Week 04 Notes

* The action attribute on a form element is the URL of where the input will be sent to.
* Instead of using the index of the form element on the DOM, give it a name attribute so you can access it by name.
* Preferably use the [' ‘] notation to avoid getting confused with methods that work on the form object.
* Question: If I understand correctly, adding a type of “submit” to a button allows us to listen for an event called “submit” ?
* Encapsulation example: When you drive a car you don’t know what is under the hood
* Question: I thought that abstraction was not needing to know how a method is implemented and encapsulation is guaranteeing that data is always valid?
* Inheritance example: An apple and an orange share some attributes. They are both spherical. They can inherit that attribute from a parent sphere class.
* Polymorphism example: There are different kinds of submit buttons all across the web, they are all submit buttons but some do different things like some submit files and others submit messages.
* Question: I see how static methods can be defined in the class declaration, I do not understand how they are useful in the textbook example as it seems that the static method “description” is only defined but never called by the class. I understand that static methods cannot be called by instances. My initial thought therefore would be that the static keyword is a way to abstract a method from the instances of that class, but it can add functionality to a class if the class definition needs to call it.   
  I guess my question is: how are static methods different than private methods? They seem to be the same thing to my understanding.
* Prototypes are especially useful in adding functionality to classes after they are defined or when you do not have access to the class source code.
* Question: I get that it is more memory efficient to share one memory slot for an attribute across all instances of a class, but what happens when you change that value? Is that value changed across all instances of that class? Or is it changed just on that instance?
* Some properties are enumerable some are not
* I did not know you could do monkey patching in JavaScript. You can add functionality to built-in objects.
* Question: When creating a new object, does the class constructor get called automatically?
* You can export functions to another file with the export keyword