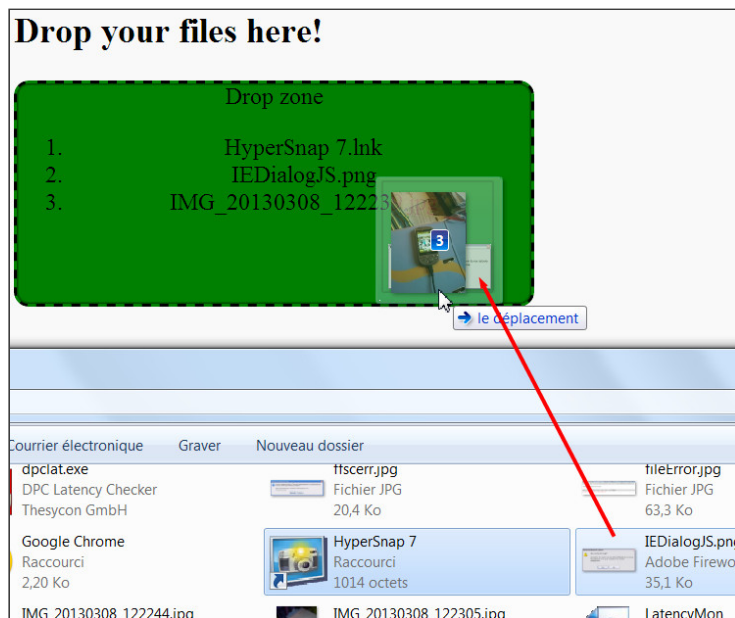


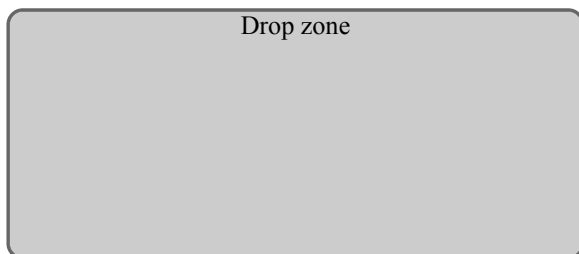
First example: drag and drop files to a drop zone, display file details in a list



Try the example below directly in your browser (just drag and drop files to the greyish drop zone), or [play with it at CodePen](#):



Drop your files here!



Complete source code from the example:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <style>
      div {
        height: 150px;
        width: 350px;
        float: left;
```

```

border: 2px solid #666666;
10. background-color: #ccc;
margin-right: 5px;
border-radius: 10px;
box-shadow: inset 0 0 3px #000;
text-align: center;
cursor: move;
}
.dragged {
border: 2px dashed #000;
20. background-color: green;
}
.draggedOver {
border: 2px dashed #000;
background-color: green;
}

</style>
<script>
30. function dragLeaveHandler(event) {
console.log("drag leave");
// Set style of drop zone to default
event.target.classList.remove('draggedOver');
}
function dragEnterHandler(event) {
console.log("Drag enter");
// Show some visual feedback
event.target.classList.add('draggedOver');
40. }
function dragOverHandler(event) {
//console.log("Drag over a droppable zone");
// Do not propagate the event
event.stopPropagation();
// Prevent default behavior, in particular when we drop images or links
event.preventDefault();
}
50. function dropHandler(event) {
console.log('drop event');
// Do not propagate the event
event.stopPropagation();
// Prevent default behavior, in particular when we drop images or links
event.preventDefault();
// reset the visual look of the drop zone to default
60. event.target.classList.remove('draggedOver');
// get the files from the clipboard
var files =event.dataTransfer.files;
var filesLen = files.length;
var filenames = "";
// iterate on the files, get details using the file API
// Display file names in a list.
70. for(var i = 0 ; i < filesLen ; i++){
filenames += '\n' +files[i].name;

```

```

        // Create a li, set its value to a file name, add it to the ol
        var li =document.createElement('li');

li.textContent = files[i].name;document.querySelector("#droppedFiles").appendChild(li);
    }
    console.log(files.length + ' file(s) have been dropped:\n' + filenames);
}
</script>
</head>
<body>
<h2>Drop your files here!</h2>
<div id="droppableZone"ondragenter="dragEnterHandler(event)"ondrop="dropHandler(event)
                                ondragover="dragOverHandler(event)"
ondragleave="dragLeaveHandler(event)">
    Drop zone
    <ol id="droppedFiles"></ol>
</div>
</body>
88. </html>

```

Note that:

- We prevented the browser default behavior in the `drop` and `dragover` handlers, otherwise if we dropped a media file (an image, an audio or video file), the browser will try to display/play it in a new window/tab. We also stop the propagation for performance reasons, as when we move a dragged object it can raise many events to the parents of the drop zone element.
- *Lines 73-74* create a `` element. Its value is initialized with the file name of the current file in the collection, and added to the `` list.

In principle, this example is very similar to the "fruit" examples we saw in the previous section, except that this time we've been working with files. *And when we work with files, it is important to prevent the browser's default behavior.*