



# MIS3690 WEB TECHNOLOGIES

**BABSON COLLEGE**  
**TOIM DIVISION**



# INTRODUCTION TO JAVASCRIPT



# WHAT IS JAVASCRIPT?

- A programming language
- It is interpreted by Web browsers and servers
  - Just like HTML and CSS
    - HTML – identifies the content (elements) of a web document for so that the browser knows how to present it on the webpage.
    - CSS – defines how each element is to be formatted/displayed in the document/page.
- JavaScript creates interactive or dynamic web pages
  - Can change content and/or formatting as the user is browsing a page
- With JavaScript, a web page can
  - Respond to user events
  - Mouse-click, Mouse-over, mouse-out, enter/exit fields
  - Validate data entry in forms
  - Create custom HTML code and pages on-the-fly

# HOW TO USE JAVASCRIPT?

- Three key concepts for using JavaScript:
  - **Element** – what is the web page element that the user will interact with to start a JavaScript function
  - **Event** – what is the event, associated with the element, that will "trigger" the JavaScript function
  - **Action** – what must the JavaScript function do?
  - All three must be defined for the JavaScript function to work.
    - THIS IS HOW THE **ACTION** IS LINKED TO THE **EVENT** and **ELEMENT**.

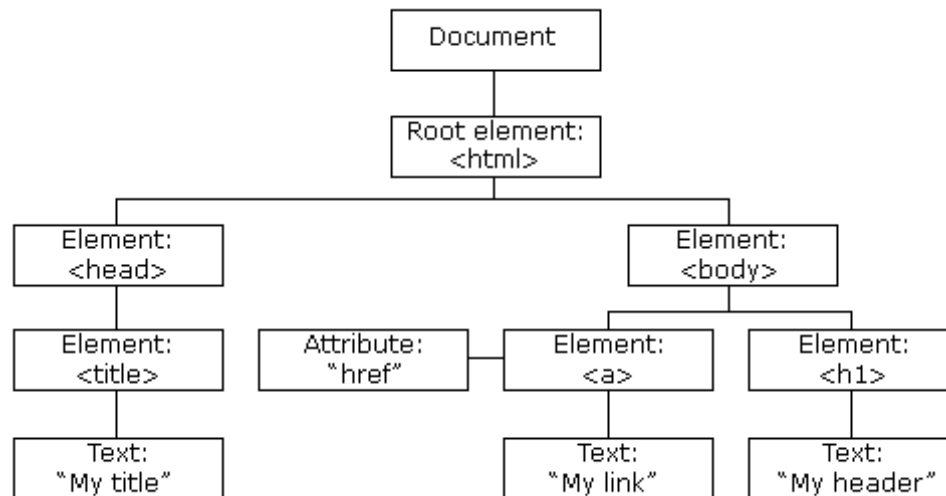
# JAVASCRIPT EXAMPLES

- Please refer to the file:
  - `CSII-InClass-demo.htm`
- What is the event?
- What are the elements?
- What are the actions?
- JavaScript uses the concept of "functions" to define the actions.
- Note the parenthesis – open and close – right next to the name of the function.
  - These are mandatory!

# JAVASCRIPT – SOME BASIC CONCEPTS

## ■ Manipulating Web page elements

- We need to understand how the elements are organized so that we can know WHAT to manipulate to achieve our end.
- Called **DOM** – Document Object Model



# OBJECT ORIENTED LANGUAGE

- JavaScript is Object Oriented
  - We need to understand how JavaScript identifies the elements so that we know HOW to write the JavaScript statements
- The HTML DOM is a standard **object** model and **programming interface** for HTML. It defines:
  - The HTML elements as **objects**
  - The **properties** of all HTML elements
  - The **methods** to access all HTML elements
  - The **events** for all HTML elements

# MANIPULATING WEB PAGE ELEMENTS

- We first need to get the "strings" to the element (like a puppet):

- `document.getElementById(x)`

- A pre-defined function that gets the strings to the Web page element whose id is x

- Example:

- ```
document.getElementById("title")
```

- Save that in a variable.

- Example:

- ```
let myTitle=document.getElementById("title");
```

- Manipulate the Web page element.

- Example:

- ```
myTitle.style.color="red";
```



# JAVASCRIPT IS "OBJECT ORIENTED"

- Treats everything as an object
  - Every web page element is considered an object
  - Uses the DOT Notation
- Examples of JavaScript Object Types
  - Window (the outside-most element of a web page)
  - Document (the web page you create)
  - <h> tag (an element inside the document)
  - <hr/>, <br/> (other elements inside the document)
  - Table (element inside the document)
    - <tr>, <td> (elements inside the table inside the document)
  - Form (element inside the document)
    - <input>, <button>, <textarea>... (elements inside form inside document)

# OBJECT TERMINOLOGY

- Type of object is called an "object class"
  - Form
  - h1 heading
- Specific object is called an "object instance" or sometimes just an "object"
- Every object should have an id that we define
  - `<form id="form1" action="" method="post">`
  - `<h1 id="top">`

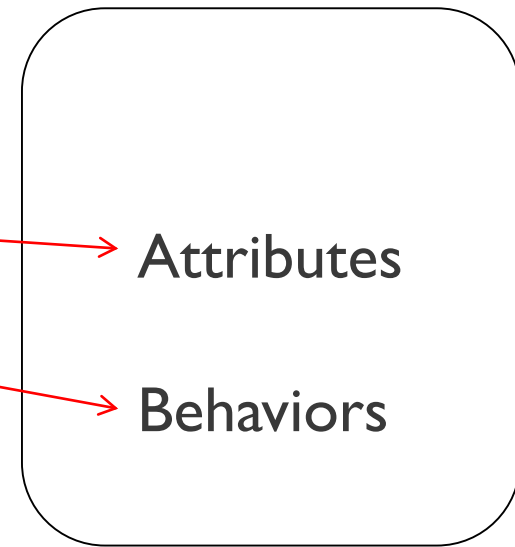
# CLASSES AND OBJECTS

## ■ Class

- is a blueprint for objects of a particular type
- Defines the structure (number, types) of the **attributes**
- Defines available **behaviors** of its objects

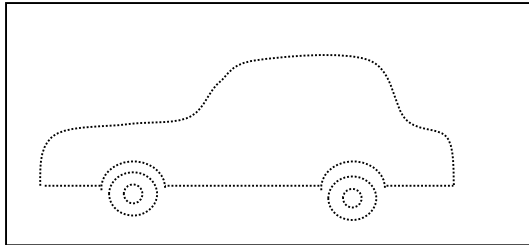
## ■ Object

- is an instance of a class



# CLASS VS. OBJECT EXAMPLE

## ■ Class: Car



## ■ Attributes:

- owner
- color
- amountOfGas
- is4Wd

## ■ Behaviors

- start engine
- refuel

## ■ Object: myOldHonda



## ■ Attributes:

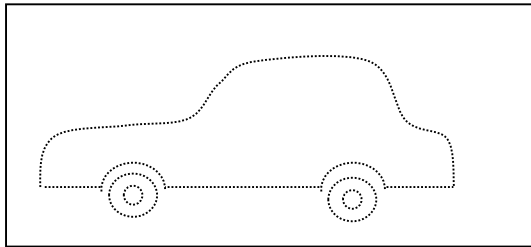
- owner = "Zhi "
- color = "blue"
- amountOfGas = 16
- is4Wd = false

## ■ Behaviors



# CLASS VS. OBJECT EXAMPLE

## ■ Class: Car



## ■ Attributes:

- owner
- color
- amountOfGas
- is4Wd

## ■ Behaviors

- start engine
- refuel

## ■ Object: myNewJeep



## ■ Attributes:

- owner = "Zhi "
- color = "black"
- amountOfGas = 20
- is4Wd = true

## ■ Methods



# OBJECTS HAVE PROPERTIES

- Properties describe the characteristics of an Object
- Dot Notation: `object.property`
  - `document.title` (the title property of a web page doc.)
  - `image.src` (the source property of the image element)
- Different types of objects have different properties

# OBJECTS HAVE METHODS

- Methods are functions that are performed by an object (object class)
  - think of them as verbs
- Dot Notation: `object.method(argument)`
- Examples:
  1. `document.getElementById(x)`
    - "document" is the object
    - "`getElementById(x)`" is the function (or method) that is part of this object.
    - It gets the "puppet strings" to the object whose id is "x".
  2. `window.open()`
    - `window` is the object
    - `open()` is the method – it opens a new window

# REFERENCING OBJECT PROPERTIES AND METHODS

Element Object  
(e.g., `h1` tag)

Element Properties  
(examples)

`id`  
`name`  
`style`

`-color`  
`-fontFamily`  
`-textAlign`

Document Object

Properties (examples):

`title`  
`URL`

Methods examples:

`write ("");`  
`getElementById("id");`



# INTERESTING NOTATION IN JAVASCRIPT

- All CSS styles can be used – must be written differently – for example:
  - `text-align` (CSS) == `textAlign` (JavaScript)
  - `background-color` (CSS) == `backgroundColor` (JavaScript)
  - `font-family` (CSS) == `fontFamily` (JavaScript)
  - `font-weight` (CSS) == `fontWeight` (JavaScript)
- When combining multiple words into one term, JavaScript capitalizes the first letter of each word, starting from the second word.
  - **camelCase**



# JAVASCRIPT FUNCTIONS

- A function is a set of instructions to the browser to do something
  - Typically – associated with an event, an element, and an action.
- Events can be asked to "trigger" functions
- Examples:
  - When user clicks (this is an event) on button (the element), function causes background color to change (the action)
  - When user moves mouse over (another event) an image (the element), function causes image to grow bigger (the action)

# VARIABLE IN JAVASCRIPT

- A variable is a temporary holding place for keeping web page elements, their properties, or values.
- We create variables in JavaScript using
  - `let userChoice;`
    - The `let` statement declares a block scope local variable, optionally initializing it to a value.
  - `const number = 42;`
    - Constants (`const`) are block-scoped, much like variables defined using the `let` statement. The value of a constant cannot change through reassignment, and it can't be redeclared.
- You can name a variable anyway you want – just do not use "reserved" words (e.g., don't name a variable as "form" or "element" or "color")

# JAVASCRIPT VARIABLES

- Variable's value can change, if we use `let`.

- Example:

```
let x;
```

- this just "declares" or creates a variable named "x".

```
x=10;
```

- this assigns a value of 10 to the variable "x".

- Assume there are some JavaScript statements here;

```
x=20;
```

- this assigns a different value of 20 to the same variable "x"

- Variable can be created and given a value in one step.

- Example:

```
let x=10;
```

# OPERATING ON VARIABLES

- + Addition
- - Subtraction
- \* Multiplication
- / Division
- ++ increment by 1
- -- decrement by 1

# JAVASCRIPT CODE

- You must end each JavaScript statement with a semicolon (;) or a new line (or both)
- Each statement is either
  - A JavaScript command (we will learn about these)
  - or a JavaScript function