

SAÉ 1.03

Démo PHP ➡ HTML

<?php ... ?>

TP 9 - Page 3 :

*Héritage de son usage initial pour le Web, un script PHP doit obligatoirement être placé entre un tag ouvrant **<?php** et un tag fermant **?>***

Tout ce qui est placé en dehors de ces tags ne sera pas interprété comme du code PHP mais s'affichera simplement à l'écran. Sans entrer dans le détail, ce comportement a de l'intérêt pour créer des pages Web mais n'en a aucun dans notre cas.

Ça n'avait pas d'intérêt pour le TP 9 mais ça va en avoir pour la SAÉ !

Exemple de script PHP mixant du PHP et du HTML :

```
<?php
    $jours = ['Lundi', 'Mardi', 'Mercredi', 'Jeudi', 'Vendredi',
'Samedi', 'Dimanche'];
?>
<!DOCTYPE html>
<html lang="fr">
<head>
    <title>Document</title>
</head>
<body>
    <h1>Semaine</h1>
    <ul>
<?php
    foreach($jours as $un_jour) {
?>
        <li><?php echo $un_jour ?></li>
<?php
    }
?>
    </ul>
```

```
</body>
</html>
```

Tout ce qui est en **bleu** est en dehors des tags `<?php ?>` et se trouve donc affiché tel quel. C'est comme si on avait écrit en PHP un gros **echo** de tout ce qui est en bleu.

Tout ce qui est en **rouge** est dans des tags `<?php ?>` et se trouve donc interprété comme du code PHP.

C'est une succession d'ouvertures et de fermetures de tags `<?php ?>`.

Ce n'est pas parce qu'on ferme un bloc PHP (tag `?>`) que les variables sont oubliées lors de l'ouverture du tag `<?php` suivant. Il en va de même pour les structures de contrôle, comme les boucles par exemple : si une boucle est démarrée, elle peut se terminer dans un autre bloc `<?php ?>` plus bas, c'est d'ailleurs le cas dans la démo ci-dessus.

Description du code

Bloc 1 (**rouge**)

Dans le code précédent, on commence par une portion de code PHP qui initialise un tableau **\$jours** contenant des chaînes de caractères (les noms des jours de la semaine).

Bloc 2 (**bleu**)

C'est le traditionnel en-tête d'une page HTML. On est en dehors d'un bloc PHP, toute cette partie est donc affichée en l'état.

Notez que ce bloc se termine par l'ouverture d'une liste HTML (``).

Bloc 3 (**rouge**)

Cette portion de code PHP démarre une boucle : un **foreach()** qui est la structure de boucle traditionnelle pour parcourir un tableau en PHP (TP 9 - Page 14).

Vous noterez que la boucle n'est pas formée dans ce bloc PHP, elle le sera dans le prochain bloc.

A chaque passage dans la boucle, c'est ce qui se trouve entre le `{` ouvrant la boucle et le `}` la fermant, plus loin, qui sera affiché. Cette portion qui est affichée à chaque tour de boucle est composée d'une partie **bleue** (affiché en l'état) et d'une partie **rouge** (interprétée par PHP).

Bloc 4 et 6 (**bleu**)

Comme expliqué ci-avant, à chaque tour de boucle on affiche, en l'état, une ouverture et une fermeture d'une ligne de liste HTML sous la forme d'un `...`.

Bloc 5 (**rouge**)

A l'intérieur de chaque `...` un tout petit bout de code PHP est exécuté, à chaque passage dans la boucle, qui écrit le nom du jour, qui provient de la variable `$un_jour` alimenté par le `foreach()`.

Bloc 7 (**rouge**)

La fermeture de la boucle `foreach()`.

Bloc 8 (**bleu**)

Le reste de la page HTML qui termine la liste `` et ferme le code HTML.

Test

Pour tester le code ci-dessus, vous pouvez le faire directement sur votre ordinateur (IUT) de cette manière. En supposant que votre script est nommé **test.php** :

```
| php test.php
```

Pour générer un fichier **test.html** :

```
| php test.php > test.html
```

et ouvrez le **test.html** dans un navigateur pour voir le résultat.