CS 4220

- Current Trends in Web Design & Development -

Cydney Auman

AGENDA

01 Strings Continued

02 Loops

03 Functions

04 Arrays and Objects

05 Code Demo & Lab!

String Continued

JavaScript provides properties and methods on strings. A comprehensive list can be found at:
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String

PROPERTIES

`length` returns the length of a string as a number

METHODS

`toLowerCase()` converts the string value to all lowercase characters

`toUpperCase()` converts the string value to all uppercase characters

`includes(value)` return a boolean on whether a string may be found within another string

`split(value)` returns an array of strings, by separating the string based on the value passed

While Loop

The word **`while`** is followed by an expression in parentheses and then a statement of code to be executed. The loop keeps entering that statement as long as the expression produces a value of `true`.

```
let number = 1;
while (number < 10) {
  console.log(number);
  number++;
}
```

For Loop

There are three parts to a `for` loop. The first part initializes the loop. The second part is the expression that checks whether the loop must continue. The final part updates the state of the loop after every iteration. In most cases, this is shorter and clearer than a while loop.

```
for (let number = 1; number < 10; number++) {
  console.log(number);
}
```

Functions

Functions are an important part of JavaScript programming. It provides a way to structure our code and reduce repetition.

Functions are created with an expression that starts with the keyword **`function`**. Functions have an optional set of arguments and a body, which contains the statements that are to be executed when the function is called.

```
function add(a, b) {
  return a + b;
};
add(2, 3);
```

Functions Continued...

In JavaScript functions can be created in several ways. Each of these ways will be demonstrated in the corresponding `examples.js` file.

- Declaration notation
- Functions as Values
- Arrow Functions

**There's no deep reason to have both arrow functions and function expressions in the language. Essentially, they do the same thing. Arrow functions were added in ES6, mostly to make it possible to write small function expressions in a less verbose way. We'll be using them a lot throughout this course.*

Arrays

JavaScript provides a data type specifically for storing sequences of values. An **array** is written as a list of values between square brackets, separated by commas.

```
const alpha = ['a', 'b', 'c', 'd'];
```

Elements in the array can be accessed by index. The first index of an array is zero. So the first element is retrieved with `alpha[0]` which return the value 'a'.

Javascript arrays also have a length property. This tells us how many elements it has inside the array. `alpha.length` will return 4.

**The contents of an array in JavaScript can be almost anything and in fact JavaScript arrays can have mixed data types.*

Basic Array Methods

JavaScript provides many methods on arrays. A comprehensive list can be found at:
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array#

`push(`*`value`*`)` method adds one or more elements to the end of an array and returns the new length of the array.

`pop()` method removes the last element from an array and returns that element. This method changes the length of the array.

`unshift(`*`value`*`)` method adds one or more elements to the beginning of an array and returns the new length of the array.

`shift()` method removes the first element from an array and returns that element. This method changes the length of the array.

Objects

An **object** is a collection of related data consisting of key/value pairs. These key/value pairs can be we added or removed anytime. One way to create an object is by using a curly brace notation. An object like this is referred to as an object literal — we are writing out the object contents as we create it.

```
const course = {
    department: 'Computer Science'
    number: 4220,
    recommendedPrereqs: ['CS1220', 'CS2011', 'CS3220'],
    description: function() {
        console.log(this.department + ' ' + this.number + ' ' + 'covers topics in Web Design & Development.');
    }
}
```

**The value of an object member can be almost anything - in the example above this object has a string, a number, an arrays, and a functions.*

Objects Continued..

Accessing Properties

To access properties from an object use the object name followed by a dot and then by the key name. Alternatively, you can use bracket notation which can be used with a string or variable.

```
console.log(course.department)     // 'Computer Science'
console.log(course['number'])     // 4220
```

Adding Properties

To add properties to an object the format is generally the object name followed by a dot and then by the key name.

```
course.room = 'SH 343'
course['startTime'] = '6:00pm'
```

Removing Properties

To remove properties from an object use the keyword **delete** followed by the object name dot key.

```
delete course.startTime
```

Objects Iterating

A **`for...in`** loop iterates over the properties of an object in an arbitrary order. Because order is not guaranteed, then it is best not to add, modify, or remove properties from the object during iteration. Additionally, `for...in` loops should not be used with Arrays anytime index order is important.

```
for (const key in course) {
   console.log(key);
   console.log(course[key]);
}
```

```
console.log('Week 02');
console.log('Code Examples');
```

Lab, Homework and Prep

Lab

- Complete Q1 - Q3
- Submit to CSNS by Monday 2/03 at 9:00pm

HW

- Complete Q4 - Q8
- Submit to CSNS by Monday 2/10 at 11:59pm

Preparation for Next Week

- Read Eloquent Javascript Chapters 5 & 6