Week III – Object Method this.

* Object usually are created to represent entities fo the real world,like users, and so on . In the real world a user can act in JavaScript action are represented by function and properties.
* It’s common that and object method need to acces the information stored in the object to do it’s job, to acces the object , a method can use the THIS keyword.
* We can use the keyword this in any fuction, even if it’s not a method o fan object.

Practice :

<https://jsfiddle.net/0bqnmpw7/>

Chapter IV – Objects

* Object literal in avaScript is a self-contained set of related values and functions. They act as a collection of named properties that map to any JavaScript value such as strings, numbers etc.
* Creating object literal example:

Const Spiderman = { };

Créate an object using a constructor function

Const Spiderman = new object ();

* the object literal notation is the preferred way of creating objects. The obvious reason is because it requires less typing and provides a concise way of initializing an object and its properties in one statement.
* Calling a method we can use the dot notation or the Brackets

Example:

superman.fly()

<< 'Up, up and away!'

superman['fly']()

<< 'Up, up and away!'

* The in operator can be used to check whether an object has a particular property.
* Another way is to use the hasOwnProperty() method. As mentioned earlier, objects can inherit properties from other objects, so all objects have a method called hasOwnProperty() . This can be used to check whether an object has a property that is its own, rather than one that has been inherited from another object.
* An object literal can be passed as a parameter to a function. This is useful when there are a large number of parameters, as it allows the arguments to be provided by name and in any order. This means you don't have to remember the order to enter them when invoking a function.

Chapter VI – The DOM

* The document object allows you to Access elements of a web page and enable interaction with the page by adding and removing elements , changing the order,content and attributes of elements etc.
* We can use the nodeType object number to find out what type of node it is.

Example:

Body.nodeType

What type of node is

1= element

2=attribute

3=text

4=comment

5=body

We can also use the nodeName property to find the name of the element

Example :

Body.nodeName

* Method DOM level 0 we can Access commonly
* Document.body returns the body element of a web page, as we saw in the previous example.
* Document.images returns a node list of all the images contained in the document.
* Document.links returns a node list of all the <a> elements and <area> elements that have an href attribute.
* Document.anchors returns a node list of all the <a> elements that have a name attribute.
* Document.forms returns a node list of all the forms in the document. This will be used when we cover forms in Chapter 8.
* QuerySelector
* he document.querySelector() method allows you to use CSS notation to find thefirstelement in the document that matches that matches a CSS selector provided as an argument. If no elements match, it will return null .
* The document.querySelectorAll() method also uses CSS notation but returns a node list ofallthe elements in the document that match the CSS query selector. If no elements match, it will return an empty node list.
* The querySelector() method can be called onanyelement, rather than just document .
* The childNodes property is a list of all the nodes that are children of the node concerned.
* The parentNode property returns the parent node of an element.
* The nextSibling property returns the next adjacent node of the same parent.
* The previousSibling property returns the previous adjacent node.
* The contains method will check to see if an element has a particular class
* A text node can be created using the document.createTextNode() method.

Chapter VII – Events

* Even Listeners

The addEventListener() method is called on a node object, the node to which the event listener is being applied, The addEventListener() method can also be called without a node, in which case it is applied to the global object, usually the whole browser window.

Practice:

<https://jsfiddle.net/wb0ptk6d/>

<https://github.com/michell179708/web-fronted/blob/main/week03/practice.html>