AkiraChix
SHE BUILDS • SHE SERVES • SHE LEADS

Outline

- Import (continuing from last week)
- Callbacks
- Promises
- Revision

Importing

Importing

The import and export statements in JavaScript help you to share code across multiple files.

These statements embrace code splitting, where we distribute code across multiple files to keep it reusable and maintainable. The former is true because we can import a piece of code into multiple files. The latter is true because there is only one source where you maintain the piece of code.

Usually, we put a list of what to import in curly braces import {...}

But if there's a lot to import, we can import everything as an object using import * as <obj>

Importing

Ensure that users.js from last weeks class is in the same folder.

Create and import.js file

```
import {getName, getLocation} from './users.js';

getName('John');

getLocation('Savelberg Retreat Centre');
```

What is your output for the above?

Callbacks

Callbacks

To understand the callback I will make a brief analogy.

Suppose we are talking on the phone. When talking, a situation arises to resolve immediately. We put the call on hold, we do what we have to do and when we finish, we return to the call that we left on hold.

Well, simply with this example we can give us an idea in general, what is a callback. Basically, as the name says it is.

Now, speaking in programming language.

A callback is a function that will be executed when an asynchronous operation has been completed

Callbacks

A callback is passed as an argument to an asynchronous operation. Normally, this is passed as the last argument of the function. Doing this it’s a good practice, so keep that in mind.

The callback structure seems like this:

```
function sayHello() {
    console.log('Hello everyone');
}

setTimeout(sayHello(), 3000)
```

Callbacks

What we did in the above example was, first to define a function that prints a message to the console. After that, we use a timer called setTimeout (this timer is a native Javascript function).

This timer is an asynchronous operation that executes the callback after a certain time. In this example, after 3000ms (3 seconds) will be executed the sayHello function.

Callback Pattern

As we mentioned at the beginning, as great developers we should respect the callback position as a parameter. Which should always be placed as the last one. This has for the name the callback pattern.

In this way, our code will be more readable and will be maintained more easily when other programmers work on it.

Promises

Promises

The promises in Javascript are just that, promises. We know that when we make a promise, it means that we will do everything possible to achieve the expected result. But, we also know that a promise cannot always be fulfilled for some reason.

Just as a promise is in real life, it is in Javascript, represented in another way; in code.

Promises

The constructor of a promise receives an argument: a callback, which has two parameters:

- resolve
- reject

These are functions already defined in Javascript, so we should not build them ourselves.

This callback, which has these two functions as parameters, is called the executor.

The executor runs immediately when a promise is created.

Promises

Let's see an example of a promise:

```
let promise = new Promise(function(resolve, reject) {
    // things to do to accomplish your promise

    if(/* everything turned out fine */) {
        resolve('Stuff worked')
    } else { // for some reason the promise doesn't fulfilled
        reject(new Error('it broke'))
    }
})
```

Promises

Once the executor finishes executing, we will send one of the functions that it has as an argument.

- In case it is fulfilled, we use the *resolve* function.
- In case it fails for some reason, we use the *reject* function.

Promises have three unique states:

- Pending: The asynchronous operation has not been completed yet.
- Fulfilled: The asynchronous operation has completed and returns a value.
- Rejected: The asynchronous operation fails and the reason why it failed is indicated.

Revision

Revision Questions

What does bind method do in JS ?

What is Closure?

What does destructuring mean?

Write a method using this identifier.

Write import.js function using import * as <obj>