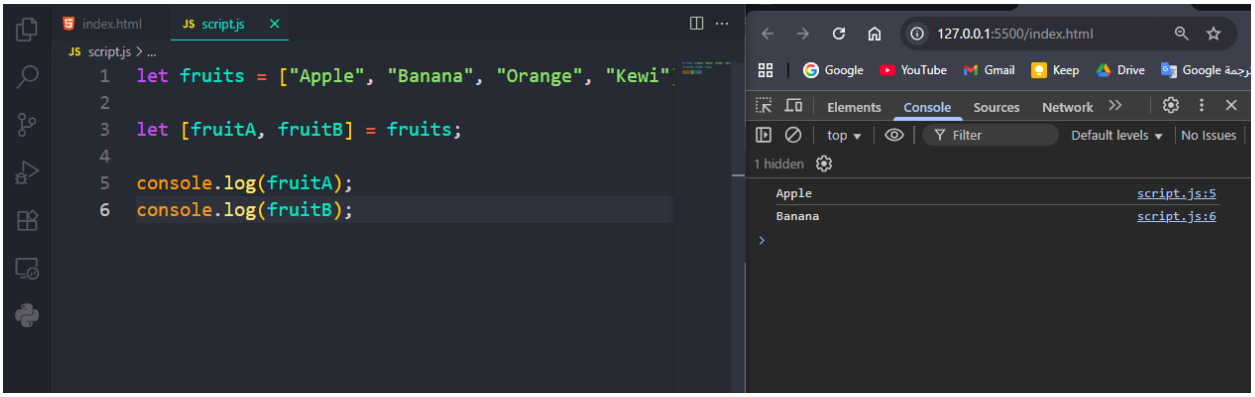


```
let obj1 = {
  name: "User",
  job: "Developer",
  country: "Egypt",
};

let newObj = { ...obj1, name: "Abdelrahman" };
console.log(newObj);
```

```
                                                   script.js:8
▸ {name: 'Abdelrahman', job: 'Developer', country: 'Egypt'}
```

11- Destructure an array to get the first and second elements into variables.



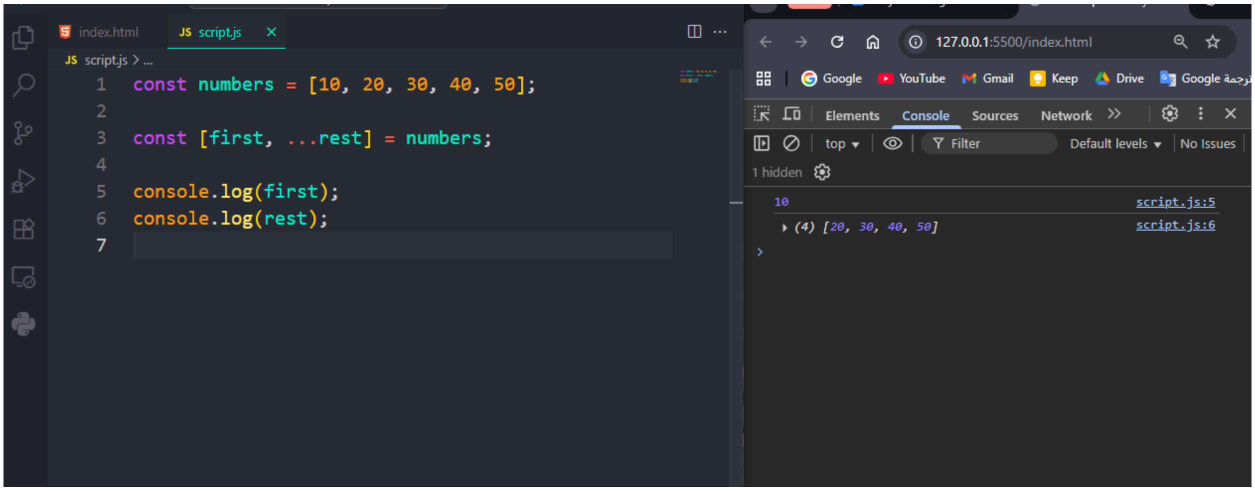
```
let fruits = ["Apple", "Banana", "Orange", "Kewi"

let [fruitA, fruitB] = fruits;

console.log(fruitA);
console.log(fruitB);
```

```
Apple                                script.js:5
Banana                               script.js:6
```

12- Destructure an array to get the first element and the rest into another array.



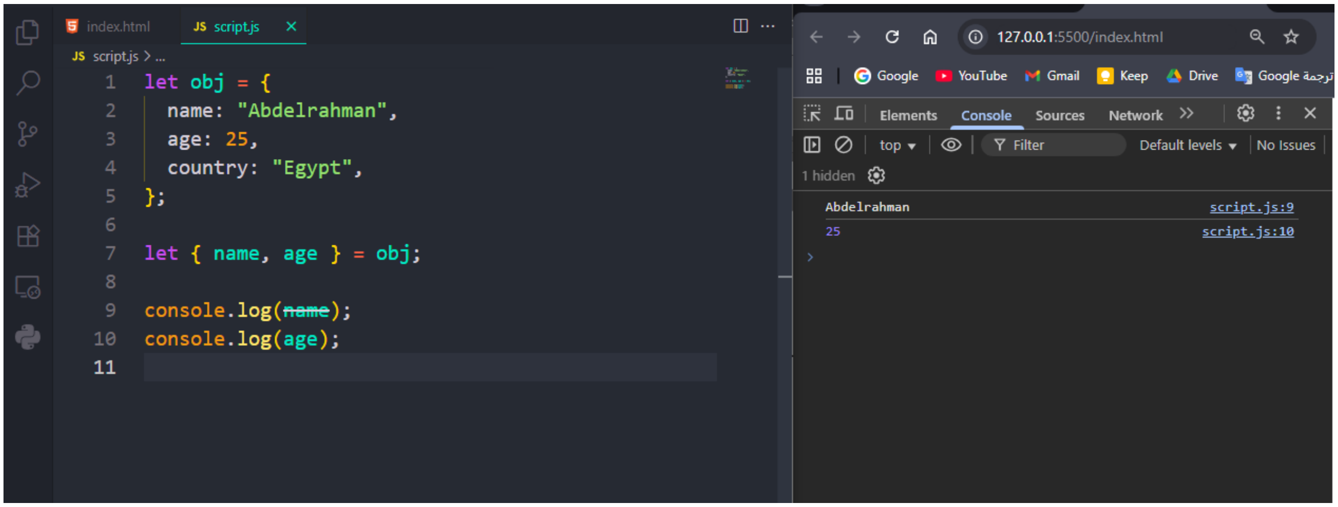
```
const numbers = [10, 20, 30, 40, 50];

const [first, ...rest] = numbers;

console.log(first);
console.log(rest);
```

```
10                                   script.js:5
▸ (4) [20, 30, 40, 50]               script.js:6
```

13- Destructure an object to extract two properties into variables.



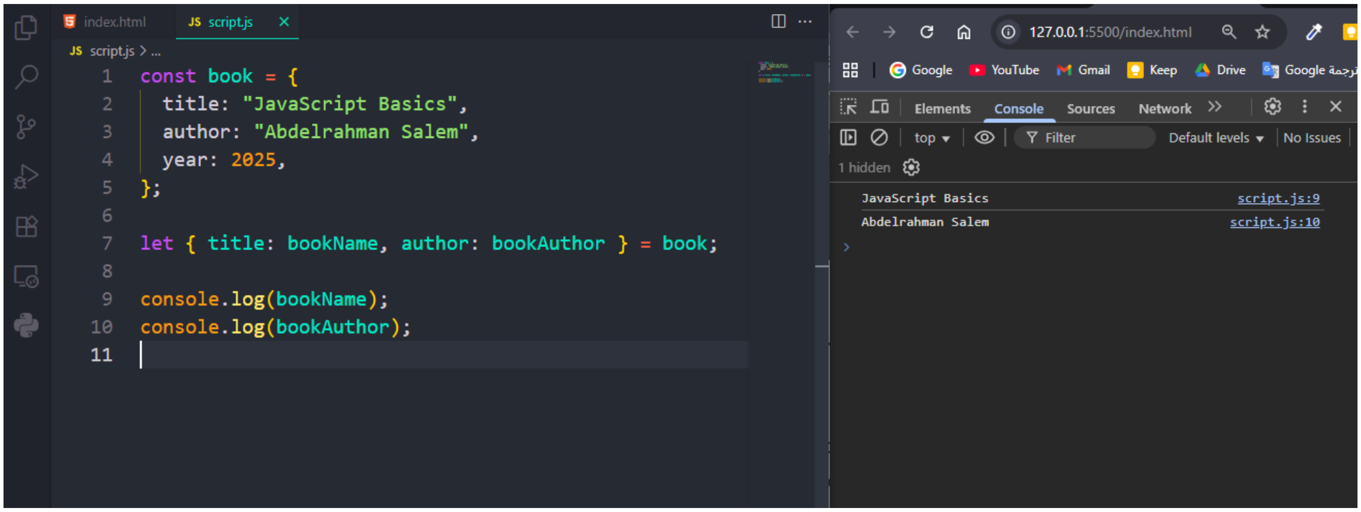
```javascript
let obj = {
  name: "Abdelrahman",
  age: 25,
  country: "Egypt",
};

let { name, age } = obj;

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

```
Abdelrahman                                   script.js:9
25                                            script.js:10
```

14- Destructure an object and rename the extracted properties.



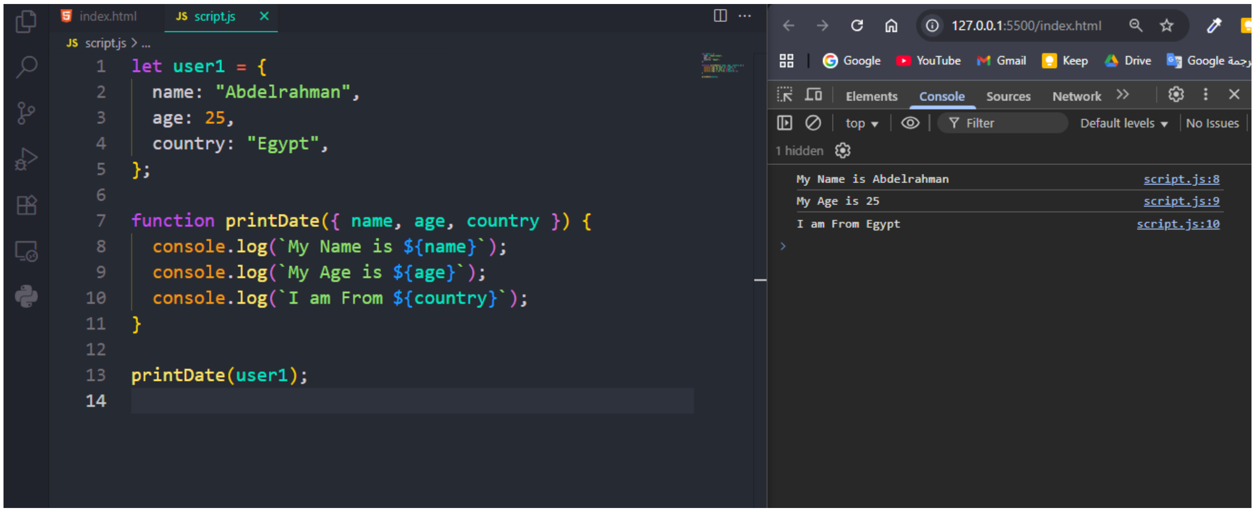
```javascript
const book = {
  title: "JavaScript Basics",
  author: "Abdelrahman Salem",
  year: 2025,
};

let { title: bookName, author: bookAuthor } = book;

console.log(bookName);
console.log(bookAuthor);
```

```
JavaScript Basics                             script.js:9
Abdelrahman Salem                             script.js:10
```

15- Write a function that takes an object as a parameter and uses destructuring in the parameter list to extract specific properties.



```javascript
let user1 = {
  name: "Abdelrahman",
  age: 25,
  country: "Egypt",
};

function printDate({ name, age, country }) {
  console.log(`My Name is ${name}`);
  console.log(`My Age is ${age}`);
  console.log(`I am From ${country}`);
}

printDate(user1);
```

```
My Name is Abdelrahman          script.js:8
My Age is 25                    script.js:9
I am From Egypt                script.js:10
```