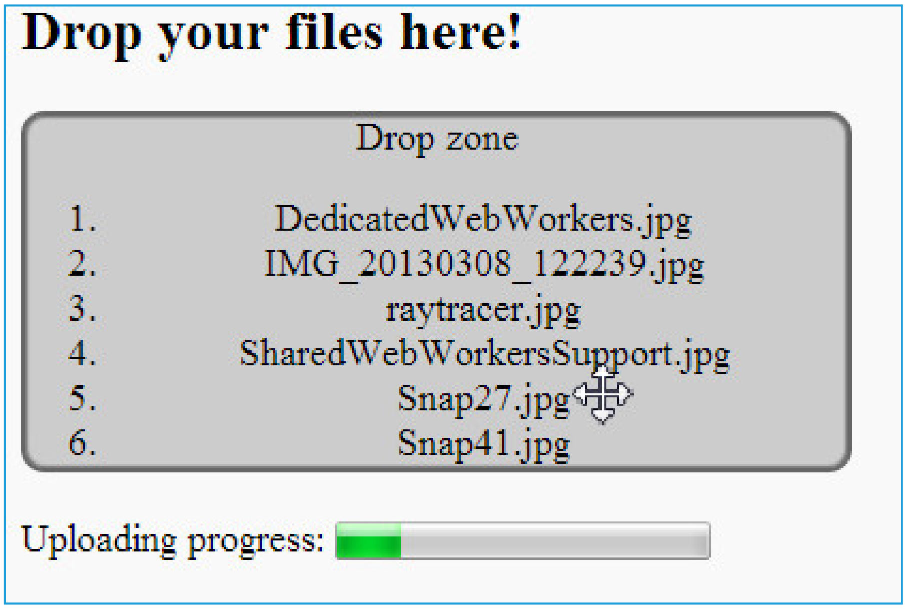
Uploading files using XMLHttpRequest level 2 (XHR2)

This time, let us mix an example we saw earlier (how to upload files using XHR2, with progress monitoring) with another example that uses drag and drop. To achieve this, we just copy and paste some code in a method called `uploadAllFilesUsingAjax()` and add a`<progress>` element to the drag and drop example.

EXAMPLE

Try this interactive example at JSBin (this example does not work on CodePen. We are using a fake remote server and it cancels the connection as soon as we try to connect):



Drop your files here!

Drop zone

1. DedicatedWebWorkers.jpg
2. IMG_20130308_122239.jpg
3. raytracer.jpg
4. SharedWebWorkersSupport.jpg
5. Snap27.jpg
6. Snap41.jpg

Uploading progress:

Source code extract (we omitted the CSS):

```
<!DOCTYPE html>
<html>
 <head>
   <style>
     ...
   </style>
   <script>
     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');
     }
     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();
     }
     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
       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.
       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);
       uploadAllFilesUsingAjax(files);
     }
     function uploadAllFilesUsingAjax(files){
       var xhr = new XMLHttpRequest();
       xhr.open('POST', 'upload.html');
       xhr.upload.onprogress = function(e) {
          progress.value = e.loaded;
          progress.max = e.total;
       };
       xhr.onload = function() {
         alert('Upload complete!');
       };
       var form = new FormData();
       for(var i = 0 ; i < files.length ;i++) {
          form.append('file', files[i]);
       }

       // Send the Ajax request
       xhr.send(form);
     }
  </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>
    <br/>
95. Uploading progress: <progress id="progress"></progress>
    <body>
    <html>
```

We have highlighted the interesting parts in the example above. We build an object of type `FormData` (this comes from the standard JavaScript DOM API level 2), we fill this object with the file contents (*line 77*), then we send the Ajax request (*line 81*) and monitor the upload progress (*lines 66-69*).

Instead of uploading all the files at once, it might be interesting to upload one file at a time with visual feedback, such as: "uploading file MichaelJackson.jpg.....". We will leave this exercise up to you.