Dylan Warner

5/15/2021

WDD 330

W04 Readings

**Chapter 8 - Forms**:

* Forms had traditionally been processed in the backend using PHP, but we can now with Javascript process form information on the frontend.
* Form elements can be accessed through the document object. The document object contains a “forms” property through which we can access all of the forms contained on a page using the square bracket notation. Ex. const form = document.forms[0];
* “const form = document.forms[0];” is equivalent to “const form = document.getElementsByTagname('form')[0];”
* All of the elements of a form can also be accessed through the form object Ex. form.elements;
* Forms have a number of useful properties and methods: form.submit(), form.reset(), form.action().
* Form events also exist that can help: focus, blur, change, submit.
* The submit event is the most important. In javascript we intercept the form before it is sent to the backend or sent to a different URL and act on the form data in the frontend.
* There are several different input fields also that we will change according to the type of input that we want to include in the form: text, password, checkbox, radio, hidden, file, and others.
* Drop-down lists can also be included in forms.
* Text-area tags can be used in forms when large amounts of text want to be entered.
* Form validation can also be done in the frontend by setting requirements for input fields and rejecting input if requirements are not met.

**Chapter 12 - Object-Oriented Programming in JavaScript:**

* Encapsulation - each part of our program should not need to know about any other part of our application to function.
* Polymorphism - various objects can share the same method, but also have the ability to override shared methods with a more specific implementation.
* Inheritance - take an object that already exists and inherit all its properties and methods. We can then improve on its functionality by adding new properties and methods.
* Classes - Classes are used to define a blueprint for an object.
* Constructor Function - A way to create objects.

const Dice = function(sides=6){

this.sides = sides;

this.roll = function() {

return Math.floor(this.sides \* Math.random() + 1)

}

}

If this is done a new object can be made using the new operator: const redDice = new Dice();

* Objects can be created from other objects, and then added to.

**Chapter 15 - Modern JavaScript Development:**

* Javascript Module - A module is a self-contained piece of code that provides functions and methods that can then be used in other files and by other modules.
* Modules help with organization efforts and make files reusable which improves maintainability.