JS JavaScript
TUTORIAL

Returning Multiple Values from a Function

Summary: in this tutorial, you will learn to define JavaScript functions that return multiple values.

JavaScript functions can return a single value. To return multiple values from a function, you can pack the return values as elements of an array or as properties of an object.

Returning multiple values from a function using an array

Suppose the following `getNames()` function retrieves the first name and last name from a database in the backend or from the result of a third-party API call and returns them as elements of an array:

```
function getNames() {
    // get names from the database or API
    let firstName = 'John',
        lastName = 'Doe';

    // return as an array
    return [firstName, lastName];
}
```

The following shows how to get the return value from the `getNames()` function:

```
let names = getNames();
```

Because the `names` variable is an array, you can reference its elements using the square brackets, like this:

```
const firstName = names[0],
      lastName  = names[1];
```

In ES6, you can use the destructuring assignment syntax to unpack values from an array more intuitively, like this:

```
const [firstName, lastName] = getNames();
```

In this code, the `firstName` and `lastName` variables will take the first and second elements of the return array.

> A function that returns an array with exactly two elements are commonly used in the useState React hook.

Returning multiple values from a function using an object

If you want to assign a name to each returned value to make it more readable and easier to maintain, you can use an object:

```
function getNames() {
  // get names from the database or API
  let firstName = 'John',
      lastName = 'Doe';

  // return values
  return {
    'firstName': firstName,
    'lastName': lastName
  };
}
```

Since the names of the properties are the same as the variables, you can shorten them using the object literal syntax extensions in ES6 as follows:

```
function getNames() {
  // get names from the database or API
  let firstName = 'John',
      lastName  = 'Doe';
  return { firstName, lastName };
}
```

And you can get the return value as an object like this:

```
let names = getNames();

let firstName = names.firstName,
    lastName  = names.lastName;
```

If you want to unpack properties from an object, you can use the object destructuring syntax as follows:

```
let { firstName, lastName } = getNames();
```

Summary

- JavaScript doesn’t support functions that return multiple values. However, you can wrap multiple values into an array or an object and return the array or the object.
- Use destructuring assignment syntax to unpack values from the array, or properties from objects.