

CSIS 3380  
Advanced Web Programming with  
JavaScript & AJAX

Data Types and Array  
Week 3

# Review of previous lesson

---

- Variables
- Functions
- Conditionals
- Comparision Operators

# Data Types in Javascript

---

- Boolean
  - true and false
- Number
  - 1
  - 42
- String
  - 'foobar'
- Object
  - A collection of properties
  - properties are written as name:value pairs, separated by commas
  - Example: { firstName: 'Edmund', lastName: 'Leng', age: 42 }

# Dynamic typing

---

- JavaScript is a loosely typed or a dynamic language
- Variables in JavaScript are not directly associated with any particular value type, and any variable can be assigned (and re-assigned) values of all types

```
var foo = 42;    // foo is a Number  
var foo = 'bar'; // foo is a String  
var foo = true;  // foo is a Boolean
```

# Object

---

- Software objects are conceptually similar to real-world objects
- Objects have *properties* (states) and *methods* (behaviour)
- Example: Dogs have properties (name, color, breed, hungry) and methods (barking, fetching, wagging tail)
- In Javascript, an object is a container for a list named values, called properties and methods
- The properties are **attributes** of the object
- The methods are **actions** that can be performed on the object
- Methods are stored in properties as **function definitions**.

# Object Literals

- Examples of object

```
var studentA = { id: 1, name:'John Doe', age:22 };
```

```
var studentB = {  
  id: 2,  
  name: 'John Smith',  
  age: 25,  
  address: '555 1st Ave Burnaby BC',  
  attendedCourses: function() {  
    return ['CSIS3380', 'CSIS3381'];  
  }  
};
```

*Note how methods are function definitions*

# Accessing the Object properties and Methods

---

- You can access object properties with the below syntax:
  - *objectName.propertyName*
- *Example:*
  - studentA.name
  - studentB.age
- You access an object method with the below syntax:
  - *objectName.methodName()*
- *Example:*
  - student.attendedCourses()

# this in javascript

- The **this** keyword in javascript refers to the object that “owns” the function

- Example:

```
var studentB = {  
  id: 2,  
  name: 'John Smith',  
  age: 25,  
  address: '555 1st Ave Burnaby BC',  
  description: function() {  
    return this.name+' lives at '+this.address;  
  },  
  attendedCourses: function() {  
    return ['CSIS3380', 'CSIS3381'];  
  }  
};
```

Question: What is the returned value when we invoke **studentB.description()** ?



# Useful Number methods

---

- toString()
  - returns a number as a string
  - Example:

```
var x = 123;  
x.toString(); // display a String representation of the Number
```

- Number.parseInt()
  - Parses a string and returns a whole number
  - Example:

```
Number.parseInt("10"); // returns 10  
Number.parseInt("10.33"); // returns 10
```

# Useful String methods

Methods	Details
indexOf()	returns the position of the <b>first</b> occurrence of a specified text in a string
slice(start, end)	extracts a part of a string and returns the extracted part in a new string
toUpperCase()	Converts String to upper case
toLowerCase()	Converts String to lower case
split()	Converts a String into an array

```
'Blue Whale'.indexOf('Whale');           // returns 5
'The morning is upon us'.slice(4, 11)      // returns "morning"
'hello'.toUpperCase()                     // returns "HELLO"
'a,b,c,d,e'.split(',')                    // returns [ 'a', 'b', 'c', 'd', 'e' ]
```

# Array

---

- a neat way of storing a list of data items under a single variable name
- Each element referenced by a number
  - Start at “zeroth element”: 10 element array has elements: 0,1,2 ,...,8,9
- Accessing a specific element
  - `arrayName[index]`
- The length property of the array tells you how many elements there are in an array

# Example of an array

Name of array →

c[ 0 ]	-45
c[ 1 ]	6
c[ 2 ]	0
c[ 3 ]	72
c[ 4 ]	1543
c[ 5 ]	-89
c[ 6 ]	0
c[ 7 ]	62
c[ 8 ]	-3
c[ 9 ]	1
c[ 10 ]	6453
c[ 11 ]	78

Position number (index or subscript) of the element within array C

A 12-element array.

# Array

---

- Creating an array

```
var shopping = ['bread', 'milk', 'cheese', 'hummus', 'noodles'];
```

- Assessing and modifying the array elements

```
shopping[0];    // returns "bread"

shopping[0] = 'tahini';
shopping;       // shopping will now return [ "tahini", "milk", "cheese", "hummus", "noodles" ]
```

- Getting length of array

```
shopping.length; // should return 5
```

# Useful array methods???

Methods	Details
join(separator)	Converts array into a string
pop()	Removes last element
push()	Add new element at the end
shift()	Removes the first element
unshift()	Adds a new element at the beginning
reverse()	Reverse the elements in the array

# Examples

---

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits.pop();           // Removes the last element ("Mango") from fruits  
fruits.push("Kiwi");    // Adds a new element ("Kiwi") to fruits  
fruits.shift();         // Removes the first element "Banana" from fruits  
fruits.unshift("Lemon"); // Adds a new element "Lemon" to fruits
```

# A more complete list of Javascript methods

---

- [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Number](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Number)
- [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/String](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String)
- [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Array](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array)



# Lab

- Try out the exercise at [https://www.w3schools.com/js/js\\_array\\_methods.asp](https://www.w3schools.com/js/js_array_methods.asp)
- We have a cost in the form "\$120". That is: the dollar sign goes first, and then the number. Create a function **extractCurrencyValue(str)** that would extract the numeric value from such string and return it.
- Example: 

```
extractCurrencyValue('$120') // returns 120
```
- Write a function **ucFirst(str)** that returns the string str with the uppercased first character
- Example: 

```
ucFirst('john') // returns "John"
```

# Lab



- Write the function **camelize(str)** that changes dash-separated words like “my-short-string” into camel-cased “myShortString”.
- That is: removes all dashes, each word after dash becomes uppercased.
- *Hint: use split to split the string into an array, transform it and join back*
- Example:

```
camelize('list-style-image');    // returns 'listStyleImage'  
camelize('-webkit-transition');  // returns 'WebkitTransition'
```

# Lab

- You have an array of user objects, each one has user.name. Write a function **getName(users)** that converts it into an array of names.
- Example:

```
var john = { name: 'John', age: 25 };  
var pete = { name: 'Pete', age: 30 };  
var mary = { name: 'Mary', age: 28 };  
  
var users = [ john, pete, mary ];  
  
var names = getName(users);  
console.log(names)           // returns [ 'John', 'Pete', 'Mary' ]
```