Lecture 6: JavaScript

Functions, user-defined functions, events, forms

HTML form

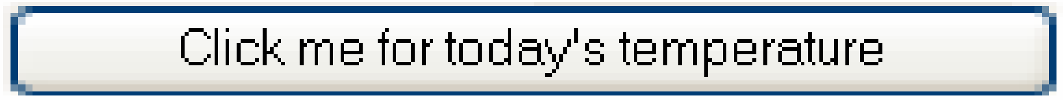
- A **form** is a grouping of buttons and other event-driven elements within a page, delimited by the tags <form name=“FORM_NAME”> … </form>

```
<html>
   <head>
    </head>
    <body>
        <form name="MyForm1">
           <input type="button" value="Click Here" />
           <input type="text" name="temperature" value="110" />
        </form>
    </body>
</html>
```

Button and its event handler

<input type="button" value="BUTTON_LABEL" onClick = "JAVASCRIPT_CODE" />

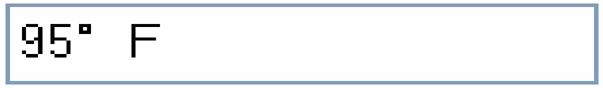
- **type = "button"** specifies to the browser that this is a button element.
- value="BUTTON_LABEL" specifies the label to display on the button face; BUTTON_LABEL should be substituted by any text you what to display on the button face.
- onClick = "JAVASCRIPT_CODE" specifies the JavaScript code to be executed when the button is clicked by a user.
- E.g. <input type="button" value="Click me for today's temperature" onClick = "**alert('93° F ');**"/>



Text object

<INPUT TYPE="text" NAME="*textName*" VALUE="*textValue*“ />

- TYPE = “TEXT” specifies that this is text input element of a form.
- *NAME="textName"* specifies the name of the *Text* object. You can access this text object in JavaScript by
  - Document.formName.textName where formName is the name of a form which contains this text object.
- *VALUE="textValue"* specifies the initial value of the *Text* object. The value of the text object can be changed by user or by javascript. You can access the value of this text object in JavaScript by
  - Document.formName.textName**.value**
- E.g. <INPUT TYPE="text" NAME="temperature" value="95° F" />


95° F

HTML images with JavaScript

<IMG [NAME="*imageName*"] SRC="*Location*" />

- An image is **an object** of document's images in JavaScript.
- *NAME="imageName"* specifies the name of the *Image* object. imageName should be replaced by any name you want to name your image. You can access this image in JavaScript by
  - **docment.images**.imageName
- *SRC="Location"* specifies the URL of the image to be displayed in the document. You can access this value using the *src* property.
  - **docment.images**.imageName.src
- Change the image
  - **document.images**.imageName.src = "imageFilePath"

JavaScript Function Example

```
function FahrToCelsius(tempInFhr)
{
    celius = ( 5 / 9 ) * ( tempInFahr – 32 );
    return celius;
}
```

Modified from Reed’s book.

JavaScript Function

- A JavaScript function is an abstraction of sequence of instructions for a certain task.
- Parameters – Input of the task
- Local variables – Local memory to perform the task
- Return statement – Return the result of task back to the calling program (which will be accepted into the calling program's memory).
- Note: Some functions perform Input/Output tasks; the result of these functions are not only reflected by the return statement and memory, but also reflected in the reality (display in monitor, print in printer, input of keyboard, etc.)

Advantages of using functions

- Help minimize the amount of the detail that the programmers must keep track of. You only need to care about
  - Function name
  - Input of the function
  - Output of the function (result)
- Help minimize the length and complexity of the code.
  - The lengthy sequence of instructions for a task are organized as functions.
  - Every time the task is needed, you just need to call the function without repeating the instructions again and again.

JavaScript predefined functions

- JavaScript’s predefined functions represent a collection of useful, general-purpose abstractions of tasks.
- Input/Output: prompt, confirm, alert
- Math functions: Math.sqrt

Structure of User-defined Functions

```
function FUNCTION_NAME(PARAMETER_1, PARAMETER_2, …,
    PARAMETER_n)
{
   STATEMENT1;
   STATEMENT2;
   ….
   STATEMENTm;
   return VARIABLE_NAME;
}
```

- Note: parameters and return statement are optional; you can define a function without parameters as well as a function without return statement.

Event handling using functions

```
<html>
  <head>
  <script>
  function promptAndCalc()
     {
       tempInFhr = prompt("Please enter temperature in Fahrenheit");
       tempInFhr = parseFloat(tempInFhr);
       celius = (5/9) * (tempInFhr - 32);
       alert("Temperature in celius is " + celius);
     }
  </script>
  </head>
  <body>
       <input type="button"
                  value="Click me for your celius temperature"
                  onClick="promptAndCalc();"/>
   </body>
</html
```

Conditional Execution – if statement

```
if (BOOLEAN_TEST) {
  STATEMENTS_TO_EXECUTE_IF_TRUE
}
else {
  STATEMENTS_TO_EXECUTE_IF_FALSE
}
```

Conditional Execution Example

```
if (currentTemp > 80)
{
   alert(“Hot!”);
}else
{
   alert(“Okay”);
}
```

The Places to Put JavaScript

- Inside <head>…</head>
  - Illustration: <**head**>… <script>…</script>… </**head**>
  - The **functions** and **variables** defined in <script>…</script> inside **head** are guaranteed to be **established** before the execution of any JavaScript codes in <body>…</body>.
  - The Javascript codes inside **head** are guaranteed to be executed before any codes in <body>…</body>.
- Inside <body>…</body>
  - Illustration: <**body**>… <script>…</script>… </**body**>

Summary

- Form and its elements (button, input text)
- Change image dynamically
- Javascript Function and user-defined functions
- Event-handling using user-defined function
- Conditional-execution (won’t be in the exam)