So what does that **`secure_filename()`** function actually do? Now the problem is that there is that principle called "never trust user input". This is also true for the filename of an uploaded file. All submitted form data can be forged, and filenames can be dangerous. For the moment just remember: always use that function to secure a filename before storing it directly on the filesystem.

So first we need a couple of imports. Most should be straightforward, the **`werkzeug.secure_filename()`** is explained a little bit later. The `UPLOAD_FOLDER` is where we will store the uploaded files and the `ALLOWED_EXTENSIONS` is the set of allowed file extensions.

Why do we limit the extensions that are allowed? You probably don't want your users to be able to upload everything there if the server is directly sending out the data to the client. That way you can make sure that users are not able to upload HTML files that would cause XSS problems (see Cross-Site Scripting (XSS)). Also make sure to disallow `.php` files if the server executes them, but who has PHP installed on their server, right? :)