

Laboratory Session #6

LAB GOALS

- Using the <input> tag with type ="file"
- Interacting with DOM
- Event handling

Step 1: Start with a simple HTML web site similar to figure below, just make sure you put in a <script> tag in it:

```
<!DOCTYPE html>
<html>
<head>
<script type="text/javascript">
</script>
</head>
<body>
</body>
</html>
```

Step 2: Place the following documentation lines in the HTML file, right below the <!DOCTYPE html>. Make sure these are placed in the HTML documentation tag.

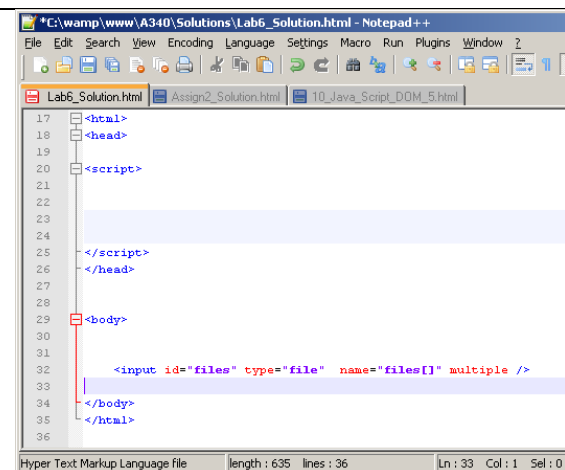
Your name
Web Programming
Lab: 6 (JavaScript, input tag, event handling, DOM manipulation)
Semester, Year

Step 3: Set the title of the web page to "JavaScript Arrays, input tag, event handling". Then set the keywords for the page to JavaScript, Web Programming, DOM Manipulation, and Event Handling. Finally make sure the character set for the web page is set to UTF-8.

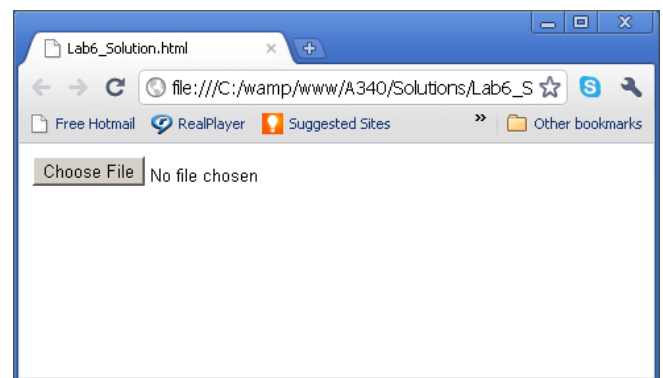
Step 4: At the top of the page, include your name (centered and as H1 header). Following your name, include one or more paragraphs describing what you learned from this lab.

Step 5: Place the following <input> line in the body of your HTML page. Note the following line will display a button which allows the user to choose one or more files from the file dialog box. Note that this tag is somewhat browser dependent. At the time of this writing, IE appears to only allow the user to select one file, and the other major browsers (Chrome, FireFox, etc) allow multiple files to be selected.

<input id="files" type="file" name="files[]" multiple />

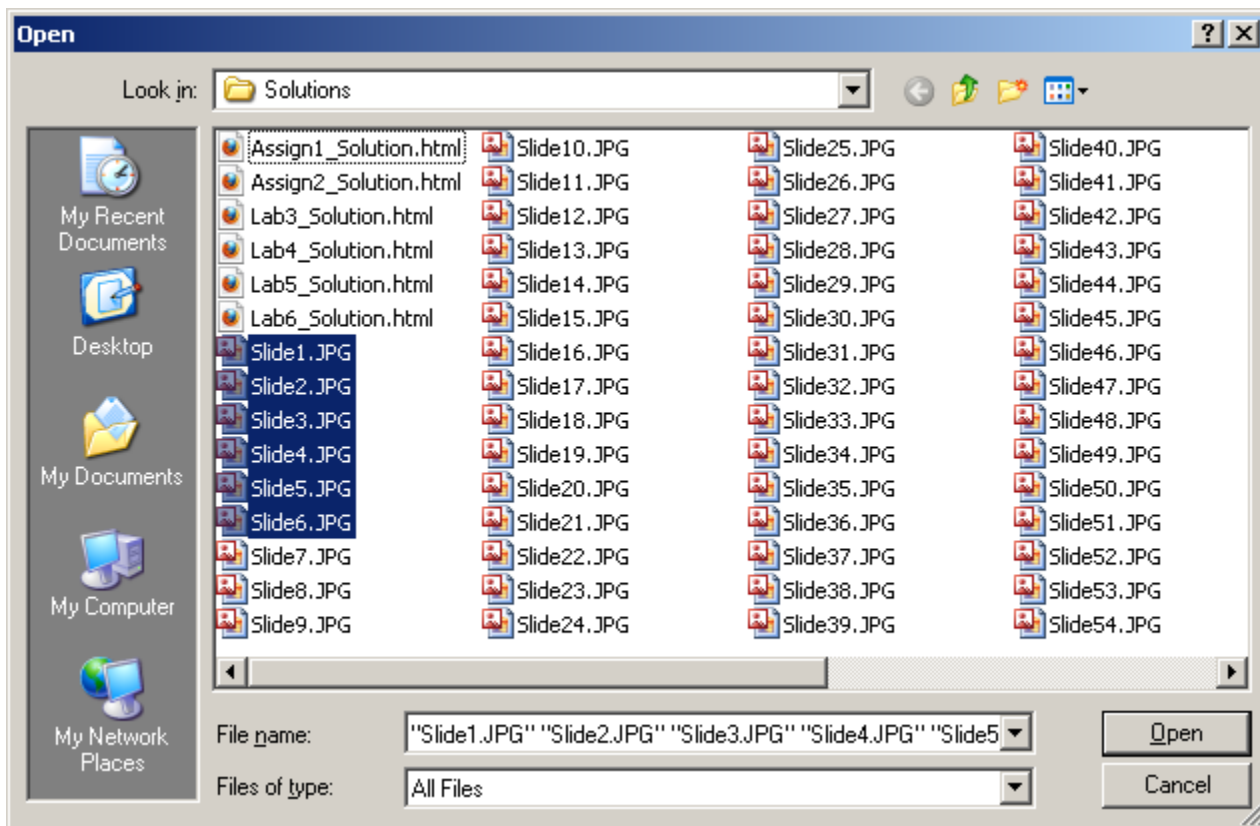


Let's save this file and load it in a Chrome browser.

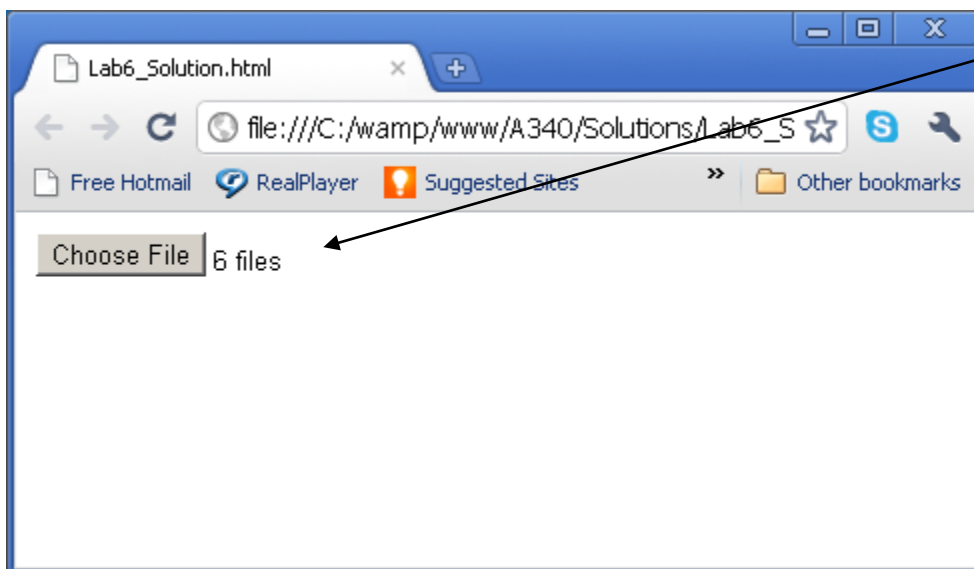


Now, let's click the "Choose File" button.

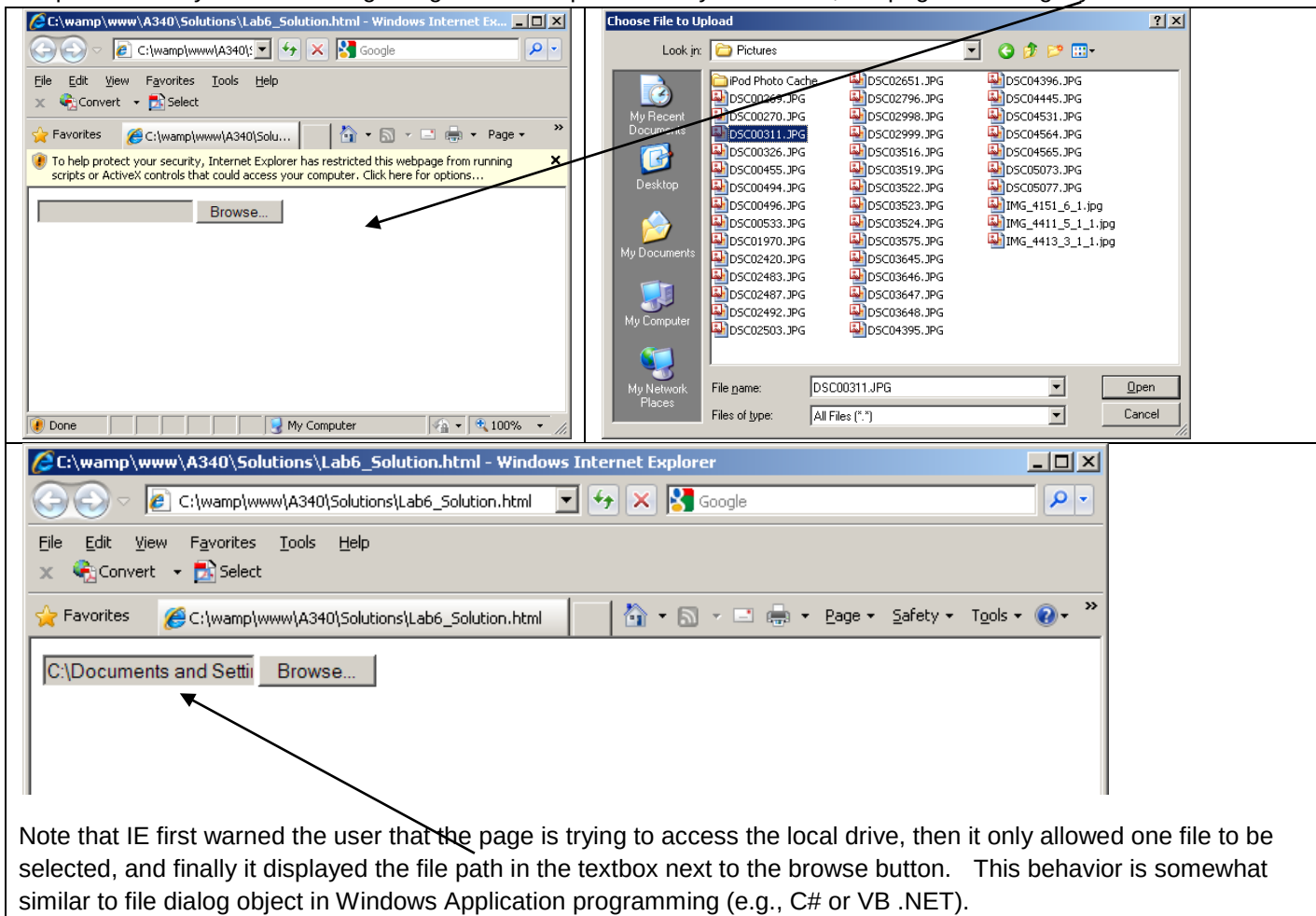
Clicking the “Choose File” button will display the FILE Dialog box and allows the user to select one or more files (depending on the browser). We are using Google Chrome, so let’s select a few files and click OPEN. (see below)



Note that, immediately to the right of the “Choose File” button, we will see how many files have been selected.



Step 6: Let's try the same thing using Internet Explorer. As you can see, the page looks slightly different.



Note that IE first warned the user that the page is trying to access the local drive, then it only allowed one file to be selected, and finally it displayed the file path in the textbox next to the browse button. This behavior is somewhat similar to file dialog object in Windows Application programming (e.g., C# or VB .NET).

Step 7: Examining the file names brought back by the input tag:

In step 5, recall the input tag `<input id="files" type="file" name="files[]" multiple />` Aside from the "type" attribute, it provides the following attributes:

<code>id= "files"</code>	// the id by which this object is identified in the DOM
<code>name = "files[]"</code>	// An array which holds the files selected by the user
<code>multiple</code>	// Allow multiple input to be brought back.

The "id", and the "name" attributes above are going to be useful when we decide to examine what is brought back by the input tag.

Step 8: Let's go back to our HTML file and add a new button as well as an event handler which allows us to extract the information brought back by the <input> tag. Note that at this point the event handler showDirectory() is an empty function.

```

head>
<script>
    function showDirectory()
    {
    }
</script>
</head>

<body>
    <h1> Show the Directory and Select one or more files. </h1>
    <input id="files" type="file" name="files[]" multiple />
    <br />

    <button onclick="showDirectory();"> Show files in the directory </button>
    <br />
</body>

```

Step 9: At this point, we would like to extract the file names that were selected by the <input> tag from the file dialog. To do so, we need to work with the DOM. The following statement will retrieve the <input> object using it's ID.

```
var myFiles = document.getElementById("files");
```

Now that we have the object in our local variable "myFiles", we can extract the individual files and display them. The "length" property of the object, indicates how many files were selected, and the "files[]" property holds the file name, file type and file size. The loop below extracts this information:

```

for (var i=0; i < myFiles.files.length; i++)
{
    document.write("<b>File Name: </b>" + myFiles.files[i].name + " <br />");
    document.write("<b>Type: </b>" + myFiles.files[i].type + " <br />");
    document.write("<b>Size: </b>" + myFiles.files[i].size + " <br />");
    document.write("<p>");
}

```