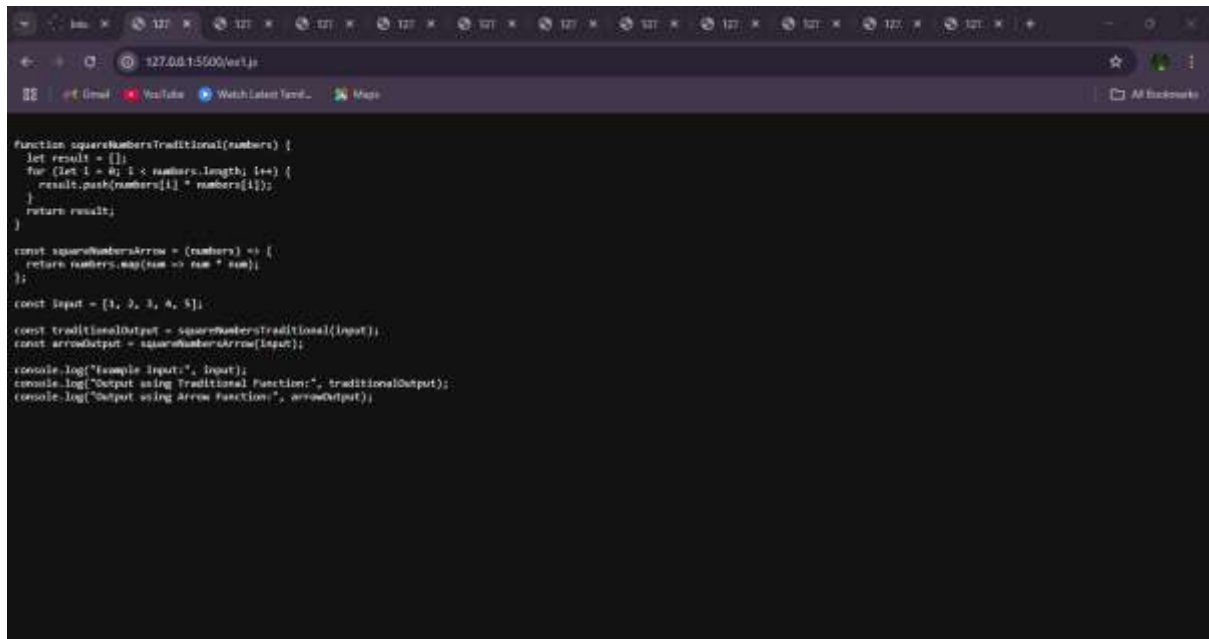


Ex1



```
function squareNumbersTraditional(numbers) {
  let result = [];
  for (let i = 0; i < numbers.length; i++) {
    result.push(numbers[i] * numbers[i]);
  }
  return result;
}

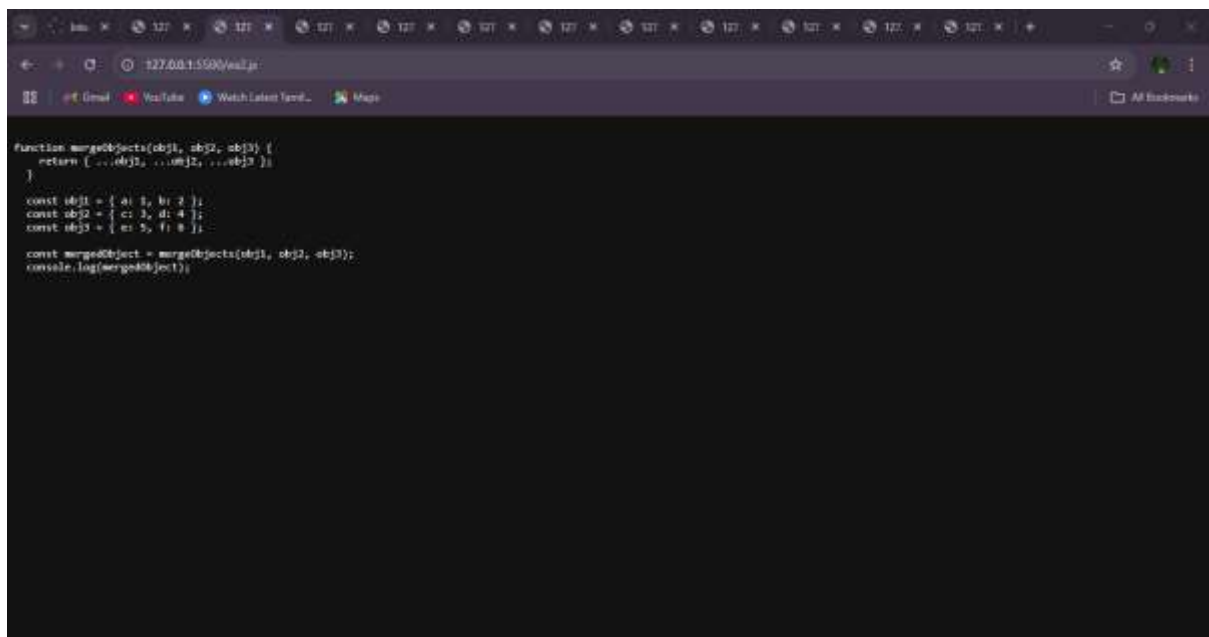
const squareNumbersArrow = (numbers) => {
  return numbers.map(num => num * num);
};

const input = [1, 2, 3, 4, 5];

const traditionalOutput = squareNumbersTraditional(input);
const arrowOutput = squareNumbersArrow(input);

console.log("Sample inputs", input);
console.log("Output using Traditional Function:", traditionalOutput);
console.log("Output using arrow function:", arrowOutput);
```

Ex2



```
function mergeObjects(obj1, obj2, obj3) {
  return [...obj1, ...obj2, ...obj3];
}

const obj1 = { a: 1, b: 2 };
const obj2 = { c: 3, d: 4 };
const obj3 = { e: 5, f: 6 };

const mergedObject = mergeObjects(obj1, obj2, obj3);
console.log(mergedObject);
```

Ex3

```
function getStudentScore(students, name) {
  const studentMap = new Map();
  students.forEach(student => {
    studentMap.set(student.name, student.score);
  });
  return studentMap.get(name);
}

const students = [
  { name: 'Alice', score: 85 },
  { name: 'Bob', score: 92 },
  { name: 'Charlie', score: 78 }
];

const result = getStudentScore(students, 'Bob');
console.log(result);
```

Ex4

```
function removeDuplicates(numbers) {
  const uniqueNumbers = [...new Set(numbers)];
  return uniqueNumbers;
}

const numbers = [1, 2, 2, 3, 4, 5];
const results = removeDuplicates(numbers);
console.log(results);
```

Ex5

```
function filterWordsByLength(words) {
  return words.filter(word => word.length > 3);
}

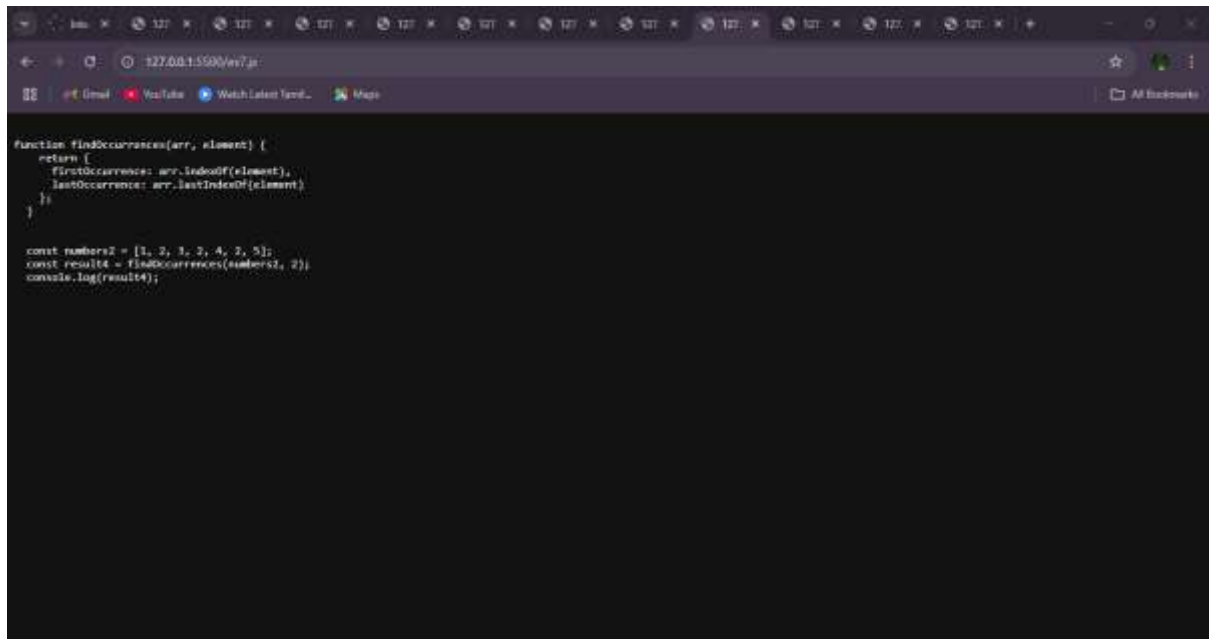
const words = ['apple', 'banana', 'cherry', 'strawberry', 'orange', 'fig', 'grape'];
const result2 = filterWordsByLength(words);
console.log(result2);
```

Ex6

```
function calculateProduct(numbers) {
  return numbers.reduce((product, num) => product * num, 1);
}

const numbers1 = [1, 2, 3, 4, 5];
const result3 = calculateProduct(numbers1);
console.log(result3);
```

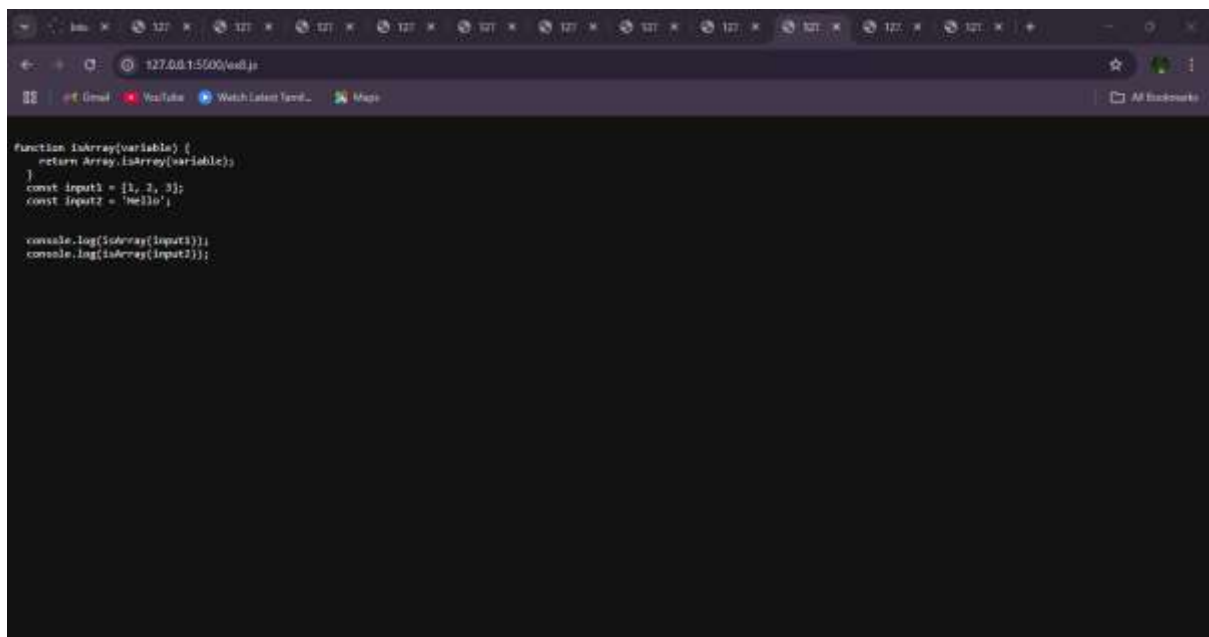
Ex7



The screenshot shows a web browser with the address bar displaying `127.0.0.1:5500/ex7.js`. The browser's bookmark bar includes links to Gmail, YouTube, Watch Later Feed..., and Maps. The main content area displays the following JavaScript code:

```
function findOccurrences(arr, element) {  
  return [  
    firstOccurrence: arr.indexOf(element),  
    lastOccurrence: arr.lastIndexOf(element)  
  ]  
}  
  
const numbers2 = [1, 2, 3, 2, 4, 2, 5];  
const result4 = findOccurrences(numbers2, 2);  
console.log(result4);
```

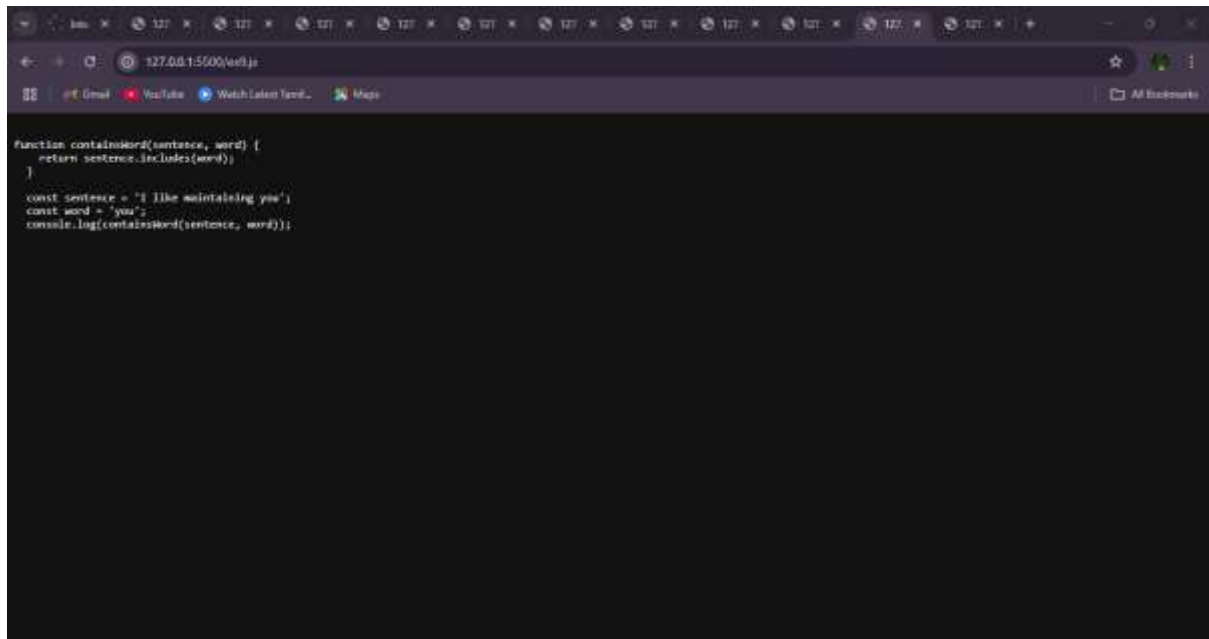
Ex8



The screenshot shows a web browser with the address bar displaying `127.0.0.1:5500/ex8.js`. The browser's bookmark bar includes links to Gmail, YouTube, Watch Later Feed..., and Maps. The main content area displays the following JavaScript code:

```
function isArray(variable) {  
  return Array.isArray(variable);  
}  
  
const input1 = [1, 2, 3];  
const input2 = 'hello';  
  
console.log(isArray(input1));  
console.log(isArray(input2));
```

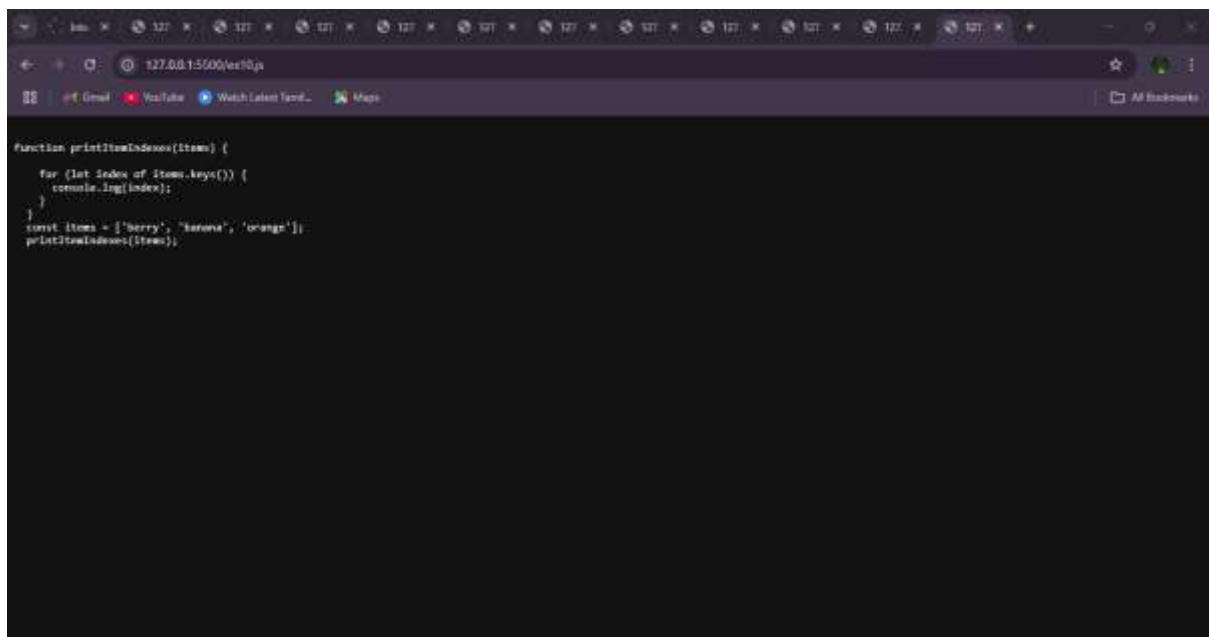
Ex9



A screenshot of a web browser window with a dark theme. The address bar shows the URL `127.0.0.1:5500/ex9.js`. The browser's tab bar at the top shows several tabs, all with the same URL. The main content area displays the following JavaScript code:

```
function containsWord(sentence, word) {  
  return sentence.includes(word);  
}  
  
const sentence = 'I like maintaining you';  
const word = 'you';  
console.log(containsWord(sentence, word));
```

Ex10



A screenshot of a web browser window with a dark theme. The address bar shows the URL `127.0.0.1:5500/ex10.js`. The browser's tab bar at the top shows several tabs, all with the same URL. The main content area displays the following JavaScript code:

```
function printItemIndexes(items) {  
  for (let index of items.keys()) {  
    console.log(index);  
  }  
}  
  
const items = ['berry', 'banana', 'orange'];  
printItemIndexes(items);
```