



# MIS3690 WEB TECHNOLOGIES

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



# PROCESSING FORM INPUTS



# AGENDA

- Quick review of "if" statements
- Processing inputs from forms
- CS15-InClass-1.htm – creating a simple form and processing inputs
- CS15-InClass-2.htm – writing a Celsius-Fahrenheit Converter.

# DIFFERENT WAYS OF WRITING THE IF STATEMENT

## ■ Just the simple IF

```
if (condition)
    { some JS statements }
```

## ■ IF with an alternate set of actions

```
if (condition)
    { some JS statements }
else
    { some other JS statements }
```

# DIFFERENT WAYS OF WRITING THE IF STATEMENT – CONTINUED...

- Extending the IF with many alternate conditions

```
if (condition 1)
  { some JS statements}
else if (condition 2)
  {some other JS statements}
else if (condition 3)
  {some other different JS statements}
else
  {another different JS statements}
```

# YOU CAN COMBINE MULTIPLE CONDITIONS..

- Using **AND** (written as **&&**)

- Using **OR** (written as **||**)

- Example:

```
if ( (userinput < 0 ) || (userinput > 50) )  
    { let the user know he/she entered the wrong value}  
else if ( (userinput >=0) && (userinput < 25) )  
    { do something; // the condition defines the number as being between 0 and 24, both  
    inclusive}  
else // notice how the else has no conditions attached  
    {do something else; // the condition defines the number as between 25 and 50,  
    both inclusive}
```

# COMPLEX BRANCHING (SWITCH)

- Syntax: see ➔
- variable is compared to each case
- Upon match, corresponding JS statements run
- If no match, default statements run
- [https://www.w3schools.com/js/tryit.asp?filename=tryjs\\_switch](https://www.w3schools.com/js/tryit.asp?filename=tryjs_switch)

```
switch (variable) {  
    case value1:  
        JS statements;  
    break;  
    case value2:  
        JS statements;  
    break;  
    case value3:  
        JS statements;  
    break;  
    ...  
  
    default:  
        JS statements;  
}
```

# USING FORMS WITH JAVASCRIPT

- A key use of a scripting language is to validate form-data-entry
- Ensure that the user enters
  - the right data in the right field
  - the right values (if valid ranges are specified)
  - Values in mandatory fields
  - And many more....



# PROCESSING FORM DATA

- We need to assign an ID to the form.
- We can use the form ID to access all of the elements within the form.
- We need to make sure that each of the elements within the form have assigned NAMES

# PROCESSING FORM DATA (TEXTBOX)

- We can get the strings to the form:
- First declare our variable, myForm.
- `myForm = document.getElementById("formid");`
- We can then manipulate the elements and values entered by users
- `myForm.lastname.value` – is the value entered in the field called "lastname" inside this form!

# PROCESSING FORM DATA (TEXTBOX)

- `myForm.lastname.value.length` – gives you the number of characters the user typed into the form field, lastname.
- Can you use this to check if the field is blank?
- `isNaN()` – is a function that checks to see if it is a number (or numerical) value.
  - `isNaN(myForm.age.value)` – will return true if the user entered a non-numerical value and false if the user typed a numerical value.

# PROCESSING FORM DATA - DROPDOWN

```
<select name="state">  
  <option value="MA"> Massachusetts </option>  
  <option value="RI"> Rhode Island </option>  
</select>
```

`myForm.state.value` – gives you the value selected by the user from the drop-down list.

# CSI5-INCLASS-1.HTM

- Use file CSI5-InClass-1.htm provided
  - A form has been built for you.
  - It has 2 simple input options: 2 text boxes, a drop-down list, and a text area.
- Objective: Capture the user inputs from the textboxes and dropdown list and display the values in the text area
- Check to make sure that the user has entered values in the text boxes
- Check to make sure that the age value is a numerical value

# CELSIUS (C) – FAHRENHEIT (F) CONVERTER – CSI5-INCLASS-2.HTM

- Let us create a form with two text fields - one F and the other C
- Add a button for CONVERT and one for CLEAR or RESET
- Let us write a function that will convert one from the other, both ways
- Let us also validate to make sure that the user does not "try anything funny".

# HOW IS THIS DONE – PART 1?

- Once we have the form (there is one available, [CSI5-InClass-2.htm](#))
  - We add an "id" to the form – this will help us manipulate elements within the form – in this case, the two input fields.

```
let myForm = document.getElementById(the form's id)
```

- We should also name each field – this allows us to access / manipulate each using

```
myForm.fieldname
```

- If we name the Celsius input field as "celsius", then it can be accessed by

```
myForm.celsius
```

## HOW IS THIS DONE – PART 2?

- We can access the value entered in each field (e.g. Celsius) using:

```
myForm.celsius.value
```

- We can even get the number of characters entered in each field (e.g., Celsius) using:

```
myForm.celsius.value.length
```

- We need to convert the value entered (a text) into a decimal or float

```
parseFloat() – predefined JavaScript function
```



# HOW IS THIS DONE – PART 3?

- Lastly, we need IF statements to check to see which field the user has entered a value!
- If the user enters a value in Celsius field and nothing in the Fahrenheit field, we should convert Celsius to Fahrenheit
  - $F = C * (9/5) + 32$
- If the user enters a value in Fahrenheit field and nothing in the Celsius field, we should convert Fahrenheit to Celsius
  - $C = (F - 32) * (5/9)$
- If the user does not enter both values, let the user know.
- If the user enters a non-numeric value, let the user know.

# HOW IS THIS DONE – PART 4?

- Let us get to work on the Converter!

