

Pourquoi a-t-on besoin d'un autre langage que HTML et CSS ? :

PHP est un langage de programmation dédié au WEB. Contrairement à HTML, il est capable de suivre un algorithme, c'est-à-dire d'effectuer une séquence d'opérations conditionnelles ou non (manipuler des variables, faire des tests, des boucles...). Il est assez proche d'autres langages comme JAVA ou C++. Son « plus » est d'être parfaitement adapté à un usage pour le WEB.

Attention : **Seul un serveur peut interpréter le langage PHP**. Notre logiciel client (Internet Explorer, Chrome, etc....) est incapable de comprendre PHP. Il se contentera d'afficher ce que le serveur lui enverra. Mon programme en PHP doit donc se situer sur un serveur capable de le comprendre. Tous les logiciels serveurs actuels savent interpréter PHP. C'est donc le cas du serveur que nous avons déjà utilisé. Il faudra donc placer nos fichiers PHP dans le répertoire dédié (cf TP « **TRANSFORMER SON PC EN SERVEUR HTTP** »)

Au terme de ce cours, nous allons **réaliser un accès par mot de passe**. Avant cela, nous allons faire plus simple :

1/ Mon premier fichier PHP

- Créer une page HTML où devra simplement s'afficher votre nom et votre prénom (n'oubliez pas de mettre toutes les balises nécessaires (<HTML>, <HEAD>, <BODY>, etc...) . Vous la nommerez par exemple **ex1php.php**. Vous la sauvegarderez dans le répertoire dédié de votre serveur (**P:/SERVEUR WEB/_web.zmwsc**)
- Ouvrez le fichier créé dans le navigateur pour vérifier que tout va bien

Nous allons ajouter du code PHP.

Pour signifier que c'est du code PHP et plus de l'HTML, il faut utiliser ce balisage :

```
<?php ..... ?>
```

L'instruction pour écrire de l'HTML est « echo »

Elle s'utilise ainsi :

```
<?php  
echo " <H2> bonjour </H2> " ;  
?>
```

- Insérer ce bout de code dans votre fichier **ex1php.php** et aller vérifier l'affichage dans le navigateur (rq : si vous voyez apparaître des bouts de code PHP, c'est que vous avez oublié quelque chose d'essentiel : vous devez démarrer votre serveur, c'est-à-dire : lancer **MiniWebServer.exe** qui se trouve dans **P:/SERVEUR WEB/** .Vous devrez alors vous rendre avec votre navigateur à l'URL <http://votre-adresse-IP/ex1php.php> (vous pouvez aussi taper <http://localhost/ex1php.php>). Si tout c'est bien passé, vous devez voir afficher **bonjour** et rien d'autre autour.

2/ faire des calculs avec PHP

Bon, l'intérêt d'utiliser **PHP** est assez obscur sur l'exemple précédent. C'est parce qu'on ne faisait que de l'affichage. Faisons un peu de calcul maintenant : nous allons générer deux nombres aléatoirement (c'est-à-dire au hasard) et en faire la multiplication.

Il existe une instruction en PHP permettant de générer un nombre aléatoire :

```
$i=rand(1,5); // la variable $i sera un entier compris entre 1 et 5
```

Entrezle code ci-dessous(en gras ce qu'il y a à compléter dans le fichier ex1php.php) :

```
<?php  
echo " <H2> bonjour </H2> " ;  
$i=rand(1,5);  
echo $i ;  
?>
```

Vous vous apercevez qu'à chaque fois que vous recharger la page, un nombre aléatoire compris entre un et cinq est affiché.

Vous pouvez améliorer en ajoutant un peu de texte :

```
<?php  
echo " <H2> bonjour </H2> " ;  
$i=rand(1,5);  
echo "<h3>voila un nombre au hasard : </h3>".$i ;  
?>
```

Rq : n'oubliez pas le « . » entre le texte et la variable \$i. C'est ce point qui va permettre d'assembler (on dit « concaténer » en informatique) les deux éléments.

Exercice : Faire un programme PHP qui affiche 2 nombres entre 1 et 9 aléatoirement, et affiche le résultat de la multiplication des 2 !

(rq : pour effectuer le produit de 2 variables, il suffit de faire **\$i * \$j** ;. Si vous voulez créer une troisième variable **\$k** qui vaut le produit des deux, il suffit de faire **\$k = \$i*\$j**)

3/ Passer des paramètres d'un fichier à un autre

Notre fichier PHP travail tout seul. Bien évidemment, l'intérêt de PHP pour le WEB est de pouvoir recevoir des données d'autre chose que de lui-même ! C'est ce que vous faites lorsque vous remplissez un formulaire sur internet : le contenu du formulaire est envoyé au programme PHP ! C'est ce que nous allons regarder ici : comment passez des paramètres au fichier PHP. Nous verrons qu'il existe deux manières : soit par formulaire, soit directement par l'URL(de cette manière : <http://site.com/result.php?parametre1=12>)

3-1 : passage de paramètre par l'URL

Nous allons créer une **seconde page** PHP que nous nommerons **result.php** , que nous sauvegarderons toujours au même endroit(**P:/SERVEUR WEB/_web.zmwsc**), et qui contiendra pour débiter le code suivant (n'hésitez jamais à utiliser les copier/coller ...) :

```
<HTML>
<HEAD>
</HEAD>
<BODY>
<H1>RESULTAT :</H1>

<?php
?>

</BODY>
</HTML>
```

Nous allons faire pour que **result.php** affiche votre nom que vous lui aurez transmis :

Dans l'espace PHP de **result.php**, on va écrire le code de « récupération » du paramètre puis son affichage :

```
$a = $_GET['nom']; // la variable a va prendre la valeur de la valeur de nom dans l'url
echo $a ;
```

pour passer le paramètre, il suffira de demander à votre navigateur d'aller ici :

<http://localhost:8080/result.php?nom=votrenom>

2 informations de plus :

. Pour passer 2 paramètres, il faut utiliser le caractère « & »

Par exemple : <http://localhost:8080/result.php?nom=riss&prenom=lionel>

(il faudra alors récupérer les deux paramètres de la manière suivante :

```
$a = $_GET['nom'];
$b = $_GET['prenom'];
```

. Vous pouvez bien entendu créer un lien dans un fichier plutôt que de taper les paramètres directement dans la fenêtre d'URL du navigateur. Par exemple :

```
<a href ="result.php ?nom=martin&prenom=jacques"> vers page result</a>
```

- ➔ Grace à ces deux informations, **modifier votre pageex1php.php** pour y ajouter un lien qui transmette votre nom et votre prénom à **result.php**

Pour aller plus loin :

Nous allons reprendre notre fichier, et allons en faire un programme pour aider notre petit frère qui est en CE2 à apprendre ses tables de multiplication !

Ce que nous souhaitons faire :

. que la page **1exphp.php** génère deux nombres aléatoires et les affichent à l'écran (normalement, c'est déjà le cas) mais qui n'affiche pas le résultat de la multiplication !!

. Le résultat devra apparaitre lorsqu'on cliquera sur un lien « résultat » qui pointerà sur la page **result.php** et transmettra les deux nombres (on ne va plus envoyer nos noms et prénoms mais les deux nombres aléatoires !)

➔ A vous de réaliser cela :o) ((si vous êtes en retard, demandez moi le code ...)

3-2 : passage de paramètre par formulaire

Notre petit programme précédent est très bien mais ne permet pas de prendre en compte la proposition de résultat de notre petit frère ! Nous allons donc lui proposer de le saisir au clavier. Pour cela, nous allons utiliser un formulaire :

votre résultat :

Voilà le code HTML du formulaire ci-dessus, qui envoie les informations à **result.php** :

```
<form action="result.php" method="post">// on déclare notre formulaire
votre résultat :<br>
<input type="text" name="proposition" />// entrée de formulaire de type texte
<input type="submit" value="Postez"/>// bouton d'envoi du formulaire
</form>
```

Ici, nous transmettons juste la valeur "proposition" (exactement comme si nous faisons un lien comme ceci : <http://localhost/result.php?proposition=...>)

Il est possible aussi de transmettre par formulaire des champs « cachés » qui n'apparaîtront pas visiblement sur la page. Ce sont les champs de type « hidden »

exemple : `<input type="hidden" name="nombre1" value = 5 />`

transmettra la variable nommée nombre1 qui vaudra 5 " (exactement comme si nous faisons un lien comme ceci : <http://localhost/result.php?nombre1=5&proposition=...>)

Pour récupérer la ou les variables dans **result.php**, il faudra juste écrire dans **result.php** ceci :

```
$p = $_POST['proposition'];// la variable $p prendra alors la valeur du paramètre transmis en text
$i = $_POST['nombre1'];// la variable $i prendra alors la valeur du paramètre transmis en hidden
```

Votre travail après avoir **lu et compris** les explications précédentes :

- ➔ Modifier **ex1.php.php** pour qu'il envoie la proposition de résultat du petit frère ainsi que la valeur des deux nombres aléatoires à **result.php** par formulaire.
- ➔ Modifier **result.php** pour qu'il calcul et affiche le résultat et la proposition du petit frère.

4 - Verifier que le résultat est correct par un test (IF ... ELSE. ...) :

Pour l'instant, c'est notre petit frère tout seul qui vérifie que son résultat est correct. Nous allons modifier **result.php** pour qu'il puisse afficher « bravo » si le résultat est correct et « tu t'es trompé » si le résultat est faux.

Nous allons utiliser une structure de base en informatique qui consiste à faire un test. On teste une condition, si celle-ci est vraie, on fait une action, si elle est fautive, on fait une autre action. Cela donne ceci :

```
if (expression) {
instructions }
else {
instructions}
```

Exemple en PHP :

```
if ($i==1) {
echo "un"; }
else {echo"le nombre n'est pas un";}
```

Ce petit bout de programme fait la chose suivante : si la variable \$i est égale à un, on affiche le mot « un », sinon, on affiche la phrase « le nombre n'est pas un »

Votre travail après avoir lu et compris les explications précédentes :

- ➔ modifier **result.php** pour qu'il affiche « bravo » ou « tu t'es trompé » selon l'exactitude de la proposition.

4- Petit projet à faire : Accès par mot de passe

Ecrire une page HTML (qu'on appellera **log.html**) contenant un formulaire de saisie de mot de passe (il s'agit juste d'une page contenant un formulaire envoyant un champ text , et pointant sur un fichier **acces.php**)

Ecrire le fichier **acces.php** qui devra vérifier que le mot de passe est bien « x742b ». Si ce n'est pas le bon mot de passe, il affichera à l'écran en énorme « accès refusé ». si c'est le bon mot de passe, il affichera : « bienvenue » et affichera une image de votre choix.

6 – écriture dans un fichier (retour au petit frère)

On souhaite maintenant vérifier que notre petit frère est au point sur ses tables de multiplication, mais on ne souhaite pas vérifier son travail au fur et à mesure mais ensuite. Il va donc falloir mémoriser ses résultats ! Pour cela, nous allons les sauvegarder dans un fichier. On utilisera un simple fichier texte qu'on nommera **result.txt**.

Voilà ce qu'il faut savoir sur la manipulation des fichiers en PHP :

Manipulation de fichier en PHP :

Ouvrir un fichier :

```
entier = fopen(chaine nomdufichier, chaine mode);
```

```
$fp = fopen (« ../dudul.txt », « w » );
```

Fermer un fichier

```
fclose($fich) ;
```

ecrire dans un fichier :fputs() ou fwrite()

```
$fp = fopen("php_8_fichier.txt","a"); // ouverture du fichier en écriture
fputs($fp, "\n"); // on va a la ligne
fputs($fp, "$nom|$email"); // on écrit le nom et email dans le fichier
fclose($fp);
```

lire dans un fichier :fget() ou fread()

```
$Ligne = fgets($fp,255) ; // recupere 255 caracteres du fichier
```

Tester la fin d'un fichier :feof()

```
while(!feof($fp)) {$Ligne = fgets($fp,255);}
```

taille d'un fichier : filesize (text.txt)

```
$fp = fopen(fichier.txt, « r » ); // ouvre ne lecture fichier.txt
$out = fread ($fp , filesize (text.txt)) ; // mets dans $out tout le fichier puisqu'il s'arrête à la taille du fichier
```

Mode :	
Mode	Description
r	ouverture en lecture seulement
w	ouverture en écriture seulement (la fonction crée le fichier s'il n'existe pas)
a	ouverture en écriture seulement avec ajout du contenu à la fin du fichier (la fonction crée le fichier s'il n'existe pas)
r+	ouverture en lecture et écriture
w+	ouverture en lecture et écriture (la fonction crée le fichier s'il n'existe pas)
a+	ouverture en lecture et écriture avec ajout du contenu à la fin du fichier (la fonction crée le fichier s'il n'existe pas)