

6. JavaScript Forms and Events

Forms and Form Elements

An HTML form contains most commonly used GUI elements including

- Text entry fields
 - For entering names, street address, zip
 - For entering comments
 - Email address
- Password fields
 - For entering password and PIN numbers
- Radio buttons
 - Allowing user pick one (and only one) choice within a group of options
 - Selecting one preferred color option: red, blue, green, etc
 - Example: two-door or four-door car
- Button
- Submit button
- Reset button
- Check boxes
- Pop-up menus
 - Select-and-go menu for long list of choices such as States
- Entry lists

Using form and supported inputs to support web page and web site user interaction activities:

- A user fills inputs in the given form
- Client side form validation (faster and efficient)
 - Verifying passwords
 - Verifying zip codes
 - Validating Email addresses
 - Validating the entered data are in required format
 - Verifying there is no empty field
 - etc
- Click the "Submit" button to send the form to the Web server
- CGI (common gateway interface) program, such as PHP, Perl, Java Code, C++, or C#, etc., resides on server side interprets the data and acts on it
- Process the form data and send back a confirmation note to the user
- Record and save the user entered form data for further processing

Button

`<INPUT TYPE=button>`
Event handler: `onClick`

Examples:

Make a button with a name called "gray", when clicked, it changes the back ground color to white:

```
<form>  
    <input type="Button" value="Gray"  
        onClick="document.bgColor='gray'">  
</form>
```

Checkbox

`<input type =checkbox>`
Event handler: `onClick`

FileUpload

`<input type =file>`
Event handler: `onChange`

Hidden

`<input type =hidden>`
A nonvisual data field
Event handler: `none`

Option

`<option>`
Event handler: `onChange`

Password (for sensitive data)

`< input type =password>`
Event handler: `onChange`

Radio

`< input type =radio>`
Event handler: `onClick`

Reset (a button for resetting the form)

< input type =reset>

Event handler: onClick

Select (a drop-down menu for selecting Option items)

<SELECT [multiple]> .. </SELECT>

Event handler: onChange

Submit (a button for submitting a form)

< input type =submit>

Event handler: onClick, onSubmit

Text (creating a single line of text)

< input type=text>

Event handler: onChange

TextArea (creating a multiline text entry field)

< input type =textarea>

Event handler: onChange

Events and Event Handlers

onAbort

- User aboard the page loading
- Image (JavaScript 1.1)

OnFocus, onBlur

- Text elements
- User left the object

onChange

- Select, text input elements
- User changed the object

onClick

- Button elements, Link
- User clicked the object
- Examples

Create a button, name it SubmitText, display Submit on the top of the button; when clicked, it calls SendText() function:

```
< input type = "Button" name="SubmitText"
value="Submit"          onClick="SendText()" >
```

onDbClick

- Document, Link, Image, Button elements
- User double clicked the object

onError

- Image, Window (1.1)
- The script encounter an error

onKeyDown, onKeyPress, onKeyUp

- Document, Image, Link, text elements
- Key board related

onLoad, onUnload

- The object is loaded; the user left the window
- Window; Image
- Examples
Load and run the timeTest() function when the browser is loaded:
<body bgColor = "white" onload = "timeTest()">
- use onUnload as a garbage collector for closing all windows.

onMouseDown, onMouseUp

- Document, Link, Image, Button elements

onMouseover, onMouseout

- Link; Image and Layer
- The mouse moved over the object; the mouse moved off an object

onReset, onSubmit

- Form reset; form submission
- Triggered onreset; onsubmit event when the button is clicked
- Use onSubmit event handler to validate forms

onSelect - The user selected the contents of an object

onSubmit - The user submit a form

Example 6-1: A simple text field input form.

```

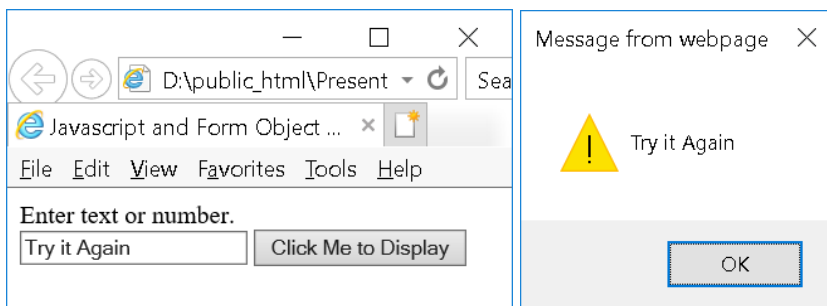
<html>
<!-- form_object.html -->

<head>
<title>Javascript and Form Object Data Type</title>
<body>
Enter text or number.
<form name=Form1>
    <input name=text1 value= ****>
    <script language=JavaScript>
        // Document object has many forms:
        // form[0], form[1], form[2],...
        // - the form[0] is the first form.
        // - the form[0] is also the same object as "Form1"
        //
        // Elements included in the Form1 are:
        // INPUT element0 -- textfield called "textT1"
        // INPUT element1 -- button

        formObj =document.forms[0].elements[0]

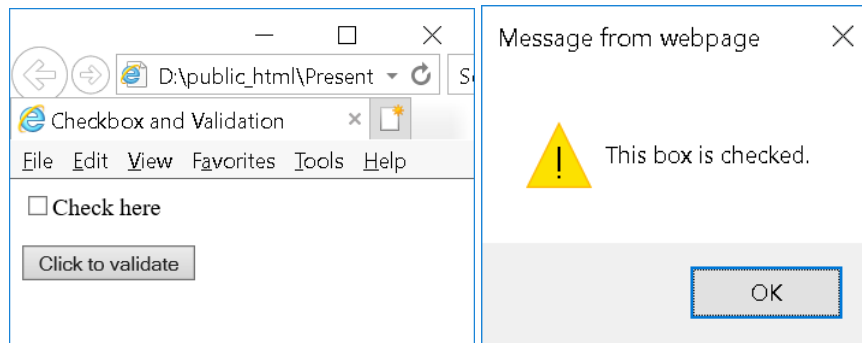
    </script>
<!--
        <INPUT onclick=alert(formObj.value) TYPE = "button" value="Click
Me to Display"
-->
        <input onclick=alert(document.Form1.text1.value) type = "button"
value="Click Me to Display"
</form>
</body>
</html>

```

**Example 6-2:** Create a check box with validation.

```
<html>
<!-- checkbox.html -->
<head>
<title>Checkbox and Validation</title>
<script language="JavaScript">
function checkThisBox(form) {
    if (form.checkbox1.checked) {
        alert("This box is checked.")
    } else {
        alert("The box is not checked.")
    }
}
</script>
</head>

<body>
<form>
<input type="checkbox" name="checkbox1">Check here<P>
<input type="button" name="button1" value="Click to validate"
onClick="checkThisBox(this.form)">
</form>
</body>
</html>
```



Example 6-3: Upload a file to a Web server.

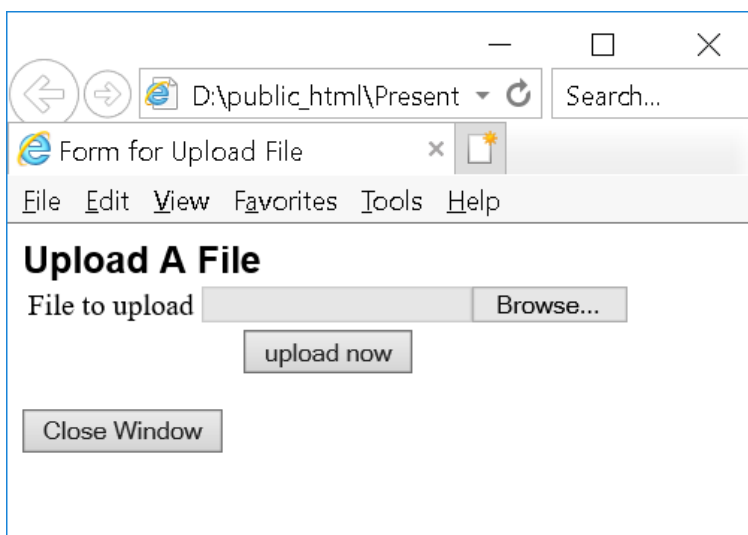
```

<html>
<!-- fileupload.html -->
<head>
<title>Form for Upload File</title>
<script language="JavaScript">

    function closeWindow()
    {
        window.close()
    }

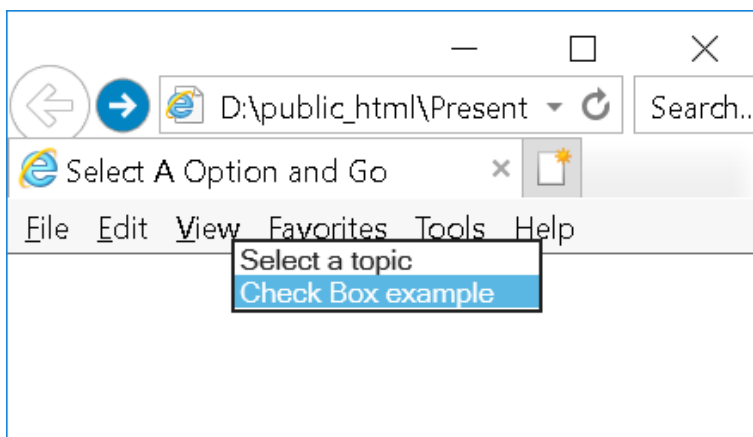
</script>
</head>
<body>
<font face="Arial" size="-1"><h2>Upload A File</h2></font>
<form method="post" action="/cgi-bin/thisUPLOAD.CGI"
enctype="multipart/form-data">
<table>
    <tr>
        <td>File to upload</td>
        <td><input type="file" size="20" name="upfile"></td>
    </tr>
    <tr>
        <td colspan="2" align="center"><input type="submit"
value="upload now"></td>
    </tr>
</table>
</form>
<form><input type="button" value = "Close Window" onClick =
"closeWindow()"></form>
</body>
</html>

```



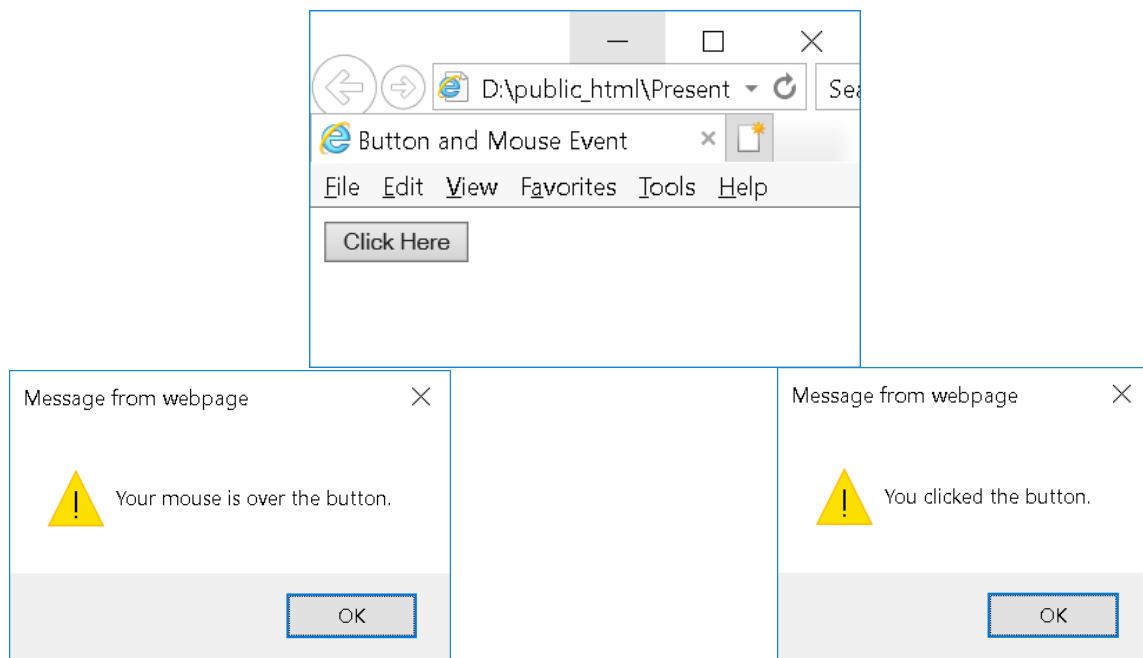
Example 6-4: Use option control and function.

```
<html>
<head>
<!-- option.html -->
<title>Select A Option and Go</TITLE>
<script language=javascript type="text/javascript">
function goPage(newLoc)
{
    newPage = newLoc.options[newLoc.selectedIndex].value
    if (newPage != "")
    {
        window.location.href = newPage
    }
}
</script>
</head>
<body bgcolor=white>
<div align=center>
<!--
<form action="golocation.cgi" method=get>
-->
<form>
<select name="newLocation" onChange="goPage(this.form.newLocation)">
<option value="" selected>Select a topic
<option value="checkbox.html">Check Box example
</select>
<noscript>
<input type=submit value="Go There!">
</noscript>
</form>
</div>
</body>
</html>
```



Example 6-5: An example shows how to tie together the button and events: `onLoad`, `onClick`, and `onMouseOver`. As soon as you move the mouse over on the top of the button, the `alert()` display “Your mouse is over the button”. Therefore, to be able to click the button, we will use “Tab” key to move the Windows focus on the button, then hit “Enter” key to show “You click the button” message.

```
<HTML>
<html>
<!-- button_mouse.html -->
<head>
<title>Button and Mouse Event</title>
</head>
<body onLoad="alert('PAGE loaded!')">
<form>
<input type="button"
  value="Click Here"
  onClick="alert('You clicked the button.')"
  onMouseOver="alert('Your mouse is over the button.')">
</form>
</body>
</html>
```



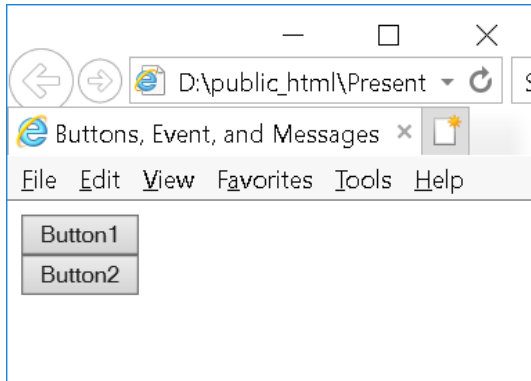
Example 6-6: An example shows how to call functions through the button controls and events.

```
<html>
<!-- butmsgs.html -->
<head>
<title>Buttons, Event, and Messages</title>
<script language="JavaScript">
function changeDsp(newTarget,Text){
    newTarget.value=Text
    alert(Text)
}
</script>
</head>
<body>

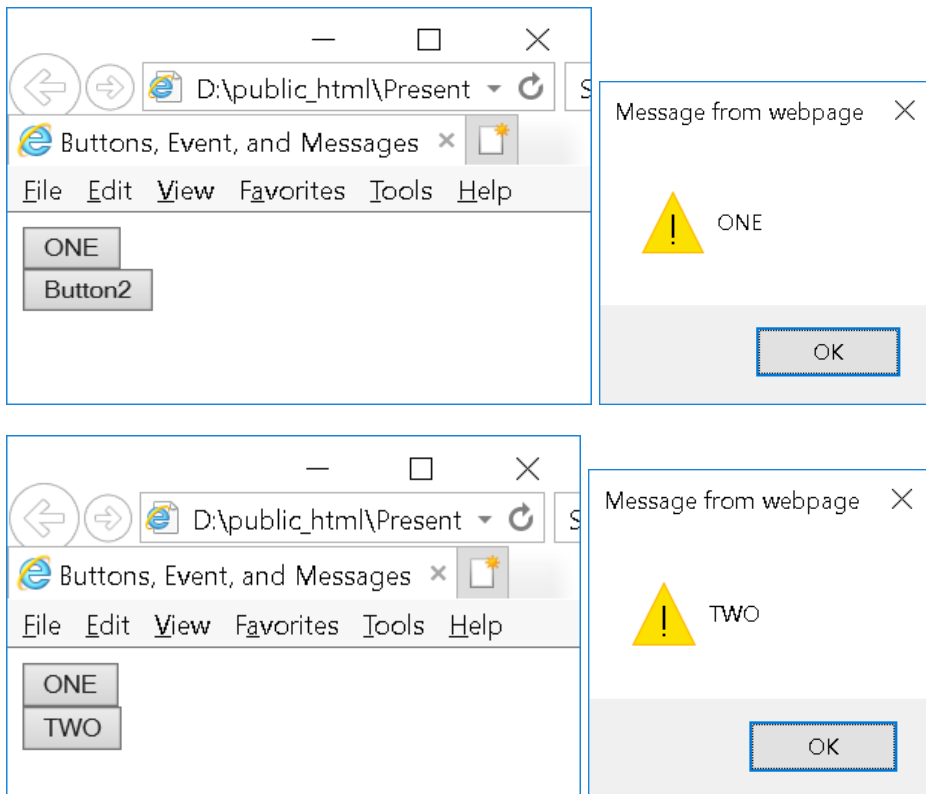
<form>
    <input type="button" value="Button1"
onclick="changeDsp(this, 'ONE') "><br>
    <input type="button" value="Button2"
onclick="changeDsp(this, 'TWO') "><br>
</form>

</body>
</html>
```

Before Click on Button1 and Button2:

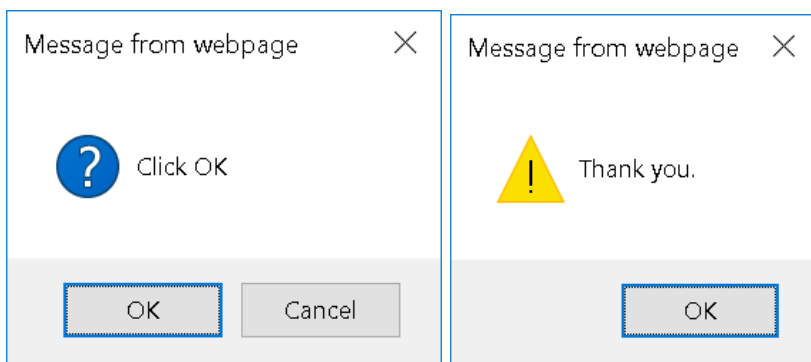
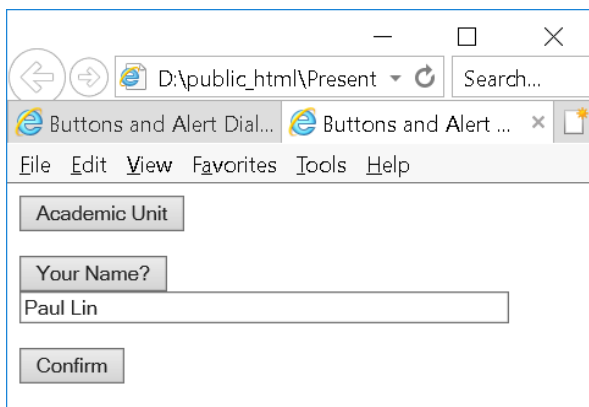
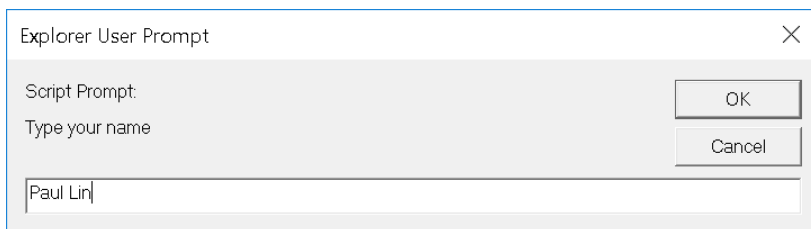
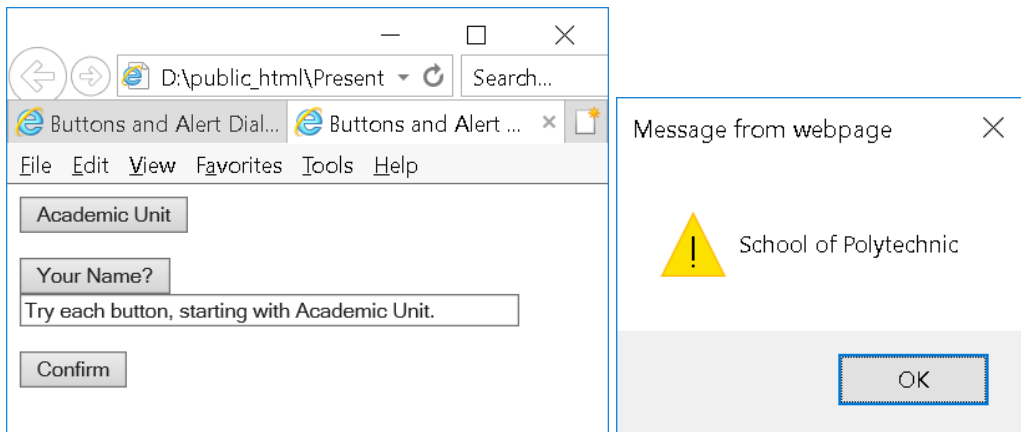


After Click on Button1 and Button2:



Example 6-7: An example that uses input button, text field of a form, window functions `prompt()`, and `confirm()`.

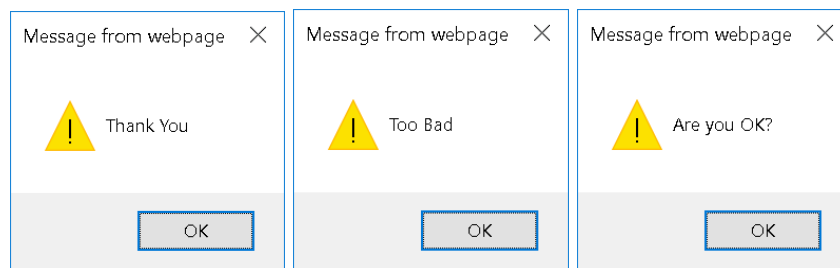
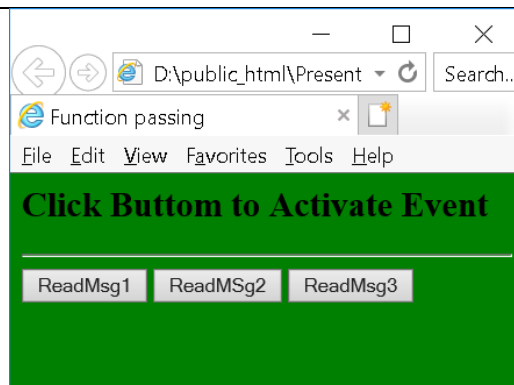
```
<html>
<!-- button_alert.html -->
<head>
<title>Buttons and Alert Dialog Box</title>
</head>
<body>
<form name="new_form">
<input type="button" value="Academic Unit."
  onclick="alert(' School of Polytechnic')">
<p>
<input type="button"
  Value="Your Name?"
  onClick="document.new_form.new_text.value=prompt('Type your
name', 'Here')">
<input type="text" name="new_text"  size="50"
  value="Try each button, starting with Academic Unit.">
<p>
<input type="button" value="Confirm"
  onclick="if(confirm('Click OK')){alert('Thank you.')}
  else{alert('CLICK OK!!!')}">
</form>
</body>
</html>
```



Example 6-8: An example that uses buttons to activate events for reading messages.

```
<html>
<!-- buttonclick.html-->
<head>
  <title>Function passing</title>
  <script language=javascript type="text/javascript">
    <!-- Hide script from older browsers

    function docDisplay(message) {
      alert(message)
    }
    // End hiding script from older browsers -->
  </script>
</head>
<body bgcolor=green>
<h2>Click Button to Activate Event</h2>
<hr>
<form>
  <input type=button value="ReadMsg1" onClick="docDisplay('Thank
You') ">
  <input type=button value="ReadMSg2" onClick="docDisplay('Too
Bad') ">
  <input type=button value="ReadMsg3" onClick="docDisplay('Are you
OK?') ">
</form>
</body>
</html>
```



Example 6-9: This program use JavaScript to verify user input and password.

```
<html>
<!-- passwd.html-->
<head>
  <title>Password Check</title>
  <script language=javascript type="text/javascript">
    <!-- Hide script from older browsers

    function validForm(passForm) {
      if (passForm.passwd1.value == "") {
        alert("You must enter a password")
        passForm.passwd1.focus()
        return false
      }
      if (passForm.passwd1.value != passForm.passwd2.value)
      {
        alert("Entered passwords did not match")
        passForm.passwd1.focus()
        passForm.passwd1.select()
        return false
      }
      return true
    }

    // End hiding script -->
  </script>
</head>
<body bgcolor=white>
<form onSubmit="return validForm(this)" action="someAction.cgi">
  Your name: <input type=text size=30>
  <p>Choose a password: <input type=password name="passwd1">
  <p>Verify password: <input type=password name="passwd2">
  <p><input type=submit value="Submit"> <input type=reset>
</form>
</body>
</html>
```

The screenshot shows a web browser window with the title "Password Check". The address bar displays "D:\public_html\Present". The browser's menu bar includes "File", "Edit", "View", "Favorites", "Tools", and "Help". The main content area has a white background and contains the following form elements:

- A text input field labeled "Your name:".
- A password input field labeled "Choose a password:".
- A password input field labeled "Verify password:".
- Two buttons at the bottom: "Submit" and "Reset".