ITMD 465/565
Rich Internet Applications

Lecture 6

Fall 2019 – September 25, 2019

Tonight's Agenda

- Introduction to some ES6 features
- Ajax introduction ?

ES6+ (ES 2015+)

Beyond legacy JavaScript

ES6+

- The 6 edition of the ECMAScript-262 standard for JavaScript
- Also known as ECMAScript 2015
- Was finalized and published in June 2015
- This added significant new syntax and features
- Native support in most browsers at this time for most features
- Can use a tool like Babel to transpile to ES5 for compatibility
- New additions are published yearly. Browser support lags behind slightly.
- ES7 or ECMAScript 2016 is widely supported by current major browsers
- https://en.wikipedia.org/wiki/ECMAScript#6th_Edition_-_ECMAScript_2015
- https://kangax.github.io/compat-table/es6/

ES6 Features

- http://es6-features.org/
- ES6 adds some new syntax and features. Some of the bigger changes are:
  - Classes
  - Block-Scoped Constructs let and const
  - Arrow Functions
  - Default Parameters
  - Rest and Spread Parameters
  - Destructuring Assignment
  - Template Strings
  - Multi-line String
  - Maps & Sets
  - Modules
- http://kangax.github.io/compat-table/es6/

Class

- ES6 added a class syntax similar to other languages.
- Just syntactic sugar, still prototypal under the hood.
- Constructor is used to setup parameters and set properties.
- https://javascript.info/classes

```
class Song {
  constructor(title, artist, duration) {
    this.title = title;
    this.artist = artist;
    this.duration = duration;
    this.isPlaying = false;
  }

  start() {
     this.isPlaying = true;
  }
}
```

Class

- Classes can have getter and setter properties/methods
- These allow you to add methods that handle class properties when you use dot notation to get or set their value
- https://javascript.info/property-accessors

```
class Song {
  constructor(title) {
    this.title = title;

  }

  get title() {
    return this.title;
  }

  set title(t) {
    this.title = “new title: ” + t;
  }
}
```

```
let s = new Song(‘hello’);
s.title; //returns ‘hello’
s.title = ‘goodbye’;
s.title; //now returns ‘new title: goodbye”
```

Classes

- Classes can have inheritance
- Uses the extends keyword to define a “subclass”
- Still uses prototype inheritance in the background
- https://javascript.info/class-inheritance
-
```
class RockSong extends Song {
  constructor(title, artist, duration, type) {
    super(title, artist, duration);
    this.type = type;
  }

  logType(){
   console.log(this.type);
}
}
```

Block Scope

- Remember that when using the var keyword to declare a variable there is no block scope, only functional scope.
- ES6 adds two new keywords that declare block scoped constructs
  - let
  - const
- let is similar to var but has block scope
- *let is the new var*
- const declares an immutable variable that also has block scope. Once the value is assigned it is fixed and can't be changed.

Arrow Functions

- This is a new syntax to declare a function
- Using an arrow function fixes the problems associated with the **`this`** keyword. The this keyword will have the save value as in the context of the enclosing function. It fixes the problem when creating closures and makes the that = this less necessary.
- If the function executes a single statement it will implicitly return the result of that statement.

Old Style

```
function add (a, b) {
   return a + b
}
```

New Style

```
const add = (a, b) => a + b;
```

Default Parameters

- Default parameters are parameters to a function that are given some default values in the function declaration.
- The value can be changed when calling the function

```
const add = (a = 5, b = 6) => {
    return a + b;
}
```

```
function add(a = 5, b = 6) {
    return a + b;
}
```

Rest

- Rest Parameter
- You can use the rest prefix of … to extract the rest of the args from a function.
- This lets you accept unlimited number of parameters and process them dynamically.
- They come in as an array and only has the arguments that were not provided explicitly.
- function add(a, b, …more) {
  // more variable in this block is an array of as many params that were passed
  }
- add(5, 6, 8, 2, 5, 6);
- Inside the add function more is. [8, 2, 5, 6]

Spread

- Spread Operator
- Similar to rest parameter in look but functions differently
- It spreads out the values of an array to separate elements
- Uses the … before the variable name

```
Var params = [“hello”, true, 7];
var other = [1, 2, …params ];     // results are [1, 2, “hello”, true, 7]

Var val = [1, 2];
function add(a, b) { return a + b }
add(…val)    // outputs 3
```

Destructuring

- Destructuring allows you to extract values from arrays and object
- Uses the array brackets or object curly brackets

```
let arr = [5, 8, 2, 5];
let [x, y, z, w] = arr;
// x is 5, y is 8, z is 2, w is 5
```

```
var o = { p: 42, q: true };
var { p, q } = o;
```

Template and Multi-line String

- Use the back tick (`) to define and wrap the string.
- This is for both template strings and multi-line strings.
- With template strings you can then evaluate variables in the string with ${}
- Multi-Line String
- var text = `This is a mult line string that
  continues on to the next line.`;
- Template String
  var name = 'brian';
  var greeting = `Hello ${name}!`;

ES6

- Additional ES6 Resources to review
- http://es6-features.org/
- https://scotch.io/tutorials/demystifying-es6-classes-and-prototypal-inheritance
- https://github.com/getify/You-Dont-Know-JS
- http://blog.teamtreehouse.com/get-started-ecmascript-6
- https://webapplog.com/es6/
- https://codeburst.io/es6-tutorial-for-beginners-5f3c4e7960be
- https://html5hive.org/es6-and-babel-tutorial/

Assignments

Reading/Assignments

- Lab will be posted tonight.
- Read chapter 6 in eloquent javascript.
- Read more about ES6 from links in slides.
- Read tyler mcginnis article on execution contexts, hoisting, scopes, and closures
- https://tylermcginnis.com/ultimate-guide-to-execution-contexts-hoisting-scopes-and-closures-in-javascript/