

JS

Pull from upstream!

Commit any changes first!

Slides created in conjunction with: Ilchul Yoon, Nelson Padua Perez

Agenda

- Objects
- Prototypes
- Classes
- Writing to the DOM

Exercise 2 Questions

Let's look at the exercise description!

Objects

- Just a collection of properties
 - You can define your own, browser predefines a set of objects
 - A property can be seen as a variable associated with a value
 - Approaches to access and add properties
 - Using dot-notation
 - Using square brackets

Objects

- Property – association between a name and a value
 - When the value is a function the property is referred to as a method
 - Name can be any valid JavaScript string or anything that can be converted to a String (that includes empty string)
 - Any invalid property name can only be accessed using square bracket notation

How do we create Objects?

- Using Object Constructor
- Using Object Initializer/literal notation
- Using Object.create

Objects as Maps

- We can also view an object as an entity that associates values with strings.
 - Use the [] operator

Ex: `myObj.value == myObject["value"]`

Object Type

- All objects in JavaScript are descended from Object
- All objects have a property called `__proto__`
- The `__proto__` property points to an object (called prototype) from which properties are inherited
- Objects inherit methods and properties from `Object.prototype`
- Prototype chain
 - Set of objects defined by the `__proto__` property
 - The end of the chain is a prototype with the null value (`Object.prototype.__proto__`)

Object Prototypes

- A better way of using `Object.create()`
- Will force inheritance of properties from a parent
- Allows us to have functions across all functions

```
ObjectName.prototype.methodName(){}  

```

All objects will now have that method available.

Object Prototypes

- Methods:
 - `Object.prototype.hasOwnProperty(prop)`
 - `prop` is a direct property (not inherited through the prototype chain)
 - `Object.prototype.isPrototypeOf(obj)`
 - `Object.prototype.toString()`
 - Returns a string representation of the object
 - `Object.prototype.valueOf()`
 - Returns the primitive value of the specified object
 - In ES6, `Symbol.toPrimitive` is a symbol that specifies a function valued property that is called to convert an object to a corresponding primitive value.

Object Constructors

- Rather than handwriting all values in an object, Javascript allows for Object Constructors

Ex:

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

Classes in JavaScript

- Use keyword class
- Constructor is no longer using function, use constructor instead
- Methods can be defined with no other keywords necessary
- Not hoisted!

Let's create an Object!

What is `this`?

- Outside of any object, it refers to the global object window **or** is undefined (if you “use script”)
- Arrow functions have no concept of this.
- When in an object, it refers to the current Object
 - Works the same as in Java
 - `This.data` to access a data field in your object

Inheritance in JavaScript

- Classes *extend* each other
- References to the superclasses' methods and constructors must use the *super* keyword
- If the superclass is not created using *class*, you must link the prototypes!

JavaScript DOM Manipulation

Accessing Information:

- `document.getElementById('myID');`
- `document.getElementsByTagName('p');`
- `document.getElementsByClassName('mainMenu');`

Basics of Writing To Document from JavaScript

For now, we will only learn one way to dynamically write html from our JavaScript:

```
document.writeln("html tags and text here");
```

For example: `document.writeln("<p>Paragraph Text</p>");`

Basics of Writing To Document from JavaScript

You may also embed variables into your html now!

For example:

```
let x = "Station Wagons";
```

```
document.writeln("<p>My favorite cars are " + x + "</p>");
```

Most of the examples posted use this, so test it out!

Advanced DOM Manipulation

- `element.innerHTML = new html`
- `element.attribute = new values`
- `element.style.property = new style`
- `element.setAttribute(attribute, value);`

Advanced DOM Manipulation

- `document.createElement(element)`
- `document.removeChild(element)`
- `document.appendChild(element)`
- `document.replaceChild(new, old)`
- `document.write(text)`

WTWAW

After today make sure you know how to:

- Create a symbol (and know it's use)
- Use and manipulate maps and sets
- Create Objects all 3 ways
- Create an object constructor
- Use `document.writeln()`;