JS Advanced Functions

Higher-Order Functions

What are H.O.Fs?

- A higher order function is a function that operates on other functions
    - Either by receiving it as a parameter
    - Or by returning a function

What would you use them?

- Creating utility functions
- Leads to D.R.Y code (Don't Repeat Yourself)
- Creates more **declarative programming**
    - You describe patterns
- The opposite is **imperative programming**
    - You describe every single step
- Leads to more maintainable, readable and composable code
- Very common for libraries (like Lodash)

Functions as Input

```
function myFunction(cb) {
  cb();
}

function log() {
  console.log("Hello");
}

myFunction(log);
```

Functions as Input

```
function myFunction(cb) {
  cb();
}

myFunction(function() {
  console.log("Hello");
});
```

Functions as Input

```
function regularlyCalled() {
  console.log("Named function");
}

setInterval(regularlyCalled, 1000);

setTimeout(function () {
  console.log("Anonymous Function");
}, 1000);
```

Functions as Input

```
function repeatLog(num) {
  for (var i = 0; i < num; i += 1) {
    console.log(i);
  }
}

repeatLog(10);

repeatLog(4);
```

Functions as Input

```
function forEach(arr, callback) {
  for (var i = 0; i < arr.length; i += 1) {
    callback( arr[i], i );
  }
}

function handler(item, index) {
  console.log(item, index);
}

var data = ["one", "two", "three"];

forEach(data, handler);
```

Functions as Input

```
function forEach(arr, callback) {
  for (var i = 0; i < arr.length; i += 1) {
    callback( arr[i], i );
  }
}

forEach(["one", "two", "three"], function (item, index) {
  console.log(item, index);
});
```

Functions as Output

```
function creator() {
  return function () {
    console.log("Returned function");
  }
}

var created = creator();
created();
```

Functions as Output

```
function createGreeting(start) {
  return function(name) {
    console.log(start + ", " + name);
  }
}

var hi = createGreeting("Hi");
hi("Jane");

var hello = createGreeting("Hello");
hello("Jeff");
```

Functions as Output

```
function makeAdder(x) {
  return function (y) {
    return x + y;
  }
}

var addTen = makeAdder(10);

console.log( addTen(25) );
console.log( addTen(116) );
```

In-class Exercise / Homework

- Make a `repeat` function that runs any function
- Create an `unless` statement
- Create a `filter` function for arrays
- Create a `reduce` function for arrays
- Create a `map` function for arrays
- Create a `findIndex` function for arrays
- Create anything else you see here