

Drag and Drop Archivos y File clásico

En este documento se explica la realización de un proyecto de subida de archivos a un servidor web usando el método Drag and Drop (arrastrando y que nuestros archivos se suban al servidor y mediante el método tradicional con un “input” de tipo “File”.

GitHub del proyecto: <https://github.com/janaigus/subirArchivos.git>

Alumnos:

Airán Rodríguez Rodríguez
Janai G. Expósito Bethencourt

Método Drag and Drop

Comenzaremos viendo la interfaz gráfica de esta parte. Mas concretamente el “area” donde, se podrán arrastrar archivos y como consecuencia serán subidos al servidor.

Subida de Imagenes

Drag & Drop

Normal

Arrastre aqui sus archivos
o
Haga click aqui para seleccionarlos manualmente

Esta parte está compuesta por un “div” cuyo nombre es “caja”. Al que añadiremos los Eventos, “drag”, “dragover” y drop.

```
// Añadir el evento dragover
document.getElementById("caja").addEventListener("drag", permitir, false);
// Añadir el evento dragover
document.getElementById("caja").addEventListener("dragover", permitir, false);
// Añadir el evento drop
document.getElementById("caja").addEventListener("drop", drop, false);
```

Primer paso evento “drag” y “dragover”

Para poder gestionar de manera correcta los eventos “drag” y “dragover”, evitando su comportamiento por defecto, se les ha asignado la función “permitir”.

```
function permitir(ev)
{
    ev.preventDefault();
}
```

Segundo paso evento “drop”

Este evento es el mas importante de los tres ya que en este evento tendremos acceso a los ficheros que hayan sido arrastrados, mediante la lista `ev.dataTransfer.files`. En este ejemplo cada archivo se añadirá a una lista para su posterior envío. Una vez rellenada esa lista se llamará a la función gestionar archivos que comenzará con la carga.

```
function drop(ev)
{
    // Prevenir el comportamiento por defecto en el drop que es abrir el archivo en el propio navegador
    ev.preventDefault();
    // Hacer un bucle for para recorrer las imagenes
    for(var i = 0; i < ev.dataTransfer.files.length; i++){
        archivosCargar.unshift(ev.dataTransfer.files[i]);
    }
    // Comenzar la gestión de los archivos
    gestionarArchivos();
}
```

Tercer paso gestión de los archivos

En este paso, comenzará la carga de los archivos. Mediante la función `GestionarArchivos()`, previa comprobación de que no hay archivos a subir y que no hay ningún archivo “subiendo”, colocaremos los archivos en nuestro “feed” de información al cliente, enviaremos el archivo y posteriormente lo eliminaremos de nuestra lista de archivos.

```
function gestionarArchivos(){
    if(archivosCargar.length > 0){
        if(!cargando){
            // Mostrar la nueva lista formada
            colocarEnFeed();
            // Cargar el primer archivo de la lista
            enviarArchivo(archivosCargar[0]);
            // Quitar el primer elemento de la lista
            archivosCargar.shift();
        }
    }
}
```

Cuarto paso “colocar en feed”

Como hemos visto en el apartado anterior, desde la función gestionar archivos, se hace una llamada a *colocarEnFeed()*, en esta función colocamos el primer elemento de nuestra lista en el apartado de progreso, y los demás archivos en el apartado de espera para que el usuario tenga conocimiento de lo que pasa en cada momento, además se ha añadido una imagen a cada tipo de archivo, siendo en el caso de las imágenes una miniatura de la misma.

```
function colocarEnFeed(){
    // Mostrar la seccion del feed
    var feedRow = document.getElementById("feed");
    if(feedRow.style.display != "block"){
        feedRow.style.display = "block";
    }
    // Colocar el archivo que está siendo descargado
    var fich = archivosCargar[0];
    var tam = fich.size / 1024;
    if(tam > 1024){
        tam = tam / 1024;
        document.getElementById("tamSubiendo").innerHTML = tam.toFixed(2) + " mb";
    }
    else{
        document.getElementById("tamSubiendo").innerHTML = tam.toFixed(2) + " kb";
    }
    var srcImagen = document.getElementById("imagenSubiendo");
    // Colocar la barra de progreso a 0
    var barra = document.getElementById("barraProgreso");
    barra.style.width = "0%";
    // Switch sobre la "extension" del fichero
    switch(fich.name.substring(fich.name.lastIndexOf(".") + 1, fich.name.length).toLowerCase()){
        // Comprimidos
        case "zip":
        case "7z":
        case "rar":
        case "tar.gz":
        case "gz":
            srcImagen.src = "./img/icono-archivo-comprimido.png";
            break;
        // Imagenes
        case "jpg":
        case "png":
        case "jpeg":
        case "gif":
            srcImagen.src = URL.createObjectURL(fich);
            break;
        // Música
        case "mp3":
        case "flac":
        case "aac":
        case "ogg":
            srcImagen.src = "./img/icono-musica.png";
            break;
        // Videos
        case "mpeg":
        case "mkv":
        case "mp4":
            srcImagen.src = "./img/icono-video.png";
            break;
        default:
            srcImagen.src = "./img/icono-indefinido.png";
            break;
    }
    // Colocar el nombre y el tamaño en el div correspondiente
    document.getElementById("nombreSubiendo").innerHTML = fich.name;
    // Colocar los demás elementos en la cola
    var divCola = document.getElementById("esperando");
    divCola.innerHTML = '<h2 align="center">Esperando</h2>';
    for(var i = 1; i < archivosCargar.length; i++){
        divCola.innerHTML += archivosCargar[i].name + " | Esperando... <img src='./img/ajax_clock_small.gif'><br/>";
    }
}
```

Quinto paso enviar archivo

Desde `gestionarArchivos()` también se llamaba a esta función, lo que hace es enviar un archivo cada vez, que tal y como está creada la llamada será siempre el primero. Esta función realiza una petición de tipo POST al servidor, incluyendo el archivo, Mediante el objeto `FormData`, que se adjunta en la petición.

Además se añade el evento “progress” sobre el “upload” del objeto ajax para así obtener el progreso de nuestra operación en todo momento.

```
function enviarArchivo(archivo){
    if(objAjax){
        // Crear objeto form data
        var formulario = new FormData();
        formulario.append('archivo', archivo);
        objAjax.onreadystatechange = procesarRespuesta;
        objAjax.open('POST', 'php/ProcesarArchivos.php', true);
        // Añadir el evento a la barra de progreso
        objAjax.upload.addEventListener("progress", barraProgreso, false);
        objAjax.send(formulario);
    }
}
```

Sexto paso gestionar ajax

Como hemos visto en el paso anterior, hemos realizado varias cosas con nuestro objeto ajax. En lo referente al evento sobre la barra de progreso cabe destacar que la barra de progreso resultante siempre es la misma, siendo esta un componente de bootstrap que al cambiar su atributo `style`, con los datos procedentes del evento modificará su apariencia en consecuencia.

```
// Funcion que aumenta progresivamente la barra de progreso
function barraProgreso(ev){
    var barra = document.getElementById("barraProgreso");
    var complete = Math.round(ev.loaded / ev.total * 100);
    //progress.value = progress.innerHTML = complete;
    barra.style.width = complete + "%";
}
```

Como es habitual al trabajar con “ajax” hemos asignado una función a al evento *Onreadystatechange*. La función es *preprocesarRespuesta()*, y lo que hará será mantener una “variable centinela al corriente de nuestra petición”.

Una vez finalizada la carga se colocará en el “feed” la información necesaria, que Proviene de la respuesta dada por el servidor. Además se volverá a llamar a la Función *gestionarArchivos()* para que en el caso que se necesario, continúe la carga de los siguientes archivos

```
function procesarRespuesta(ev){
    switch(this.readyState){
        case 0:
        case 1:
        case 2:
        case 3:
            // Controlar cuando se está cargando un archivo
            cargando = true;
            break;
        case 4:
            if(this.status == 200){
                // Hacer lo necesario con response text
                cargando = false;
                // Colocar los elementos finalizados
                document.getElementById("finalizado").innerHTML += this.responseText;
                gestionarArchivos();
            }
            break;
    }
}
```

Séptimo y último paso recibir los archivos

En esta ocasión hemos elegido PHP a la hora de procesar nuestros archivos en el lado del servidor.

A la hora de recibir el archivo hemos usado la variable “\$_FILES” de PHP como si de un “input” normal se tratara y luego hemos devuelto un mensaje con la Información a colocar en el “feed”.

```
<?php
    if($_FILES){
        $mensaje = "";
        $tmp_name = $_FILES["archivo"]["tmp_name"];
        $name = $_FILES["archivo"]["name"];
        if(!move_uploaded_file($tmp_name, "../uploads/$name")){
            $mensaje = '<span class="glyphicon glyphicon-remove-circle"></span> '
                . $name . ' | No se ha podido subir el archivo correctamente<br />';
        }else{
            $mensaje = '<span class="glyphicon glyphicon-ok-circle"></span> '
                . $name . ' | Completado correctamente<br/>';
        }
        echo $mensaje;
    }
?>
```

Método “File Clásico”

Creación de la página principal

En principio crearemos una página en formato .php, en la cual pondremos un formulario con un “input” de tipo “file”, este tiene que tener varias opciones. El formulario tiene que tener el método POST y el atributo “enctype=multipart/formdata”. El “input” file es normal, aunque si queremos subir varios archivos a la vez tendremos que poner en el atributo “name” un nombre del tipo `name:"nombre[]"`, los corchetes definirán que lo que le pasemos al php sea un array. Además el atributo multiple.

```
<!-- la opcion multipart/form-data, y el metodo POST -->
<form id="frm01" action="../../../php/subidasphpPruebas.php" method="POST" enctype="multipart/form-data">
  <!-- Si queremos subir varios archivos, tenemos que "declarar" que pasaremos un array con los archivos -->
  <!-- Para esto el name lo ponemos con [] como se ve a continuacion -->
  <input type="file" id="imagenInput" name="imagen[]" multiple>
</br>
<button class="btn btn-default" type="submit" value="Enviar" name="Enviar" id="Enviar">Enviar</button>
</form>
```

Recogida de datos

Ahora miramos el código de php, como recogemos los datos y como subiremos los archivos y por ultimo guardamos datos en una variable `session` para mostrarlos en nuestra página principal.

```
if($_FILES){
    //Creamos una variable contador que usaremos para recorrer en nuestra
    //pagina principal la variable session donde guardaremos los datos de los
    //archivos que subamos
    $cont = 0;
    for($i=0; $i<count($_FILES['imagen']['name']); $i++) {
        //Cojemos una referencia temporal del archivo
        $tmpFilePath = $_FILES['imagen']['tmp_name'][$i];
```

Lo primero que haremos será preguntar si hemos recibido algún archivo mediante la variable global `$_FILES`, si es así recorreremos mediante un “for” todos los archivos que se haya mandado por el formulario y en una variable guardaremos una referencia del archivo.

Luego preguntaremos si el archivo se ha cargado, siendo esto así, haremos varias cosas antes de subirlo a nuestro servidor.

Primero, en este ejemplo hemos creado una variable contador, a la que la aumentamos y se la asignamos a una variable “session” que creamos en este momento y recogemos la imagen, en otra variable.

```
if (is_uploaded_file($tmpFilePath)){
    $cont = $cont + 1;
    //aumentamos contador y asignamos a la variable session que usaremos en la página principal
    $_SESSION['cont'] = $cont;
    //recojo la imagen
    $imagen = $_FILES['imagen']['name'][$i];
    //Obtenemos el nombre de la imagen y la extensión del archivo
    $imagen1 = explode(".", $imagen);
    //Generamos un nombre aleatorio con números y asignamos la extensión obtenida anteriormente
    $imagen2 = $imagen1[0].rand(0,9).rand(100,9999).".". $imagen1[1];
    //Subimos y preguntamos si se subió la imagen
```

En segundo lugar hacemos un explode de la imagen por el . y la asignamos a otra variable, así tenemos en la posición [0] de esta el nombre del archivo y en la posición[1] la extensión.

Finalmente generamos un nuevo nombre para la imagen que será, el nombre original mas un numero aleatorio y su extensión.

Para acabar este php, solo nos falta subir los archivos, empezamos preguntando si se han movido los archivos al servidor

```
if (move_uploaded_file($_FILES['imagen']['tmp_name'][$i], "../uploads/".$imagen2)){
    //guardamos en las session los nombres de los archivos subidos
    $_SESSION['imagen'] = "subida";
    $_SESSION['nombre'][$i] = $_FILES["imagen"]["name"][$i];
    //redirección a la página principal
    header("Location: normal.php");
}else{
    //si no se subieron las imágenes guardamos en una variable session un mensaje del error y redirección a la página principal
    $_SESSION['error'][$i] = "La subida es imposible para el archivo".$_FILES["imagen"]["name"][$i];
    header("Location: normal.php");
}
```

Como vemos en caso de subir los archivos al servidor simplemente guardamos en unas variables sesión que se ha subido y el nombre del archivo que hemos subido, por último redireccionamos a nuestra página principal donde mostraremos que archivos se han subido y cuáles no, que es en el caso de que el archivo no se suba, entonces guardamos un mensaje de error en otra variable “session”.

Aquí dejo el código completo para que se vea mejor:

```
<?php
    session_start();
    //igual que hicimos en nuestra pagina principal iniciamos sesion para poder pasar estos parametros

    // Primero comprobamos si ha llegado algun archivo
    if($_FILES){
        //Creamos una variable contador que usaremos para recorrer en nuestra
        //pagina principal la variable sesion donde guardaremos los datos de los
        //archivos que subamos
        $cont = 0;
        for($i=0; $i<count($_FILES['imagen']['name']); $i++) {
            //Cojemos una referencia temporal del archivo
            $tmpFilePath = $_FILES['imagen']['tmp_name'][$i];

            //preguntamos si hemos recojido
            if ($tmpFilePath != ""){
                //Preguntamos si se cargo el archivo
                if (is_uploaded_file($tmpFilePath)){
                    $cont = $cont + 1;
                    //aumentamos contador y asignamos a la variable sesion que usaremos en la página principal
                    $_SESSION['cont'] = $cont;
                    //recojo la imagen
                    $imagen = $_FILES['imagen']['name'][$i];
                    //Obtenemos el nombre de la imagen y la extensión del archivo
                    $imagen1 = explode(".", $imagen);
                    //Generamos un nombre aleatorio con números y asignamos la extensión obtenida anteriormente
                    $imagen2 = $imagen1[0].rand(0,9).rand(100,9999).".". $imagen1[1];
                    //Subimos y preguntamos si se subio la imagen
                    if (move_uploaded_file($_FILES['imagen']['tmp_name'][$i], "../uploads/".$imagen2)){
                        //guardamos en las session los nombres de los archivos subidos
                        $_SESSION['imagen'] = "subida";
                        $_SESSION['nombre'][$i] = $_FILES["imagen"]["name"][$i];
                        //redireccion a la pagina principal
                        header("Location: normal.php");
                    }else{
                        //si no se subieron las imagenes guardamos en una variable session un mensaje del error y redireccion a la pagina principal
                        $_SESSION['error'][$i] = "La subida es imposible para el archivo".$_FILES["imagen"]["name"][$i];
                        header("Location: normal.php");
                    }
                }
            }
        }
    }
    //del if
}
//del for
}
?>
```


Gestión de mensajes al usuario

Tanto en nuestra pagina principal como en la página php que sube los archivos al servidor debemos iniciar una sesión con: `session_start()`;

Seguimos mirando ahora como mostrar en nuestra página principal que efectivamente se han subido los ficheros

Primero como siempre preguntamos si existe una sesión, en este caso `$_SESSION['imagen']`, que nos indicara que se han subido las imágenes.

```
<?php
    if(isset($_SESSION['imagen'])){?>
    <!-- Si existe la session ['imagen'] significa que el php ha subido las imagenes -->
```

Luego dentro de nuestro código html, que “maquetamos” cada uno a su gusto recogemos la variable contador que creamos en el php para mostrar el nombre del archivo y un mensaje de que esta se ha subido correctamente.

```
<?php
    $fichSubidos = $_SESSION['cont'];
    for($i = 0; $i<$fichSubidos;$i++){
        echo $_SESSION['nombre'][$i]. " | Completado correctamente <br>";
    }

?>

    </div>
</div>

<?php
}
//Si no existe la sesion['error'] es que alguno o todos los archivos no se han subido correctamente
if(isset($_SESSION['error'])){
    $fichSubidos = $_SESSION['cont'];
    for($i = 0; $i<$fichSubidos;$i++){
        echo $_SESSION['error'][$i];
    }
}
//imprimimos error y cerramos la sesion para que no haya problemas de f5
session_destroy();

?>
```

Y en caso de que exista la variable error recorreremos las sesiones de esta variable y mostramos aquellos archivos que no se hayan podido subir con su mensaje de error correspondiente.

Por último destruimos la sesión con `session_destroy()`, para que al recargar la pagina todos estos datos desaparezcan.