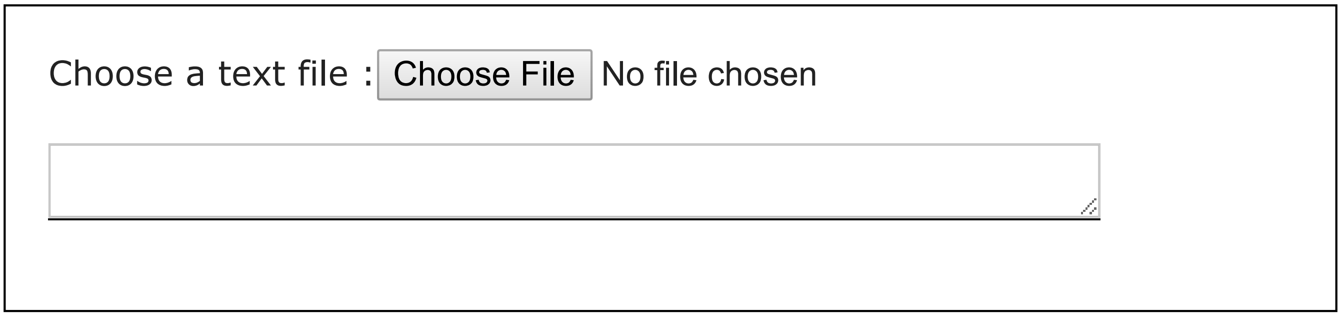
Practical examples: reading file content as text

EXAMPLE 1: READ A SINGLE FILE CONTENT

Example at JS Bin, or try it here in your browser



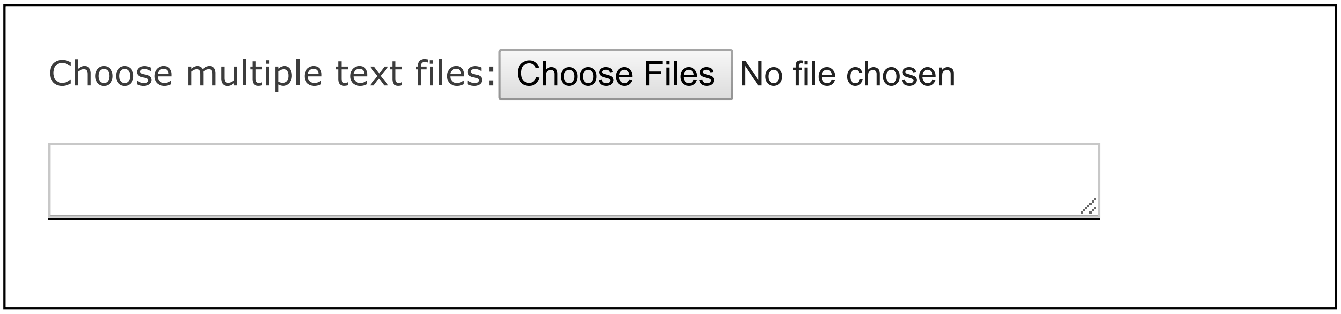
Complete source code:

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="utf-8">
 <title>Example of use of FileReader with a text file</title>
</head>
<body>
<label for="files">Choose a text file:</label><inputtype="file" id="file"
                          onchange="readFileContent(this.files)"/><br/>
<p>
<textarea rows=15 cols=50 id="fileContent"></textarea>
<script>
 function readFileContent(files) {
     console.log("In readFileContent");
     var reader = new FileReader();
   // Executed last: called when the file content is
loaded, e.target.result is
19.     // The content
        reader.onload = function(e) {
            // display content in the textarea with id="fileContent"
            document.getElementById("fileContent").value=e.target.result;
        };
        // Read in the tfile as text
        console.log("Reading file:" + files[0].name);
        // Executed first: start reading asynchronously the file, will call the
    onload
        // callback when the file is read
30.     reader.readAsText(files[0]);
     }
    </script>
    </body>
    </html>
```

This example is the one at the end of the previous page. This time, we show the complete source code above. Remember that the instruction at line 30 is executed first, then when the file is read, the browser will call asynchronously the `onload`callback at line 20.

EXAMPLE 2: A VARIATION OF THE PREVIOUS ONE, USING MULTIPLE FILES

Example on JS Bin, or try it below in your browser. This time, please select multiple text files (using shift for multiple selection):



Choose multiple text files: Choose Files No file chosen

Source code:

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="utf-8">
 <title>Example of use of FileReader with a text file</title>
</head>
<body>
<label for="files">Choose multiple text files:</label>
<input type="file" id="files"
      multipleonchange="readFilesAndDisplayAsText(this.files);"/><br/>
<p>
<textarea rows=30 cols=50 id="filesContent"></textarea>
<script>
 var filesContent =document.getElementById("filesContent");
 function readFilesAndDisplayAsText(files) {
    console.log("dans read files");
    // Loop through the FileList
    for (var i = 0, f; f = files[i]; i++) {
        var reader = new FileReader();
        // Add an onload listener to the reader
        addOnLoadListener(reader, f.name);
        // start reading, will call the listener later, when the file f is read
        reader.readAsText(f);
    }
 }
 function addOnLoadListener(reader, name) {
    // Add an onload listener that will be able to print the name of the
    // file...
    reader.onload = function(e) {
        filesContent.value += "###### READING FILE " +name + "
######";
        filesContent.value += e.target.result;
    };
 }
</script>
</body>
```

```
42.  </html>
```

Explanations:

This example is similar to the previous one except that this time we read multiple files.

- *Line 20:* this is the `for` loop that will iterate on the `files`object passed as parameter by the `onchange` listener declaration at line 10
- *Line 25:* instead of declaring the `onload` listener with a`reader.onload =`... directly in the loop, we preferred this time to write a separate function that will do that. This technique is useful when you want the listener to work with extra variables computed in the loop (in our case the `name` of the file).

SOME COMPLEMENTS ABOUT ENCODING

Note that you can optionally indicate the encoding of the file you are going to read (default is UTF-8):

```
reader.readAsText(file, 'UTF-8');
reader.readAsText(file, 'ISO-8859-1');
...
```