Chapter XII Fucther Function

The function are the first class object , this mean we can passed around the same way as every other value, they can have properties and methods, as well accepting other functions as parameter.

Properties and Methods

All function have the propertie length that returns the number of parameters the function has.

The call method can be used to set the value or the replace this inside a function to and object that it provided as first argument.

Example:

function sayHello(greeting='Hello'){

return `${ greeting }, my name is ${ this.name }`;

}

sayHello.call(clark, 'How do you do');

<< 'How do you do, my name is Clark'

sayHello.call(bruce);

<< 'Hello, my name is Bruce'

If a function doesn’t refer to an object as this in its body, it can still be called using the call() method, but you need provide null as its first argument. For example, we could call the square() function using the call() method.

Example:

Square.call(null,4);

You can add a description property to a function that describes wha it does.

If a function takes some time to compute a return value, we can save the result in a cache property. Then if the same argument is used again later, we can return the value from the cache, rather than having to compute the result again.

Example:

Return square.return.

You can invoked a function expresión by placing a parentheses at the end of the function definition (remember we use parenthesis to invoke a function).

Example:

(function(){

const temp = 'World';

console.log(`Hello ${temp}`);

})();

<< 'Hello World'

An IIFE can be used to set up a any initilization code that there’llbe no need again.

We can rewrite a function by calling again.

Example:

function party(){

console.log('Wow this is amazing!');

party = function(){

console.log('Been there, got the T-Shirt');

}

}

JavaScript is a single-threaded environment, which means only one piece of code will ever be processed at a time.

By using callbacks, we ensure that waiting for these tasks to complete doesn't hold up the execution of other parts of the program. Once the task has been completed, the callback will be invoked before returning to the rest of the program.

Generator

To define a generator function, an asterisk symbol ( \* ) is placed after the function declaration.

Functional programming is a programming paradigm. Other examples of programming paradigms include object oriented programming and procedural programming. JavaScript is a multi-paradigm language, meaning that it can be used to program in a variety of paradigms (and sometimes a mash-up of them!). This flexibility is an attractive feature of the language, but it also makes it harder to adopt a particular coding style as the principles are not enforced by the language. A language such as Haskell, which is a purely functional language, is much stricter about adhering to the principles of functional programming.

AJAX - Chapter XIII

Ajax:

Ajax is a technique that allows web pages to communicate asynchronously with a server, and it dynamically updates web pages without reloading.

The term 'Ajax' was coined by Jesse James Garrett in 2005 in the article“Ajax: A New Approach to Web Applications,”where he referred to techniques being used by Google in its recent web applications. Ajax was a neat acronym that referred to the different parts of the process being used: Asynchronous JavaScript and XML.

Asynchronous

When a request for data is sent, the program doesn’t have to stop and wait for the response. It can carry on running, waiting for an event to fire when a response is received. By using callbacks to manage this, programs are able to run in an efficient way, avoiding lag as data is transferred back and forth.

JavaScript

JavaScript was always considered a 'front-end' language, not used to communicate with the server. Ajax enabled JavaScript to send requests and receive responses from a server, allowing content to be updated in real time.

XML

When the term Ajax was originally coined, XML documents were often used to return data. Many different types of data can be sent, but by far the most commonly used in Ajax nowadays is JSON, which is more lightweight and easier to parse than XML. (Although it has never really taken off, the termAjaj is sometimes used to describe the technique.)JSON also has the advantage of being natively supported in JavaScript, so you can deal with JavaScript objects rather than having to parse XML files using DOM methods.

Fetch API

fetch('https://example.com/data')

.then( // code that handles the response )

.catch( // code that runs if the server returns an error )

Json AJAX

fetch(url)

.then( response => response.json() ); // transforms the JSON data into a JavaScript object

The Fetch API is, at the time of writing, what is known as a 'living standard', which means that the specification is being developed 'in the wild'. This means that, despite it being available to use, it’s still subject to change as developers, browser vendors and end-users provide feedback about how it works. It’s an experimental technology, and new features might get added, or the syntax and behavior of some properties and methods might change in the future

Practice

<https://codepen.io/michell17/pen/xxqWLNq?editors=1111>