**Lesson 9 - JavaScript:**

**Functions, Arrays, & Establishing a Domain**

**Lectures Notes**

**Lecture Notes**

1. Establishing a domain
   1. Think “what describes an object?”
      1. These are the properties
      2. Subject to change, or constant
      3. Color, NumberOfFingers, Id, Health
   2. And think “what can this object do?”
      1. These are the methods
      2. Describes some sort of action that this object is capable of.
      3. Jump(), Click(), Dispense(), Decay()
   3. Example:
      1. Vending Machine
         1. Properties
            1. NumCandyBars
            2. MoneyInMachine
            3. Weight
         2. Methods
            1. Dispense()
            2. GiveChange()
            3. ReadDollarBill(dollar)
      2. User
         1. Properties
            1. CashOnHand
            2. BellySize
         2. Methods
            1. Take(candyBar)
            2. GetDollarOutOfWallet() // returns dollar
   4. Bad Example!
      1. Spice Cabinet
         1. Properties
            1. NumSpices
            2. Spices // returns array
         2. Methods
            1. GetSpice()
            2. AddSpice()
         3. Note
            1. Cabinents in the real world can’t -do- anything!
            2. On their own at least…
   5. Fixed Example
      1. Spice Cabinent
         1. Properties
            1. NumSpices
            2. Spices
         2. Methods
            1. None
      2. Chef
         1. Properties
            1. SpicesOnHand
         2. Methods
            1. TakeSpiceFromCabinent(Cabinent)
            2. AddSpiceToCabinent(Cabinent)
   6. Try to think of what is possible in the real world and it will make your virtual world make more sense!
2. Fun with functions!
3. **Start with an object**

var car = {

wheels: 4,

burnGas: function(){},

accelerate: function(){ burnGas(); },

brake: function(){}

};

console.log(car.wheels);

car.accelerate();

car.brake();

1. **Convert it to a function**

* Burn gas is no longer accessible outside of the car
* Burn gas gets an underscore because it is private
* Parentheses are used at the end of car’s definition
  + These parenthesis call the car function()
  + And then car “gets” the results of the function
  + This is convenient, so that outside callers do not have to change anything at all!
* It is very easy to convert from an object to to a function without breaking anything!

var car = function(){

var wheels = 4;

var \_burnGas = function(){};

var accelerate = function(){ \_burnGas(); };

var brake = function();

return {

wheels: wheels,

accelerate: accelerate,

brake: brake,

};

}();

// Thankfully we don’t have to do console.log(car().wheels);

console.log(car.wheels);

car.accelerate();

car.brake();