

Getting details about a file: reading metadata

Imagine you have an input field like this:

```
Select one or more files: <input type="file" id="input"/>
```

This will render as a "select files" or "browse files" button. If you select one file in the file chooser dialog that has popped up, before HTML5 you couldn't do anything with it in the client-side: no access from JavaScript. With the File API, you can read what we call "file metadata": name, size, type and last modification date.

Look at the the code below: the file API defines a `files` property on the DOM node corresponding to the `<input type="file" .../>` input field. This property is an array.

Select several images: Sélectionner 6 fichiers

Name	Bytes	MIME Type	Last modified date
GuitarFX processor.jpg	1148519	image/jpeg	Thu Jul 02 2015 15:52:14 GMT+0200 (CEST)
jsBinPrefs.jpg	926021	image/jpeg	Thu Jul 02 2015 15:45:15 GMT+0200 (CEST)
localStorageInternals.jpg	1253598	image/jpeg	Thu Jul 02 2015 15:36:07 GMT+0200 (CEST)
localStorage.jpg	299444	image/jpeg	Thu Jul 02 2015 14:49:20 GMT+0200 (CEST)
forExample1.jpg	126033	image/jpeg	Thu Jul 02 2015 14:03:28 GMT+0200 (CEST)
ps4html5.jpg	330092	image/jpeg	Thu Jul 02 2015 11:23:38 GMT+0200 (CEST)

Click the button, select files and show metadata

In the example below, we get in the `selectedFile` variable, the metadata related to the first selected file:

```
var selectedFile = document.getElementById('input').files[0];
// do something with selectedFile.name, selectedFile.size, selectedFile.type
// selectedFile.lastModifiedDate
...
```

EXAMPLE 1: READ METADATA OF THE FIRST SELECTED FILE

[Here is a complete example on JS Bin](#) that uses the code above to get details about the first selected file. Please try it below on your browser (click on the button and choose one file):

Select one or more files: No file chosen

- File name:
- File size:
- File type:
- File last modification date:

Complete source code:

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset=utf-8 />
<title>Reading file metadata</title>
<script>
    function displayFirstSelectedFileMetadata () {
        var selectedFile =document.getElementById('input').files[0];
        document.querySelector("#singleName").innerHTML= selectedFile.name;
10.    document.querySelector("#singleSize").innerHTML= selectedFile.size + "
bytes";
        document.querySelector("#singleType").innerHTML= selectedFile.type;

        document.querySelector("#singleDate").innerHTML= selectedFile.lastModifiedDate;
    }
</script>
</head>
<body>
    Select one or more files: <input type="file" id="input"
                                onchange="displayFirstSelectedFileMetadata();" />
<p>
<ul>
21.    <li>File name: <span id="singleName"></span></li>
        <li>File size: <span id="singleSize"></span></li>
        <li>File type: <span id="singleType"></span></li>
        <li>File last modification date:<span id="singleDate"></span></li>
    </ul>
</body>
</html>

```

EXAMPLE 2: DISPLAY METADATA OF MULTIPLE FILES, USE A FILTER ON THE FILE TYPE

This example is a bit more complicated, as it will display details about all files selected (not only the first) and allows only images to be selected, using the `accept` attribute of the input field: `<input type="file" accept="image/*" .../>`.

[Example on JS Bin](#), or try it in your browser: click on the button, and select multiple image files. Notice that in the file selector, files that are not images will be greyed and non selectable.

Select several images: No file chosen

...

Source code extract:

```

Select several images: <input
type="file" accept="image/*" multiple onchange="filesProcess(this.files)" name="selection"/>

```

```

<p>
<div id="result">...</div>
<script>
    function filesProcess(files) {
        var selection = "<table><tr><th>Name</th><th>Bytes</th><th>MIME Type</th>
                        <th>Last modified date</th></tr>";
        for(i=0; i<files.length ;i++){
11.     file = files[i];
            selection += "<tr><td>" + file.name + "</td><td style=\"text-align:right\">"
                        + file.size + "</td><td>"
                        + file.type + "</td><td> " + file.lastModifiedDate + "</td></tr>";
        }
        selection += "</table>";

        document.getElementById("result").innerHTML= selection;
    }
</script>

```

Explanations:

- *Line 1:* we used the `multiple` attribute to allow the selection of multiple files in the file chooser (using shift or control keys). The `accept="image/*"` attribute is a filter that restricts selection to images only. Finally, the `onchange` listener will call the `fileProcess(...)` function, passing as parameter the list of selected files for the current element (`this.files`).
- *Lines 7 and 12:* we prepare the HTML code for building a `<table>` with the results.
- *Line 10:* this `for` loop builds all the rows that compose the table, adding HTML code to the `selection` string variable. At the end of the loop, this variable contains all the HTML code that corresponds to the table of results.
- *Line 18:* the table is added to the page. We use the `innerHTML` attribute of the DOM element corresponding to the `<div id="result">` in order to insert the table as its child in the DOM tree. As such, the table appears on the page dynamically.