FEWD - JAVASCRIPT

KARA YU

MIT, Meteor

DOWNLOAD BOOTSTRAP

http://getbootstrap.com/getting-started/ Scroll down, click on "Download Bootstrap"

AGENDA

- Review
- JavaScript Object Literals
- Array exercises
- Image carousel
- Card matching game (if we have time)

JS FUNCTIONS

Named function without arguments

```
function hello() {
    console.log('world');
}

hello();
```

Named function with one argument

```
function sayHi(person) {
    console.log('hello ' + person);
}

sayHi("bob");
```

For both of these, what's the environment?

JS FUNCTIONS

Unnamed function assigned to variable

```
var anon = function() {
    console.log('I am anonymous');
};

anon();
```

Defining and executing anonymous function

```
(function() {
    console.log('I am anonymous');
})()
```

JS FUNCTION ENVIRONMENTS

What is a function's environment? When does it get set?

```
function init() {
    var name = "Mozilla";
    function displayName() {
        alert (name);
    }
    displayName();
}

init();
```

- name is a local variable created by init
- displayName is a function declared and used inside a function init
- why does the inner function have access to name?
- can other functions use displayName()?

JS

WHY WOULD YOU STORE THE FUNCTION IN A VARIABLE?

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

var add5 = makeAdder(5);
var add10 = makeAdder(10);

console.log(add5(2));
console.log(add10(7));
```

- Same code. Different environments.
- The environment gets created when function is applied

JAVASCRIPT OBJECT LITERALS

- Everything in JavaScript is an object
- What's an object? It's a container for properties
- They're a little like a dictionary
- A JS object literal is a list of pairs of property names and associated values of an object, enclosed in curly braces ({})

JAVASCRIPT OBJECT LITERALS

- How do you create an object literal?
- How do you set properties?

```
var student = {};
student.attendance = "good";
student.grades = "bad"
student.likeCandy = "false";

> student //what does this print?
> delete student.grades;
```

JAVASCRIPT OBJECT LITERALS

- You can also set all the properties at once
- "First_name", "last_name" and "address" are called properties of mickey_mouse
- How do you get Mickey's first name?

```
var mickey_mouse = {
                    "first_name": "Mickey",
                    "last_name": "Mouse",
                    "address": "Disneyland"
                  }

> mickey_mouse.first_name;
> mickey_mouse["first_name"];
```

JAVASCRIPT OBJECT LITERALS

- You can also get multiple levels of nesting

```
var mickey_mouse = {
                    "first_name": "Mickey",
                    "last_name": "Mouse",
                    "address": {
                      "Anaheim": "Disneyland Magical Kingdom",
                      "Paris": "Happiest place on earth",
                      "Tokyo": "kinda outside the city"
                    }
                  }
> mickey_mouse.address.Anaheim;
> mickey_mouse["address"].Anaheim;
> mickey_mouse["address"]["Anaheim"];
```

JAVASCRIPT OBJECT LITERALS

How would you use these? * You can store multiple components in a variable

```
var state = "green";
var message = {
                "failing": "Tests are failing.",
                "passing": "Tests are passing.",
                "refactor": "Time to refactor."
              }
if (state === "red") {
    console.log(message.failing);
} else if (state === "green") {
  console.log(message.passing);
} else { // Time to refactor.
  console.log(message.refactor);
}
```

JAVASCRIPT OBJECT LITERALS

- How can you get all the properties/keys for an object?

```
Object.keys(student);

//what does this do?
for (var key in student) {
   var prop = student[key];
   console.log(prop);
}
```

JAVASCRIPT OBJECT LITERALS

Takeaways:

- Everything in JavaScript is an object (a list of properties and values)
- You can set properties of anything and then retrieve them later
- Object literals are an easy way to store multiple properties in the same variable

- create an index.html file and link it to a main.js file
- create a new object called dog with 4 properties (age, fur, color, height)
- change the fur property of the dog to "fluffy"```
- if the age <10 print "woof" to the console