

# UNIT : 1

## Forms used by Website

Reference :- HTML, JAVASCRIPT, DHTML AND PHP by IVAN BAYROSS  
Chapter :- 10 : Forms used by Website

# Text Element

- Data Entry fields, accepts single line of text entry
- Properties
  - name
  - value
- Methods
  - focus()
  - blur()
  - select()
- Events
  - onFocus() :
  - onBlur() : The onBlur event occurs when an object loses focus.
  - onSelect()
  - onChange()
- Eg. `<INPUT name="txtEno" type="text" value="01">`
- [FocusBlur](#) [onChange](#) [onSelect](#)

# Password Element

- Unique type of text entry field, all keystroke displayed as “\*”.
- Properties
  - defaultValue
  - Name
  - value
- Methods
  - focus()
  - blur()
  - select()
- Events
  - onFocus()
  - onBlur()
  - onSelect()
  - onChange()
- Eg. `<INPUT name="txtPwd" type="password" value="01">`

[Example](#)

# Button Element

- Used to trigger appropriate form level processing.
- Properties
  - name
  - value
- Methods
  - click()
- Events
  - onClick()
- Eg. `<INPUT name="btnColor" type="button" value="Date">`  
[Example](#) [Bulbonoff](#)

# Submit (Button) Element

- Special purpose button, submits current data to web server for further processing.
- Properties
  - name
  - value
- Methods
  - click()
- Events
  - onClick()
- Eg. `<INPUT name="btnColor" type="submit" value="Submit Data">`

# Reset (Button) Element

- When this button is clicked each data aware form object will be reset to their default value.
- Properties
  - name
  - value
- Methods
  - click()
- Events
  - onClick()
- Eg. `<INPUT name="btnReset" type="reset" value="Reset Data">`

# Checkbox Element

- Behaves as toggle switch. Have two states : checked or unchecked. (T/F or 1/0).
- Properties
  - name
  - Value
  - checked
- Methods
  - click()
- Events
  - onClick()
- Eg. `<INPUT name="chbTest" type="checkbox" value="Yes" checked>`
- [Example](#)

# Radio Element

- Behaves as toggle switch. Have two states : checked or unchecked. (T/F or 1/0).
- When combined in a group only a single radio button can be selected.
- Properties
  - length
  - name
  - value
  - checked
- Methods
  - click()
- Events
  - onClick()
- Eg. `<INPUT name="rbTest" type="radio" value="Yes" checked>`
- [Example](#)



# TextArea Element

- Provides a way to create custom sized, multiple line, text entry object
- Properties
  - name
  - Value
  - placeholder
- Methods
  - Focus()
  - Blur()
  - Select()
- Events
  - onFocus()
  - onBlur()
  - onSelect()

[Example](#)

# Select and Option Element

- Appears as drop-down list or scrollable list of selectable items.
- Properties
  - selectedIndex
  - defaultSelected
  - Index
  - selected
  - text
  - value
- Methods
  - Blur
  - Focus
  - Change
- Events
  - onBlur()
  - onFocus()
  - onChange()      [Example1](#)   [Example2](#)

# Regular Expression

Expression	Meaning
<code>^</code>	Matches beginning of line.
<code>\$</code>	Matches end of line.
<code>[...]</code>	Matches any single character in brackets.
<code>re+</code>	Matches 1 or more of the previous thing

# Validation

- Only Alphabet Validation
- Only digit Validation
- Only Digit and character Validation
  
- Email Validation
- Empty Field Validation
- Fixedlength Validation
- MinMax length validation

## Difference between client-side scripting vs. Server side scripting



Client Side Scripting	Server Side Scripting
The client-side environment used to run scripts is usually a browser.	The <b>server-side environment</b> that runs a scripting language is a web server.
The source code is transferred from the web server to the user's computer over the internet and run directly in the browser.	A user's request is fulfilled by running a script directly on the web server to generate dynamic HTML pages. This HTML is then sent to the client browser.
Advantages to client-side scripting including faster response times, a more interactive application, and less overhead on the web server.	The primary advantage to server-side scripting is the ability to highly customize the response based on the user's requirements, access rights, or queries into data stores.
The Disadvantages of client-side scripting are that scripting languages require more time and effort, while the client's browser must support that scripting language.	The disadvantage of server-side processing is the <u>page postback</u> : it can introduce processing overhead that can decrease performance and force the user to wait for the page to be processed and recreated. Once the page is posted back to the server, the client must wait for the server to process the request and send the page back to the client.
Example <pre>&lt;script&gt; document.getElementById('hello').innerHTML = 'Hello'; &lt;/script&gt;</pre>	Example: <pre>&lt;h1 id="hello"&gt;&lt;?php echo 'Hello'; ?&gt;&lt;/h1&gt;</pre>

**THANK YOU**