

NSI – 1ere	TP Séquence 10A : Modalités de l'interaction Homme Machine sur le WEB – Client Serveur.	LFV
------------	---	-----

1) Mise en place des fichiers et logiciels de travail

Consigne n° 1 (pour pouvoir prendre le rôle d'Alice) :

Nous allons utiliser les fichiers de l'archive TP_PHP.zip. Dézipper cette archive puis copier/coller le dossier TP_PHP vers un dossier NSI de votre dossier web (P:\EspacePersonnel\Documents\WEB\WWW\NSI).

Ouvrez votre explorateur de fichier et rendez vous à la racine de votre dossier Web puis rendez-vous dans le dossier NSI/TP_PHP et éditez avec notepad++ le fichier TP_php.php

Consigne n° 2 (pour pouvoir prendre le rôle de Bob) :

Ouvrez votre navigateur et rendez vous à l'adresse <http://192.168.xxx.yyy/NSI/> puis rendez-vous dans le dossier TP_PHP et ouvrez la page TP_php.php

2) Premiers pas

Question 1 :

Modifiez le code (Alice) afin d'obtenir ceci :

Enregistrez vos modifications (Alice) puis actualisez votre page dans votre navigateur avec CTRL+F5 pour actualiser le cache en même temps (Bob)

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <link rel="icon" type="image/png" href="ico.png"/>
5      <meta charset="utf-8">
6      <title> PHP </title>
7      <link rel="stylesheet" href="style.css">
8
9  </head>
10
11 <body>
12     <h1> TP php </h1>
13     <hr>
14     <p><?php echo 'Ce paragraphe est généré par PHP.'; ?></p>
15     <hr>
16 </body>
17 </html>

```

Question 2 :

- Quelles sont les balises qui ouvrent et ferment le code PHP ?
Balise ouvrante <?php et fermante ?>
- Quelle instruction PHP a été utilisée dans cet exemple et qu'a-t-elle fait ?
L'instruction **echo : génère dans le fichier html envoyé au client la chaîne de caractères qui la suit.**
- Comment se termine une ligne d'instruction en PHP ?
La ligne d'instruction en PHP se termine par un point-virgule.
- Qu'utilise-t-on pour délimiter une chaîne de caractères ?
Une guillemet simple ' .

A ce stade PHP n'a aucun intérêt... le code PHP placé dans ce code HTML a été remplacé par la chaîne de caractère envoyée par la commande **echo**... nous aurions pu faire la même chose directement en HTML. Voyons alors les intérêts de PHP dans les pages Web...

3) Des progrès

Question 1 :

Modifiez le code de TP_php.php de cette manière (Alice).

Enregistrez vos modifications (Alice) et actualisez

votre page dans votre navigateur (Bob).

```
<body>
  <h1> TP php </h1>
  <hr>
  <p>
    <?php
      echo 'Bonjour, nous sommes le <b>' . date("d-m-Y") . '</b> ' ;
      echo 'et il est : <b>' . date("H:i:s") . '</b>.' ;
      echo 'Ce paragraphe est généré par PHP.' ;
    ?>
  </p>
  <hr>
</body>
```

Question 2 :

- Quelle instruction PHP a permis d'afficher quelque chose de dynamique sur la page Web irréalisable en HTML ?

L'instruction `date()` qui permet d'afficher la date du jour et l'heure d'affichage à la seconde près.

A noter qu'il s'agit d'une fonction PHP. Vous pouvez retrouver à cette adresse quelques fonctions PHP : <http://cours-web.estsurinternet.com/?p=155> ou la totalité des fonctions PHP à partir du manuel officiel : <https://www.php.net/manual/fr/funcref.php>

- A quoi sert le point que l'on trouve avec l'instruction `echo` ? à faire une multiplication ou à concaténer ?

Il permet la concaténation de chaîne de caractère (rappel : avec JS on utilisait +).

- Comment est traité `` par le navigateur et pourquoi ?

Il affiche en gras les caractères entre ces deux balises... il s'agit effectivement de la balise HTML **bold** et elle est traitée comme telle....

4) Avec des variables ...

Question 1 :

Modifiez à nouveau le code de TP_php.php de cette manière (Alice)

Enregistrez vos modifications (Alice) et actualisez votre page toutes

le 3 secondes environ pendant une minute dans votre navigateur (Bob).

```
15 <?php
16 $seconde = date("s");
17 if ($seconde >= 0 && $seconde < 30){
18     $partie = 'première moitié de la minute en cours';
19 }
20 else{
21     $partie = 'deuxième moitié de la minute en cours';
22 }
23 echo 'Bonjour, nous sommes le <b>' . date("d-m-Y") . '</b> ' ;
24 echo 'et il est : <b>' . date("H:i:s") . '</b>.' ;
25 echo 'Nous sommes dans la ' . $partie . '.' ;
26 echo 'Ce paragraphe est généré par PHP.' ;
27 ?>
```

- Quelles sont les instructions et leur syntaxe qui permet de gérer les conditions ?

Les instructions `if {} else {}`.

- Comment PHP distingue-t-il les noms de variables ?

Le nom de variable est précédé d'un `$`.

PHP gère également les boucles FOR, WHILE, la sélection SWITCH... ce n'est pas l'objectif de développer davantage les instructions PHP dans ce cours mais pour ceux qui s'y intéressent vous pouvez vous documenter ici (entre autres...) : <https://www.pierre-giraud.com/php-mysql-apprendre-coder-cours/condition-if-else/>

5) Pauuuuse ! Réfléchissons un peu !

- Prendre le rôle de Bob (navigateur) et aller dans les options de Firefox : activer les outils de développement web.
- Vérifier que le code html de la page que Bob est en train de consulter est différent du code qu'Alice a déposé sur le serveur.
- Qu'est ce que cette différence entre les deux fichiers (php d'Alice sur le serveur et html de Bob sur le client) nous montre concernant le fonctionnement de php ?

Elle nous montre que le fichier php modifié et actualisé d'Alice contient bien les instructions permettant au serveur php de "construire" un fichier html qui est envoyé au navigateur de Bob. Le fichier html généré n'est pourtant pas présent sur le serveur : il est envoyé directement à Bob.

6) Entrapercevoir la puissance de PHP

On veut ficher un nuancier de couleurs dans un tableau. En HTML, les couleurs sont codées sur 24 bits (3 octets), avec un octet pour le rouge, un octet pour le vert et un octet pour le bleu. La couleur peut être notée soit en hexadécimal : #FF8000 ou en rgb : rgb(255,128,0). Il y a donc potentiellement 2^{24} couleurs, soit environ 16 millions.

En html; le code pour obtenir une ligne de tableau avec un nom de couleur et une couleur est :
'<tr><td>rgb(255,0,0)</td><td style="background-color:rgb(255,0,0);"></td></tr>'.
On peut alors faire une triple-boucle en php pour produire un grand nuancier de couleurs :

```
<?php
function AfficheLigneCouleur($rouge, $vert, $bleu){
    $ligne = '<tr><td>rgb(' . $rouge . ',' . $vert . ',' . $bleu . ')</td>';
    $ligne = $ligne . '<td style="background-color:rgb(' . $rouge . ',' . $vert . ',' . $bleu . ');"></td></tr>';
    return ($ligne);
}
?>
<!DOCTYPE html>
<html>
  <head>
    .....
  </head>
  <body>
    <h1> TP php - Couleurs en (R,G,B)</h1>
    <table style="width:250px;"><col span"2" style="width:125px:">
      <?php
        for($i=0;$i<=255;$i=$i+15) {
          for($j=0;$j<=255;$j=$j+15){
            for($k=0;$k<=255;$k=$k+15){
              echo AfficheLigneCouleur($i, $j, $k);
            }
          }
        }
      ?>
    </table>
  </body>
</html>
```

Question 1 :

Ce code est présent dans couleurs.php : avec le rôle de Bob (navigateur) testez le et observez le code html du fichier html que Bob consulte. Le constat est-il le même qu'au 5) ? **Oui, tout à fait, c'est encore plus flagrant car le fichier html reçu par Bob mais généré par les instructions php d'Alice fait plusieurs milliers de lignes de html !**