MAKING OBJECTS

Dr. Kim Lam

Review objects

- Remember that objects have
  - Properties
  - Methods/functions

- We have used objects like HTML elements, or document.
- where:
- var element = document.getElementById("example");
- element.innerHTML = "Some Text";

- Where element is the object, innerHTML is the property.
- We use the "." to access the property of the object.

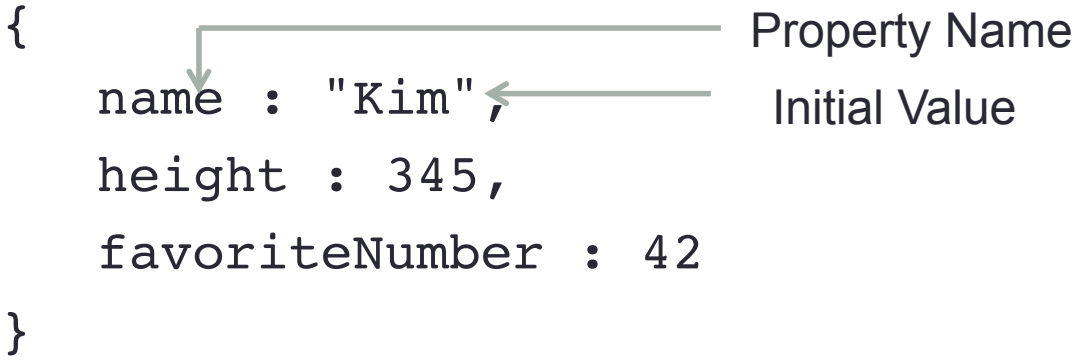
What are objects?..in JavaScript.

- Objects allow us to Group data/variables and functionality/ functions into one logical package with meaningful names
- Data/Properties:
  - Strings
  - Numbers
  - Boolean
  - Other Objects
- Functionality/Methods:
  - Functions
    - These functions are special, they have access to the objects data

Creating an object, by defining it

- We can create objects by using JavaScript Object Notation(JSON).

It is simply a list embedded in curly braces:



We see that property name(or variable name) is followed by a colon followed by the initial value. If there are more than one property, it is followed by a comma. We can mix different types for the properties.

Accessing properties

- We use the dot, “.”, operator to access properties inside and object.
- It’s a list of key value pairs!
- We have used this before to access properties of many objects that JavaScript provides.

```
program:
var myObject = { name: "Kim",
                 height: 345,
                 favoriteNumber: 42
}
console.log(myObject.name);
Output:
Kim
```

Example #1: Creating an object with JSON

- Lets consider the problem, of weather data. We going to make an object called Monday weather report.
- Data:

Temperature	3C
Chance of rain	10%

```
var myObject = { temperature: 3,
                 chanceOfRain: 0.1
}
```

The expression

```
myObject.temperature
```

will have a value of 3.

Example #2: Creating a point

- We can also represent points

x	100
y	200

```
var myPoint = { x: 100,
                y: 200}

The expression

myPoint.x

will have a value of 100
```

Changing the value of properties

- We can change the values of properties in object by using the assignment operator.
- `var myPoint = { x: 100,`
- `                y: 200}`

```
myPoint.x = 200;
```

will change the value of the x property to 200.

Assignment, function parameters and the object.

- Objects are assigned via reference.
- Similarly, object as parameters are also passed by reference.

- That is

a = { x: 1, y:2}

b = a;

- causes b and a to refer to the same object!
- a.x = 5, will also causes the changes to be visible to b!

Creating an object through the Constructor.

- A constructor is a function meant to setup an object.
  - This is a **working** definition
  - Constructors are handy if we want to create multiple objects with the same properties and methods
- We use the new keyword, to create an **new** generic object and then the constructor fills in the details.
- **eg:**

```
var obj = new Point(10,10);
```

- If we forget to use the word new, point() is just a regular function call. Doing this will usually have unintended side-effects.

Example Constructor

- function MyConstructor() {

```
        this.temperature = 3;
        this.chanceRain = 10;
        this.temperatureC = tempC;
        this.temperatureF = tempF;
        //There is no return statement
      //for constructors
}
//Object creation
var myObject = new MyConstructor();
```

Constructors can have parameters

- function CreatePoint(x,y) {

```
        this.x = x;
        this.y = y;
};

//Object creation
var myObject = new CreatePoint(100,200);
```

Example Constructor

- `var myObject = new MyConstructor();`

- So, the new keyword creates a new object
- The statement on the right hand side evaluates to an object, after MyConstructor finishes executing
  - This is because of the new keyword.
  - If we just ran the function MyConstructor, it would not evaluate to an object.
- That object reference is then assigned to myObject.

Arrays of objects

We said that Arrays can hold anything, including objects.

Let's may an array of points:

```
var listOfPoints = [new CreatePoint(100,200),
                    new CreatePoint(200,200),
                    new CreatePoint(300,300)];
Will result in

listOfPoints =>[{x:100,y:200},
                {x:200,y:200},
                {x:300,y:300}];
Accessing:
listOfPoints[0].x => 100
listOfPoints[0].y => 200
```

Methods 1/2(optional)

- Methods are functions than can use an objects internal data.
- Methods should not be run without an object
  - In JavaScript and JavaScript **only** methods can exist independently of objects.
- To access an object's data inside a method, we use the keyword **this**
- To add methods, they are added like any other properties

Methods: Object with an method

```
var point = {
    x : 100,
    y: 100,
    distance: function() {
        return Math.sqrt(this.x*this.x + this.y*this.y);
   }
}
```

When the point changes, the return value of the distance method will also change, because it uses the objects x and y values.