

# CMSC 388A

Forms!

# Agenda

- Forms in HTML
- React Forms
  - Synthetic Events

# Forms in HTML

- Means by which information passes from the user to a server
- Used to read values to be processed by JavaScript
- **<form> tag**
  - Defines the form
  - It has two attributes : **action** and **method**

# Form

- **action** : indicates where the form contents will be sent when the form is submitted
- **method** : defines how the contents will be sent
  - **post** : contents sent using the HTTP POST method. Content is does NOT appear in the request URL
  - **get** : contents sent using the HTTP GET method. Contents becomes part of the request URL

# input Tag

`<input type=" ... " id=" ... " name=" ... " />`

- Appears inside of the `<form>` tag
- Defines several input data alternatives
- The general format is à `<input type="choices" />`

# Type Choices

- text
- password
- checkbox
- radio
- file
- submit
- button
- reset
- hidden
- date
- email

# CheckBoxes

- Allow us to make a selection
- Defined by using type=**“checkbox”**
- We can tell whether an entry is selected by using the **“checked”** property
  - True- an entry has been selected
- Default selection by using checked=**“checked”**

# Form Data Access

- We can access data in forms by using getElementById(string)
- It returns a reference to an element whose id property matches with the given string and we can:
- Retrieve the value of the element (e.g., text field in a form)
  - `var login = document.getElementById("loginId").value;`
- Set a function to call when an element is clicked on (e.g., button)
  - `document.getElementById("processButton").onclick = callback;`
- Another way to associate a function is to use `addEventListener` (allows several events to be added)
  - `document.getElementById("displayValueButton").addEventListener("click", callback);`



# Form Data Access

- Get/Set Attributes of the element

- `var imageElement = document.getElementById("myImage");`
- `var imageName = imageElement.getAttribute("src");`
- `imageElement.setAttribute("src", "imageFile.jpg");`

- Let's complete ex-updateValueInTextField.html

- Animation

- setInterval

- `window.setInterval ( callback, millisecond, param1, param2, ... );`
- The return value from setInterval can be used to stop repeating

# Other Event Handler Attributes

**onclick-** mouse button is pressed and released

**ondblclick-** mouse button is double-click over element

**onmousedown-** mouse is pressed down while cursor is over the element

**onmouseup-** mouse is released while the cursor is over the element

**onmouseenter-** mouse moves onto element

**onmouseover-** mouse pointer enters into an element and its child elements

**onmouseout-** mouse moves off element

**onmousemove-** mouse pointer is moved over an element

# Forms in React

Behavior is very similar as in normal HTML **until** we insert some React goodies and built in behavior!

State and Props together will make the transition easy.

# Events

Two important pieces of information:

- **preventDefault()**, Cancels the event's default behavior (from the browser)
- **target**, Let's you look into the event data

# React Form Events

These are actually called *Synthetic Events*:

- onChange
- onInput
- onInvalid
- onReset
- onSubmit

# React Mouse Events

- onClick
- onContextMenu
- onDoubleClick
- onDrag
- onDragEnd
- onDragEnter
- onDragExit
- onDragLeave
- onDragOver
- onMouseDown
- onMouseEnter
- onMouseLeave
- onMouseMove
- onMouseOut
- onMouseOver
- onMouseUp
- onDragStart
- onDrop

# WTWAW (What To Walk Away With)

1. Create forms in simple HTML
2. Be able to create a react component that reacts to text area change and a button – e.g. exercises 4 & 5!!