ITMD 465/565
Rich Internet Applications

# Lecture 5

Fall 2019 – September 18, 2019

## Tonight’s Agenda

- Add 1 feature to the DOM Demo from last week
- Continue with last weeks slides we didn’t finish (OOP demo)

# JavaScript OOP Patterns

Let’s simulate Traditional OOP patterns in JS

# JavaScript OOP Patterns

- Patterns to implement Object Oriented Programming in JavaScript
- We will be looking at a few basic ones:
  - **Object Literal**
  - **Constructor Function (Constructor Pattern)**
  - **Constructor Function with Prototype (Constructor Pattern)**
  - **Function that returns an object (Factory Pattern)**
  - **ES6 Classes will be later**
- https://leoasis.github.io/posts/2013/01/24/javascript-object-creation-patterns/

# JavaScript OOP Pattern References

- JavaScript Object Creation Patterns
  - http://leoasis.github.io/posts/2013/01/24/javascript-object-creation-patterns/
- JavaScript OOP Patterns (3 main ones and how they implement OOP)
  - http://javascript.info/tutorial/oop
- Learning JavaScript Design Patterns Online Book (More in depth patterns)
  - https://addyosmani.com/resources/essentialjsdesignpatterns/book/#singletonpatternjavascript
- Douglas Crockford – Private Member in JavaScript
  - http://javascript.crockford.com/private.html
  - Describes public and private members and methods in Constructor pattern
- Douglas Crockford - http://www.crockford.com/

## Object Member Visibility

# Public, Private, Privileged

- These are ways we can have private and public members and methods for objects when we model them with the Constructor pattern.
- Functions and variables assigned in a constructor with the this keyword will be publically visible. Functions and variables assigned normally will be private.
- Be careful, the this keyword can get bound to the window object so it is common to see a `var that = this;` line in the object to bind that to the proper this.
- We will do an example to show these three ideas.
- http://javascript.crockford.com/private.html
- http://robertnyman.com/2008/10/14/javascript-how-to-get-private-privileged-public-and-static-members-properties-and-methods/

## Simple Visibility Example

```
function Person(fname, lname) {
        this.name = “person: ” + fname;
        var fullName = fname + “ “ + lname;

        function print() {
                alert(fullName);
        }

        this.render = function(){
                print();
        };
}

var myPerson = new Person(”Brian”, “Bailey”);
```

What can I access in here?
fullName
print()
this.*

What can I access out here?
myPerson.name
myPerson.render()

## JavaScript Closure

- A closure is an inner function that has access to the outer (enclosing) function's variables—scope chain
- http://javascriptissexy.com/understand-javascript-closures-with-ease/
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Closures
- http://www.javascriptkit.com/javatutors/closures.shtml

## JS Scope

- Here are some links to further explain JavaScript Scope.
- https://toddmotto.com/everything-you-wanted-to-know-about-javascript-scope/
- https://toddmotto.com/understanding-the-this-keyword-in-javascript/
- https://toddmotto.com/es6-arrow-functions-syntaxes-and-lexical-scoping/

## Assignments

## Reading/Assignments

- Online Quiz will be posted this weekend and emailed out. Must be completed before next class.
- New Assignment will be posted this weekend.
- Read chapter 6 in eloquent javascript.
- Read more about ES6 from links in slides.