



# MIS3690 WEB TECHNOLOGIES

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



# PROCESSING FORM INPUT – RADIO BUTTON & CHECK BOX



# VALIDATING FORM INPUTS

- Review – from last class
  - How to get values from text-boxes
  - How to put values into text-boxes
  - How to validate blank inputs and non-numerical values
- Getting a value from a drop-down
  - Understanding Array variables
  - We will use these for processing radio-buttons and check-boxes later.

# WHY ARRAY VARIABLE

- A variable can store one value only (at any given time)
- E.g., when you get a value from a text-box in a form
  - `myForm.age.value` – has only one value
- But, a radio button has more than one value
  - The user chooses ONE of these values
  - Hence, a regular variable will not suffice
  - We need a different kind of variable.

# ARRAYS IN JAVASCRIPT

- An array is a variable that can store multiple values

```
var color1 = "red";
```

```
var color2 = array ("red", "blue", "green");
```



- The value of variable color1 is "red"
- The value of variable color2 in index 0 is "red"
- The value of variable color2 in index 1 is "blue"

# HOW IS THIS RELATED TO A FORM?

```
<form action="" method="post">
```

```
  Red: <input type="radio" name="colors" value="red">
```

```
  Blue: <input type="radio" name="colors" value="blue">
```

```
  Yellow: <input type="radio" name="colors" value="yellow">
```

```
  Green: <input type="radio" name="colors" value="green">
```

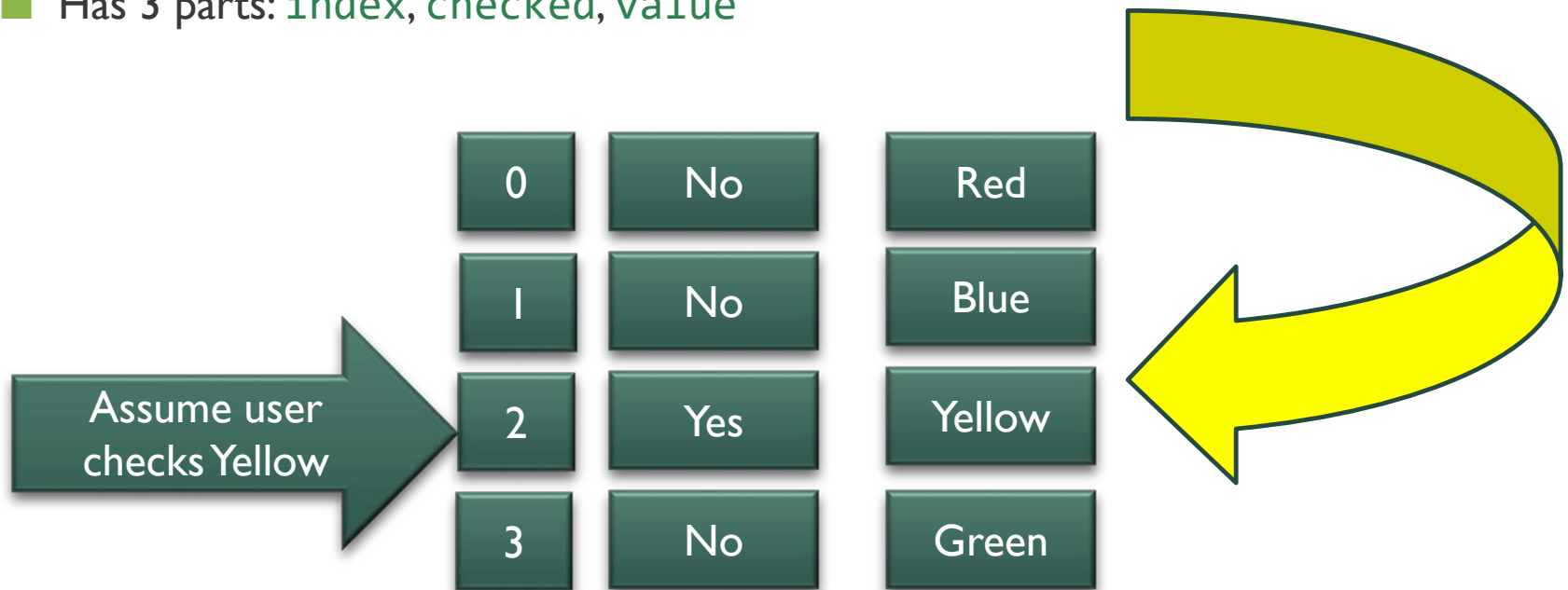
```
</form>
```

**Please select your favorite color:**

**Red:** ☐ **Blue:** ☐ **Yellow:** ☐ **Green:** ☐

# HOW IS THIS RELATED TO A FORM?

- JavaScript creates an array variable to store the form data
- Its name is the same as the name of the radio button (name is "colors")
- Has 3 parts: `index`, `checked`, `value`



# HOW DO WE ACCESS WHAT THE USER HAS CHOSEN?

- if the "checked-part" is false, we don't want that value.
- if the "checked-part" is true, we want that value.
- For each "index", we determine if it is "checked", and if yes, we get its value.
- Since we do the same thing with each "row" of the array variable, we use a "for" loop to make life easier.



# WHY NOT AN "IF" STATEMENT?

```
if (colors[0].checked)
{
    userchoice = colors[0].value;
}
else if (colors[1].checked)
{
    userchoice = colors[1].value;
}
else if (colors[2].checked)
{
    userchoice = colors[2].value;
}
else {
    userchoice = colors[3].value;
}
```

## ■ Why is this a problem:

- We need to know how many button-options are in the form.
- If we add a choice to the button or take one off, we need to change the function.
- We are doing the same exact thing over and over.
- There is a lot of "if"s to write.

# WHY FOR STATEMENT

The starting condition – start with the 0<sup>th</sup> index "i" – a variable we create

```
for (i=0; i<myform.colors.length; ++i)
{
    if (myform.colors[i].checked)
    {
        userchoice = myform.colors[i].value;
    }
}
```

The ending condition – when all the buttons are processed

We create this to store the value the user picks

- `myform.colors.length` will give you the number of radio buttons
- `++i` – what does this do?
- [http://www.w3schools.com/js/js\\_loop\\_for.asp](http://www.w3schools.com/js/js_loop_for.asp)

# CHECK BOX

- Same concept – array variable and for loop
- The user may check more than one box – that is where it is different from a radio button.
- We will learn this through [CSI7-InClass1.htm](#)

# OPENING NEW WINDOWS

- `windows.open()`

- parameters: URL, name, attributes

- `windows.open ('mypage.htm', 'mywin', 'width=300, height=200');`

- Notice the use of single-quotes inside.

- Can specify the "top" and "left" to position the window on the page

- `windows.close()`

# DROP-DOWN MENU VALUE

```
<select name="state">  
  <option value="null"> Select one </option>  
  <option value = "MA"> Massachusetts </option>  
  <option value = "NY"> New York </option>  
  <option value = "VT"> Vermont </option>  
</select>
```

- Assume that the above is defined in a form whose id (that we assign) is "formDemo".

- In our Javascript function:

- `var myForm = document.getElementById("formDemo");`

- The above will get the strings to the form and capture it inside our variable, `myForm` .

- The value the user selects from the drop-down menu can be accessed by:

- `myForm.state.value`



State is the name of the `<select>` element in our form

## DROP-DOWN MENU VALUE - CONTINUED

- We can check the value with "if" statements like,

```
if (myForm.state.value == "null")  
{  
    alert ("Please select a value");  
}
```

- Notice that when we defined our <select> element, we set the first option as "null" to force the user to select an value from the list.

# "ONCHANGE" EVENT

- Used with drop down lists.
- When the user changes the value to select, the event is triggered.