BYU-IDAHO

# JavaScript Objects

- Object Creation Functions
- Inheritance
- Properties
- Methods
- Instantiation

Oswaldo Rodríguez

BYU-IDAHO

# What are objects in JS?

An object is a collection of properties and methods (functions). **Objects in JavaScript, just as in many other programming languages, can be compared to objects in *real life*.**

For example, a real life object could be a cup. A cup has a color, a design, a weight, etc.

BYU-IDAHO

## How to create objects in JS?

Although, there are several ways to create objects in JS, I'm going to show you a couple of them:

Using the keyword new.

```
var myCar = new Object();
```

Creating an empty object;

```
var table = {};
```

BYU-IDAHO

# What are properties?

A property of an object can be explained as a variable that is *attached* to the object. Object properties are basically the same as ordinary JavaScript variables, except for the attachment to objects. The properties of an object define the characteristics of the object.

```
var myCar = new Object();
//Let's add properties to our myCar Object
    myCar.make = 'Ford';
    myCar.model = 'Mustang';
    myCar.year = 1969;
    myCar.mote;
    myCar['mote'] = 'toretto';
console.log(myCar);
```

BYU-IDAHO

## How to add the properties?

You can create the properties of an object with a simple dot-notation, or with bracket notation.

```
var myCar = new Object();
//Let's add properties to our myCar Object

      myCar.make = 'Ford';
      myCar['year'] = 1969;
      myCar.mote;
      myCar['mote'] = 'toretto';
console.log(myCar);
```

```
var myLaptop = new Object();
//Let's add properties to our myLaptop Object
      myLaptop.brand = 'Dell';
      myLaptop['model'] = 'Inspiron';
      myLaptop.serie;
      myLaptop['serie'] = 3000;
console.log(myLaptop);
```

Now that we know how to create objects and add properties to it, we can learn two different ways for creating objects.

BYU-IDAHO

## How to create objects function in JS?

In the following examples we are going to create objects using the keywords Function and Class. We use the 'this' keyword to create a reference to the property.

Declaring a function:

```
let myPhone = function(brand, model, year){
    this.brand = brand;
    this.model = model;
    this.year = year;
}
console.log(myPhone);
```

Defining a Class:

```
class Object{
    constructor(v1, v2){
        this.v1 = v1;
        this.v2 = v2;
    }
}
console.log(Object);
```

These two ways make our objects reusable!

BYU-IDAHO

## Methods in Objects JS?

Our objects will have a behavior which can be called Functions, or Methods. A method is declared as a property of an object, and can be declared when we create the object or later.

Declaring a function:

```
let myPhone = function(brand, model, year){
    this.brand = brand;
    this.model = model;
    this.year = year;
    this. method = function(){
        console.log(this.brand);
    }
}
console.log(myPhone);
```

Defining a Class:

```
class Object{
    constructor(v1, v2){
        this.v1 = v1;
        this.v2 = v2;
    }
    v3(){
        console.log(this.v1);
    }
}
console.log(Object);
```

These ones make the code reusable!

BYU-IDAHO

# Instantiation and Inheritance in JS

If we need to create a new object which has the same properties that an existing object already has, we can create an instance of an object (*instantiate*) and this will *inherit* all the properties and methods of the existing object. Let's see how it works.

```
let  myPhone  = function(brand, model, year){
      this.brand = brand;
      this.model = model;
      this.year = year;
      this. method = function(){
                  console.log(this.brand);
      }
}

//instance of the myPhone Object
const phone1 = new myPhone('iPhone', 'X', '2018');
console.log(phone1);
phone1.method();
```

BYU-IDAHO

# Instantiation and Inheritance in JS

Defining a Class:

```
class Rectangle{
    constructor(width, length){
        this.width = width;
        this.length = length;
    }
    area(){
        console.log(this.width * this.length);
}
// Instance of the Rectangle Object
const rect1 = new Rectangle(20, 40);
console.log(rect1);
rect1.area();
```

BYU-IDAHO

# Comparing Function and Classes

Declaring a function:

```
var myPhone = function(brand, model, year){
        this.brand = brand;
        this.model = model;
        this.year = year;
        this. method = function(){
                console.log(this.brand);
        }
}
const phone1 = new myPhone('iPhone', 'X', '2018');
console.log(phone1);
phone1.method();
```

Defining a Class:

```
class Object{
        constructor(v1, v2){
                this.v1 = v1;
                this.v2 = v2;
        }
        v3(){
                console.log('these are my two properties '
+ this.v1 + this.v2);
        }
}
const obj1 = new Object('firstVa', 'secondVa');
console.log(obj1);
obj1.v3();
```

BYU-IDAHO

- Thanks for your attention!
- Resources:
- https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/Basics

# The End