Core | Deadline: Sunday of Week 1 | Difficulty Level: Basic | Est. Time: 00:00-02:00

# Predict JavaScript I (Core)

Predict the outputs from code blocks featuring ES6 syntax like destructuring, scope and the spread operator!

## Learning Objectives

- Analyze and explain the reasons behind the output of code blocks using modern JS syntax, including the application of destructuring, scope, and the spread operator
- Evaluate potential problems in code blocks and propose appropriate solutions related to modern JS concepts such as destructuring, scope, and the spread operator

Welcome to another **Core assignment**! Some students like to explore the assignments before they're finished reading through the lessons, and that's okay! It can be good for your brain to have a preview of what your future challenges might be. However, before you begin this assignment, it's important that you've first:

- Completed the preceding lesson modules
- Taken the knowledge checks to confirm your understanding
- Viewed lecture material related to the assignment topics
- Completed and submitted your practice assignments

---

# Predict JavaScript I:

This assignment will ask you to recall what you have learned about the new syntax, rules, and techniques associated with modern JS topics like destructuring, scope, and the spread operator. You will be asked to predict the output of each code block and answer questions in the checklist below about some of the problems.

Each code block, along with predictions and question answers should be included in your .js file.

# *Expected Layout Example

Why did the code produce that output? If applicable, how would you get the index value that did not output?

```
 //Problem 1000: Why did the code produce that output? If applicable, how would you get the index value that did not output?
const exampleVar = "example";
const exampleArr = ["ex", "am", "ple"];
console.log(exampleArr);

//1. Prediction of the example output should be in commented-out code here.
//2. The actual output
//3. Answering any questions associated with the problem block here.

//Problem 2: ...ect
```

# Problem 1

Why did the code produce that output? If applicable, how would you get the index value that did not output?

```
const cars = ['Tesla', 'Mercedes', 'Honda']
const [ randomCar ] = cars
const [ ,otherRandomCar ] = cars
//Predict the output
console.log(randomCar)
console.log(otherRandomCar)
```

# Problem 2

Why did the code produce that output? If applicable, what would you need to do to solve any potential problems?

```
const employee = {
    employeeName: 'Elon',
    age: 47,
    company: 'Tesla'
}
const { employeeName: otherName } = employee;
//Predict the output
console.log(otherName);
console.log(employeeName);
```

# Problem 3

Why did the code produce that output? If applicable, how would you alter this code without changing either console.log?

```
const person = {
    name: 'Phil Smith',
    age: 47,
    height: '6 feet'
}
const password = '12345';
const { password: hashedPassword } = person;
//Predict the output
console.log(password);
console.log(hashedPassword);
```

# Problem 4

Why did the code produce that output? Declare a new variable for the value at the 4th index of the array and console.log it.

```
const numbers = [8, 2, 3, 5, 6, 1, 67, 12, 2];
const [,first] = numbers;
const [,,,second] = numbers;
const [,,,,,,,,third] = numbers;
//Predict the output
```

```
console.log(first === second);
console.log(first === third);
```

# Problem 5

Why did the code produce that output? Console.log the last value in the secondKey property's array.

```
const lastTest = {
    key: 'value',
    secondKey: [1, 5, 1, 8, 3, 3]
}
const { key } = lastTest;
const { secondKey } = lastTest;
const [ ,willThisWork] = secondKey;
//Predict the output
console.log(key);
console.log(secondKey);
console.log(secondKey[0]);
console.log(willThisWork);
```

# Problem 6

First, how many scopes does the following code block contain? Define each scope and guess what the output will be.

```
var beatles = ['Paul', 'George', 'John', 'Ringo'];
function printNames(names) {
  function actuallyPrintingNames() {
    for (var index = 0; index < names.length; index++) {
      var name = names[index];
      console.log(name + ' was found at index ' + index);
    }
    console.log('name and index after loop is ' + name + ':' + index);
  }
  actuallyPrintingNames();
}
printNames(beatles);
```

# Problem 7

Why did the code produce that output?

```
 function actuallyPrintingNames() {
  for (let index = 0; index < names.length; index++) {
    let name = names[index];
    console.log(name + ' was found at index ' + index);
  }
  console.log('name and index after loop is ' + name + ':' + index);
}
```

# Problem 8

Why did the code produce that output? Explain the output, including any possible errors and why they occurred. If there are no errors, explain the justification for each keyword used to declare variables.

```
const beatles = ['Paul', 'George', 'John', 'Ringo'];
function printNames(names) {
  function actuallyPrintingNames() {
    for (let index = 0; index < names.length; index++) {
      const name = names[index];
      console.log(name + ' was found at index ' + index);
    }
  }
  actuallyPrintingNames();
}
printNames(beatles);
```

# Problem 9

Why did the code produce that output? Explain why each console.log looks the way it does.

```
const planet = {
        name:"Jupiter",
        milesFromSun: 49849,
        mass: 393983,
            composition: ["gas", "liquid", "oxygen"]
}
const planetCopy = {...planet}
console.log(planet.composition[0] === planetCopy.composition[0])
```

```
console.log(planet === planetCopy)
```

## Saving Your Assignment as a Zip File

Save your .js file as a .zip file and submit it to the Learn Platform in the provided submission space below!

- [ ] Predict the Output for Problem 1 and answer the associated questions.
- [ ] Predict the Output for Problem 2 and answer the associated questions.
- [ ] Predict the Output for Problem 3 and answer the associated questions.
- [ ] Predict the Output for Problem 4 and answer the associated questions.
- [ ] Predict the Output for Problem 5 and answer the associated questions.
- [ ] Predict the Output for Problem 6 and answer the associated questions.
- [ ] Predict the Output for Problem 7 and answer the associated questions.
- [ ] Predict the Output for Problem 8 and answer the associated questions.
- [ ] Predict the Output for Problem 9 and answer the associated questions.