Week 06 – Object Oriented JavaScript

**JS OBJECTS:**

\*\*In JavaScript, almost "everything" is an object.

\*\*All JavaScript values, except primitives, are objects.

* Booleans can be objects (if defined with the **new** keyword)
* Numbers can be objects (if defined with the **new** keyword)
* Strings can be objects (if defined with the **new** keyword)
* Dates are always objects
* Maths are always objects
* Regular expressions are always objects
* Arrays are always objects
* Functions are always objects
* Objects are always objects

JS PRIMITIVES:

\*\*A **primitive value** is a value that has no properties or methods.

\*\*A **primitive data type** is data that has a primitive value.

* JavaScript defines 5 types of primitive data types:
* String, number, Boolean, null, undefined

\*\*Primitive values are immutable (they are hardcoded and therefore cannot be changed).

OBJECTS ARE VARIBALES CONAINING VARIABLES:

\*\*Variables 🡪 contain single values

* Objects
* Objects are Variables
* Containing many values 🡪 written in name : value pairs
* Objects are collections of named values 🡪 called properties
* Objects are containers for named values 🡪 called properties & methods
* \*\*Objects written as name value pairs are similar to:

- Associative arrays in PHP

- Dictionaries in Python

- Hash tables in C

- Hash maps in Java

- Hashes in Ruby and Perl

OBJECT PROPERTIES:

* Named values
* Can be both primitive values, other objects, and functions

OBJECT METHODS:

* Actions that can be performed on objects
* An **object method** is an object property containing a **function definition**

**CREATING A JS OBJECT:**

* Define and create a single object, using an object literal
* Define and create a single object, with the keyword new
* Define an object constructor, and then create objects of the constructed type.
* OBJECT LITERAL: \*\*\*
* Easiest way to create an object
* Define & create an object in one statement
* A list of name : value pairs inside curly braces

var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};

* KEYWORD NEW:
* var person = new Object();  
  person.firstName = "John";  
  person.lastName = "Doe";  
  person.age = 50;  
  person.eyeColor = "blue";
* OBJECT CONSTRUCTOR (Function): \*\*
* Creates an ‘Object Type’ that can be used to create many objects of one type

function person(first, last, age, eye) {  
    this.firstName = first;  
    this.lastName = last;  
    this.age = age;  
    this.eyeColor = eye;  
}

\*The function ‘person’ is an object constructor

\*Once you create an object constructor, you can create new objects of the same type

var myFather = new person("John", "Doe", 50, "blue");  
var myMother = new person("Sally", "Rally", 48, "green");

KEYWORD **this:**

* In JavaScript, the thing called **this**, is the object that "owns" the JavaScript code.
* It’s value, when used in a function, is the object that "owns" the function.
* It’s value, when used in an object, is the object itself.

\*\*JS Objects are mutable – addressed by reference, not by value

JS OBJECT PROPERTIES: