File input (or “upload”) in HTML forms

A form in an HTML document (Web page) can contain an input element with type="file" (<input> elements are used within a <form> element to declare input controls that allow users to input data. An input field can vary in many ways, depending on the type attribute. https://www.w3schools.com/tags/tag\_input.asp). This may let the user include one or more files into the form submission. The form is often processed so that such files are stored onto the disk of the Web server; this is why file input (or file submission) is often called “file upload.” File input opens interesting possibilities, but browser support is still limited and generally of poor quality even in newest versions. Moreover, users are often puzzled with it, since most people use file input rather rarely.

## The basics

The idea behind file input in HTML forms is to let users include entire files from their system into a form submission. The files could be text files, image files, or other data. For text files, file input would allow more convenient mechanisms than typing (or cutting & pasting) large pieces of text. For binary data, such as images, file input would be not just more convenient but usually the only practical way. For more information on the design principles of file input, see [RFC 1867](http://www.faqs.org/rfcs/rfc1867.html), *Form-based File Upload in HTML*.

Writing an HTML form with a file input field is rather simple. The difficult thing is actually to find or write a server-side script which can *do* something useful when it receives data in such a format. And the *really* difficult thing is to make such processing robust and controlled so that all data is processed properly and so that someone won’t e.g. fill your server’s disk space with gigabytes of junk, by ignorance or by malevolence.

You need to know the general basics of writing HTML forms; if you need links to tutorials and references on forms, consult [*How to write HTML forms*](http://www.cs.tut.fi/~jkorpela/forms/index.html). Then, what you need to do **in HTML** is to write a form so that

* the action attribute refers to a server-side script which is capable of handling submissions containing forms or, technically speaking, being in multipart/form-data format; as explained below, don’t even dream about using mailto: URLs in action attributes, in this context or otherwise!
* it has the attribute [method="post"](http://www.cs.tut.fi/~jkorpela/forms/methods.html)
* it has the attribute [enctype](http://www.cs.tut.fi/~jkorpela/forms/file.html#enctype)="multipart/form-data"
* it contains a field <input type="file" name="*somename*" size="*chars*"> where;
  + *somename* is a name you assign to the field as you like; the form data set will contain the content of the file “under that name”, and that name has nothing to do with the filename
  + *chars* is an integer specifying the desired width, as a number of characters, of the filename box to be displayed; the size attribute is optional, but setting it to some relatively large value (say 40) probably helps the user, since the default width of the box in current browsers is rather narrow for typical filenames. (See [notes on the size attribute](http://www.cs.tut.fi/~jkorpela/forms/file.html#size).)

Minimally, the form needs to contain [a submit element](http://www.htmlhelp.com/reference/html40/forms/input.html#submit) too. It may also contain any other fields you like, and explanatory texts, images, etc.

A common problem with file input in forms is that form data gets sent but only the name of the file is included. The reason is typically that the form element does not contain the attributes mentioned above.