

# CMPS 2680 Homework 9

Spring 2024

## Javascript Objects

### [JS Objects](#)

- A javascript object is a variable that can contain multiple values AND functions.
- A full javascript can be defined and initialized all at once using curly braces like this:

```
let box = {
  width: 5,
  height: 10,
  getArea: function() {
    return this.width * this.height;
  }
}
```

Notice, the keyword **this** is used to access data members that are inside the same object.

### [Object Constructors](#)

- An object constructor is a special function that creates objects. This can be useful if you intend to create multiple objects with the same attributes.
- An object constructor is usually capitalized, and is used to set the properties/methods of the object:

```
function Rectangle(width, height) {
  this.width = width;
  this.height = height;
  this.getArea = function() {
    return this.width * this.height;
  }
}
```

- Then, you can use this function to create objects using the NEW operator:

```
let box1 = new Rectangle(5, 10);
let box2 = new Rectangle(6.35, 12.75);
```

### [Object Properties](#)

- Variables that are associated with an object are called properties
- Object properties can be accessed using dot notation, or associative array notation.
- Using associative array notation allows you to access properties using a variable as a key.

### [Object Methods](#)

- Functions that are associated with an object are called methods
- Object methods can be called using dot notation

Due to the loosely typed nature of Javascript, properties and methods can be added to an object at any time, even after the object has been declared/initialized.

## Forms

- [HTML Forms](#)
- [Form Attributes](#)
- [Form Elements](#)
- [Input Types](#)
- [Input Attributes](#)
- [Form Method Attribute](#)
- [Form Action Attribute](#)

## Processing URL Variables

- Variables can be defined and passed to a web page using key/value pairs at the end of the URL.
- This is the default behavior for a form when using the **GET** method.
- For example, in the following URL v is actually a variable with the value dQw4w9WgXcQ

```
https://www.youtube.com/watch?v=dQw4w9WgXcQ
```

- The first variable is preceded by a question mark. Additional variables can be added with an ampersand. For example, to pass the variables firstName and lastName:

```
https://www.example.com/index.html?firstName=Paul&lastName=Royer
```

- This string of text (beginning with the ?) is known as the **query string**.

### [window.location.search](#)

- This javascript property contains the full query string, INCLUDING the question mark.
- This property can be used for both getting OR setting the query string

### [URLSearchParams](#)

- URLSearchParams is a special javascript object that can be used to access the variables in a query string. Create a URLSearchParams object using its built-in constructor like this:

```
const queryString = window.location.search;  
const urlParams = new URLSearchParams(queryString);
```

- Once you have initialized the object, it has many useful functions:

[get](#), [getAll](#), [has](#), [keys](#), [values](#), [entries](#), [toString](#)

## Full Example

To process the URL

```
https://www.example.com/index.html?firstName=Paul&lastName=Royer
```

You can do the following:

```
const queryString = window.location.search;  
const urlParams = new URLSearchParams(queryString);  
  
let firstName = urlParams.get("firstName");  
let lastName = urlParams.get("lastName");
```