

Question 1

Créez une page avec un formulaire en méthode `post` qui permet d'envoyer un fichier.

Pour cela il faudra ajouter l'attribut `enctype="multipart/form-data"` dans la balise `form`.

La balise `<input>` avec l'argument `type="file"` permet à l'utilisateur de choisir un fichier.

N'oubliez pas aussi un bouton d'envoi !

 **Rappel : il faut un attribut `name` dans les balises `<input>` pour pouvoir les récupérer dans `$_POST` ou `$_GET`.**

Question 2

Pour récupérer les fichiers envoyés par l'utilisateur, PHP nous fournit la superglobale `$_FILES` qui fonctionne de la même façon que les autres superglobales comme `$_POST` ou `$_GET`.

Testez que le formulaire a été envoyé.

Si le formulaire a été envoyé, affichez le contenu de `$_FILES`.

Qu'est-ce que contient cette superglobale ?

Question 3

La présence d'un fichier dans `$_FILES` ne signifie pas que le fichier est enregistré sur notre serveur. Cela serait bien trop risqué de laisser un utilisateur enregistrer un fichier dans notre serveur sans en vérifier le contenu !

On va donc dire que notre serveur n'accepte que les images dont l'extension est `jpeg`, `jpg` ou `png`.

On commence donc par créer un array avec toutes les extensions autorisées :

```
<?php
    $extensions = array("jpeg", "jpg", "png");
```

?>

Pour récupérer l'extension d'un nom de fichier, on utilise la fonction `pathinfo`.

Disons que vous avez récupéré le nom du fichier envoyé par l'utilisateur dans une variable `$nom_fichier`, regardez ce qu'affiche le code suivant :

```
$fichier = pathinfo($nom_fichier);  
  
echo "<pre>" . print_r($fichier, true) . "</pre>";
```

Comment peut-on récupérer l'extension du fichier ?

Question 4

Une fois l'extension récupérée, on peut vérifier qu'elle fait partie des extensions autorisées en utilisant la fonction `in_array` qui prend en premier argument la valeur qu'on cherche, et en deuxième le tableau où on cherche la valeur en premier argument.

Par exemple :

```
<?php  
    $fruits_ete = array("abricot", "pêche", "nectarine");  
  
    if (in_array("pêche", $fruits_ete)) {  
        echo "La pêche est bien un fruit d'été";  
    }  
?  
>
```

Ici, on entre dans le `if` car "pêche" existe bien dans `$fruits_ete`.

Écrivez donc maintenant la condition qui affiche "L'extensions est bonne" si l'extension du fichier est dans `$extensions`, ou sinon afficher "Cette extension n'est pas acceptée".

Question 5

On va enregistrer les fichiers de nos utilisateurs dans un même dossier qu'on appellera `images`.

On peut créer un dossier directement en PHP, avec le code suivant :

```
mkdir("nom_dossier", 0777); // crée un dossier avec tous les
droits
```

Néanmoins, on n'a besoin de le créer qu'une seule fois ! On va donc appeler `mkdir` que si le dossier n'existe pas. On peut vérifier cela avec

`file_exists("nom_dossier")` qui renvoie vrai si le dossier existe déjà, faux sinon.

Écrivez donc un condition qui regarde si le dossier *n'existe pas*, et qui le crée si c'est le cas.

Question 6

On a maintenant un moyen de vérifier le contenu du fichier de l'utilisateur, et on a un endroit où l'enregistrer.

On peut donc maintenant officiellement l'enregistrer sur notre serveur avec la fonction `move_uploaded_file` qui prend en premier argument le *nom temporaire* du fichier (cf. le contenu de `$_FILES` ...) et en deuxième, l'endroit où on veut l'enregistrer.

Regardez votre code, et cherchez où appeler `move_uploaded_file`, en vous posant les questions suivantes :

1. Quelles sont les conditions qui autorisent un fichier à être enregistré ?
2. Où est-ce que je souhaite que ce fichier soit enregistré ?

Question 7

On peut récupérer les fichiers qui sont dans un dossier dans notre code PHP avec la fonction `scandir` qui prend en argument le nom du dossier.

Appelez `scandir` pour voir le contenu du dossier `images`.

Quel type de données renvoie `scandir` ? Est-ce que cela nous permet de récupérer chaque nom de fichier ?

Question 8

Affichez chaque nom de fichier sous forme de liste.

On affiche un fichier à condition qu'il ne soit pas un dossier (on peut utiliser `is_dir` pour vérifier) et que son extension soit bien celle d'une image (on peut réutiliser `$extensions`).

Question 9

On veut maintenant faire en sorte qu'à chaque envoi d'une image, celle-ci s'affiche sur la page après qu'elle a été enregistrée dans le serveur.

Comment pourrait-on faire cela ? Comment sait-on quand l'image a été envoyée ? Comment affiche-t-on une image dans du HTML ? Où serait-il judicieux de positionner ce nouveau morceau de code dans notre code actuel ?