0Trainer guide
Examples and exercises
JS-3 – Destructuring

Contents

DEMO	Trainer demo: Array destructuring *(Corresponds to JS-3a)*
JS-3b	Exercise: Array destructuring
DEMO	Trainer demo: Object destructuring *(Corresponds to JS-3c)*
JS-3d	Exercise: Object destructuring

Trainer demo | Array destructuring

[Link to environment]

In this demo, you will show how to extract elements from an array using destructuring.

index.js

- Here we have an array of fruits.
- If we wanted to assign each fruit in the array to a separate variable, we could do something like this.
- *Write the following code:*

```
const apple = fruits[0]
const kiwi = fruits[1]
const grapes = fruits[2]
```

- But this is not the most elegant way to do this.
- With destructuring, we can do the same thing in a single line.
- *Clear the code above.*
- If we write square brackets after const, we can assign a variable name to each index of the array, like this.
- *Write the following code:*

```
const [apple, kiwi, grapes] = fruits
```

- The position of the variable name corresponds to the index of the element in the array. For example, the “apple” variable name corresponds to the apple emoji at index zero of the array, and the “kiwi” variable name corresponds to the kiwi emoji at index one.
- This is a much more concise way of expressing the same thing we had written previously.

JS-3b | Exercise: Array destructuring

[Link to environment]

This exercise gives learners practice on using array destructuring.

index.js

- This looks very similar to the previous example. We have an array of three emojis, and we want to extract them to three separate variables.

Exercise

- Your task is to rewrite the three lines of code where the variables are extracted, using array destructuring to accomplish the same thing in a single line.

Solve the exercise live before moving on.

Solution

```
const [happy, sad, shocked] = emojis
```

Trainer demo | Object destructuring

Link to environment

In this demo, you will show how to extract elements from an object using destructuring.

index.js

- Destructuring also applies to JavaScript objects as well as arrays.
- Here we have a simple object called `dog` with three properties (`name`, `breed`, and `age`).
- If we wanted to assign each property of this object to a separate variable, we could do something like this:
- *Write the following code:*

```
const name = dog.name
const breed = dog.breed
const age = dog.name
```

- As with the array example, there is a more elegant way to achieve this.
- *Clear the code above.*
- If we write curly brackets after const, we can assign a variable name to each property of the object, like this.
- *Write the following code:*

```
const { name, breed, age } = dog
```

- Again, this helps to make our code cleaner and more concise.

JS-3d | Exercise: Object destructuring

Link to environment

This exercise gives learners practice on using array destructuring.

index.js

- In this exercise, we have an object called `person` which has properties like `firstName`, `lastName` and `age`.
- We also have a function called `displayFullName`, which
    - Takes in a person object as a parameter
    - Extracts the `firstName` and `lastName` from the person object
    - Then writes a message to the console displaying the full name.

Exercise

- Your task is to take the lines where the `firstName` and `lastName` properties are extracted, and use object destructuring to express the same thing in a single line.

Solve the exercise live before moving on.

Solution

- Here is one possible solution:

```
function displayFullName(personObject) {
  const { firstName, lastName } = personObject
  console.log(`Full name: ${firstName} ${lastName}`)
}
```

- It's also possible to destructure an object within the function parameters, like this:

```
function displayFullName({ firstName, lastName }) {
  console.log(`Full name: ${firstName} ${lastName}`)
}
```

- This makes our code even more concise.