

INTÉGRER PHP DANS VOS PAGES HTML

Le code HTML est exécuté par votre navigateur (Firefox par exemple), le fichier peut se trouver n'importe où sur votre disque , vous n'avez qu'à demander l'ouverture avec le navigateur pour le visualiser.

Le code PHP ne peut pas être interprété par votre navigateur. Dans une utilisation Web, l'exécution du code PHP se déroule ainsi: lorsqu'un visiteur demande à consulter une page Web, son navigateur envoie une requête au serveur HTTP correspondant. Si la page est identifiée comme un script PHP (généralement grâce à l'extension **.php**), le serveur appelle l'interpréteur PHP qui va traiter et générer le code final de la page (constitué généralement d'HTML ou de XHTML). Ce contenu est renvoyé au serveur HTTP, qui l'envoie finalement au client.



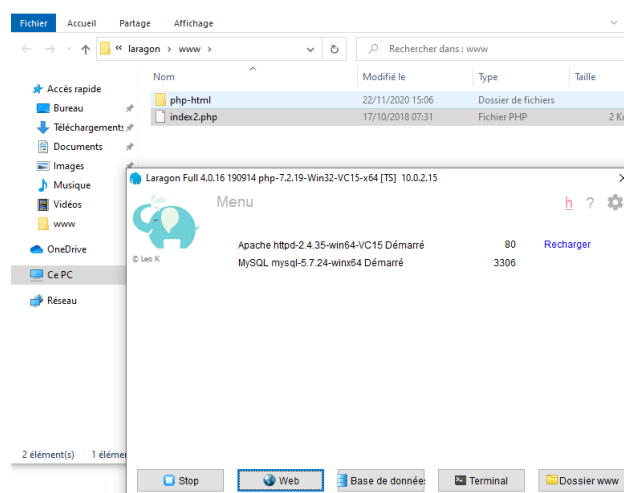
Par conséquent , si vous n'avez pas de serveur HTTP sur votre poste, vous ne pourrez pas interpréter de script PHP.

Au lycée , vous disposez de Laragon sur lequel vous avez un serveur APACHE et un serveur MySQL, mais pour qu'un script PHP soit interprété, il faut :

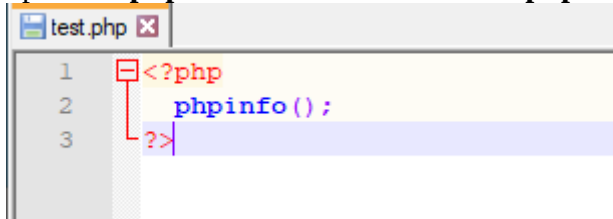
- qu'il soit placé dans le répertoire **www** de Laragon
- avec l'extension **.php**

PARTIE 1 : TEST DE FONCTIONNEMENT

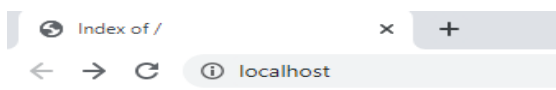
Dans le répertoire **www** renommer le fichier **index.php** en **index2.php** et créer un répertoire **php-html** dans lequel vous déposerez vos premiers fichiers php. Démarrer les services.



Dans le répertoire **php-html** créer un fichier **test.php** :

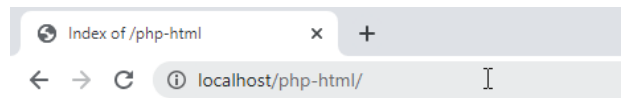


Pour exécuter ce fichier, cliquer sur le menu « **web** » ou alors dans un navigateur taper l'adresse « **localhost** » vous allez pouvoir naviguer ensuite dans l'arborescence de votre répertoire « **www** »



Index of /

- [index2.php](#)
- [php-html/](#)



Index of /php-html

- [Parent Directory](#)
- [test.php](#)

Refaire la même opération avec le script suivant test2.php :

```
<html>
<head>
</head>
<body>
  <p>Mon texte dans la page html</p>

  <?php
    echo "<h1>Bonjour le monde, c'est mon premier script PHP!</h1>";
  ?>

</body>
</html>
```

Tout fonctionne, dans votre navigateur regarder le code source de la page : edition -->code source.
Que constatez-vous?

Travail à faire:

Pour chacune des pages suivantes, donnez le résultat produit en sortie du préprocesseur Php (le résultat Html généré) après que celui ci aura réalisé son traitement

- **page1.php**

```
<html>
  <body>
    <?php
      echo "<h1> Bonjour </h1>";
    ?>
  </body>
</html>
```

- **page1.html**

```
<html>
  <body>
    <?php
      echo "<h1> Bonjour </h1>";
    ?>
  </body>
</html>
```

- **page2.php**

```
<!DOCTYPE html>
<html>
<?php
  $couleur="pink";
?>
<body style="background-color: <?php echo $couleur ;?> ;">
</body>
</html>
```

Questions:

1. A quoi sert l'extension php?
2. A quel endroit est exécuté le code PHP (serveur distant, poste à partir duquel on fait une requête http?)

PARTIE 2 : PHP – HTML : LA RÉCUPÉRATION DES PARAMÈTRES

A PARAMÈTRES PASSÉS DANS L'URL

Pour modifier la couleur de la page2.php , il est possible de transmettre le paramètre à l'aide de l'URL.

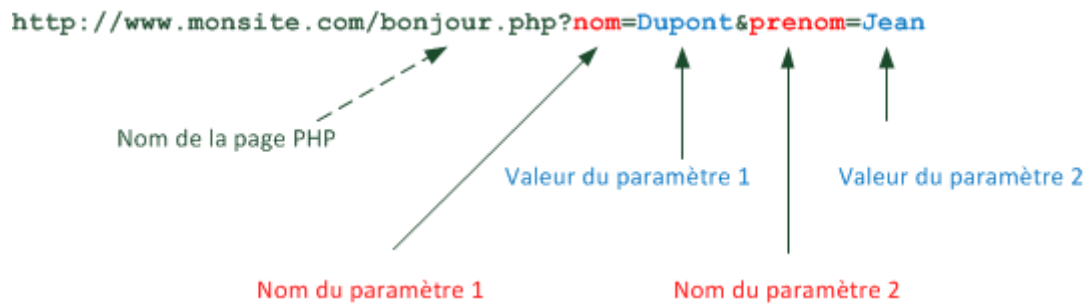
On va ajouter à la fin de l'URL « ?couleur=blue
<http://localhost/~stephane/page2.php?couleur=blue>

Le point d'interrogation "?" dit au navigateur que les paramètres suivants sont des variables. Il faudra modifier votre code PHP pour récupérer ces paramètres.

```
<?php
  $couleur=$_GET['couleur'];
?>
```

On peut retrouver les paramètres envoyés dans l'URL grâce à un **tableau associatif** nommé **\$_GET**, on obtient la valeur du paramètre en indiquant sa clé.
Ainsi, en modifiant la valeur de couleur dans l'URL, vous modifiez la présentation de la page

voir : <http://openclassrooms.com/courses/concevez-votre-site-web-avec-php-et-mysql/transmettre-des-donnees-avec-l-url>



B PARAMÈTRES PASSÉS À L'AIDE D'UN FORMULAIRE

Pour passer des paramètres dans l'URL, on peut également utiliser des formulaires :

exemple : form1.html

```
<form action='page2.php' method='get'>
  <p>Nombre de ligne: <input type="text" name="couleur" /></p>
  <input type="submit" value="OK" />
</form>
```

Nous avons une zone de saisie , et un bouton. Lorsque l'utilisateur va « cliquer » sur le bouton OK, le navigateur va construire l'URL suivante :
.....page2.php?couleur=couleurSaisie

Le champ `action='page2.php'`, indique le nom du fichier à afficher (si ce fichier se trouve dans un autre répertoire par exemple le répertoire dessin vous devez indiquer `action='dessin/page2.php'`), le champ « `name` » indique le nom du paramètre à transmettre qui va prendre comme valeur la valeur saisie par l'utilisateur.

MÉTHODE GET ET POST

Dans un formulaire vous devez indiquer, la méthode utilisée pour envoyer vos données, il existe 2 méthodes :

La méthode GET (celle qui est utilisée par défaut si rien n'est renseigné) fait circuler les informations du formulaire en clair dans la barre d'adresse en suivant le format suivant :
Exemple d'url créée à partir de la méthode GET d'un formulaire

`http://www.unsite.com/chemin/scriptphp.php?var1=valeur1`

Pour récupérer les variables vous pouvez utiliser `$_REQUEST['var1']` ou `$_GET['var1']`

La méthode POST, quant à elle, transmet les informations du formulaire de manière masquée mais non cryptée. Le fait de ne pas afficher les données dans l'URL ne signifie en

rien qu'elles sont sécurisées. Elle est préférée lorsqu'il y a un nombre important de données à transmettre ou bien lorsqu'il faut envoyer des données sensibles comme des mots de passe. Pour récupérer les variables vous pouvez utiliser \$_REQUEST['var1'] ou \$_POST['var1']

Tester les exemples du cours sur poste (attention vous devez indiquer le nom des couleurs en anglais)

Remplacer le champ input par une liste déroulante :

Couleur :

Remplacer le champ input par des boutons radios :

Couleur :

☐ Bleu ☐ Rouge ☐ Vert

Travail à faire :

Tester et essayer de comprendre les scripts suivants :

```
test2.php
<?php
    for($i=1;$i<=10;$i++)
        echo '<p>on a fait '.$i.' tour(s) dans la boucle</p>';
?>
```

```
pays.php
<html>
<head>
</head>
<body>
<table border=1>
<?php
    echo '<tr>';
    $pays=array( 'France', 'Italie', 'Espagne', 'Allemagne' );
    foreach($pays as $nom){
        echo '<th>'.$nom.'</th>';
    }
    echo '</tr>';
?>
</body>
</html>
```

--> modifier le script de pays.php, pour afficher le tableau suivant :

Autriche	Allemagne	Danemark	Espagne	France	Grèce	Irlande	Royaume-Uni
4.9	9.3	4.8	9.4	9.7	9.9	4.3	4.7

Pour vous aider :

voir le cours : https://openweb.eu.org/articles/tableaux_css

Exercice 2 :

Tester et essayer de comprendre les scripts suivants :

damier1.php

```
<table border=1 width=600>
<?php
echo '<tr>';
for ($i=0;$i<10; $i++){
    if($i%2 == 0)
        echo '<td>&nbsp;</td>';
    else
        echo '<td bgcolor=black>&nbsp;</td>';
    }
echo '</tr>';
?>
</table>
```

damier2.php

```
<table border=1 width=60>
<?php

for ($i=0;$i<10; $i++){
    echo '<tr>';

    if($i%2 == 0)
        echo '<td>&nbsp;</td>';
    else
        echo '<td bgcolor=black>&nbsp;</td>';

    echo '</tr>';
}

?>
</table>
```

--> A l'aide de ces 2 scripts, créer un programme damier.php, qui va afficher un damier de 10 lignes et 10 colonnes , une case blanche, une case noire etc..