Dylan Warner

6/3/2021

WDD 330

W07 Readings

**Chapter 11: Further Functions**

* Functions in Javascript are first-class objects which means they can be passed around in the same way as every other value. They can have their own properties and methods, as well as accepting other functions as parameters and being returned by other functions.

Call and Apply Methods-

* The *call()* method can be used to set the value of this inside a function to an object that is provided as the first argument.
* The first argument that call() needs is the value being assigned to this, and the following arguments are the parameters required by the function being called.
* The apply() method is similar except that the arguments of the function are provided as an array even if there is only one.

Custom Properties-

* You can add your own properties to functions by declaring them like such:

*square.description = 'Squares a number that is provided as an argument'*

Immediately Invoked Function Expressions-

* An Immediately Invoked Function Expression – or IIFE – (pronounced 'iffy') is an anonymous function that, as the name suggests, is invoked as soon as it’s defined.
* IIFE’s are created by wrapping the entire function in parentheses and also adding parentheses after the function declaration.

Temporary Variables-

* Variables declared within an IFFE are temporary variables and only available within the scope of the IIFE.

Functions that Define and Rewrite Themselves-

* In Javascript functions can be defined with a redefinition inside of them so that if the function is called again after it is called the first time it will have a new definition.

Recursive Functions-

* Recursive functions are also useful and possible within Javascript.

Function Scope-

* Variables defined within functions can be made available outside of a function by passing it to another function within the function where it is defined.

Generators-

* Generators are functions that produce iterators. Calling a generator does not run the code within the function but creates an iterator that implements a *next()* method.

Currying-

* A function is said to be curried when not all arguments have been supplied to the function, so it returns another function that retains the arguments already provided, and expects the remaining arguments that were omitted when the original function was called. A final result is only returned once all the expected arguments have eventually been provided.

**Chapter 13: AJAX**

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

Clients and Servers-

* Examples of clients are web browsers and web browsers make a request to a server where there are resources stored, usually web pages, which are processed by the client and served to the user.
* AJAX can be used to request as well as send information to a server.
* Data can be received from the server in different forms, but is usually sent in JSON.
* **Response Interface** - deals with the object that’s returned when the promise is fulfilled. Response objects have a number of properties and methods that allow us to process the response effectively.
* **Request Interface** - We can get more fine-grained control over the request being made by providing a Request object as an argument.
* **Headers Interface** - HTTP headers are used to pass on any additional information about a request or response. Typical information contained in headers includes the file-type of the resource, cookie information, authentication information and when the resource was last modified.