



Suite du PHP

420-JQA-JQ

Je sais afficher, je veux maintenant boucler!

- Il existe quatre types de boucles en PHP:
 - While
 - Do...While
 - For
 - Foreach

```
<?php
$compteur = 1;

while ($compteur < 5)
{
    echo "Le compteur est rendu à ". $compteur."<br/>";
    $compteur++;
}
echo "Changeons de boucle! <br/>";
do
{
    echo "Le compteur est rendu à ". $compteur."<br/>";
    $compteur++;
}while($compteur < 5);
```

Le compteur est rendu à 1!
Le compteur est rendu à 2!
Le compteur est rendu à 3!
Le compteur est rendu à 4!
Changeons de boucle!
Le compteur est rendu à 5!

Je sais afficher, je veux maintenant boucler!

```
<?php

$couleurs = array("magenta", "cyan", "jaune");

for($i = 0; $i < count($couleurs); $i++)
{
    echo $couleurs[$i]."<br/>";
}
echo "Changeons de boucle! <br/>";
foreach ($couleurs as $couleur)
{
    echo $couleur."<br/>";
}
```

magenta
cyan
jaune
Changeons de boucle!
magenta
cyan
jaune

Des variables c'est bien, un tableau c'est mieux

- Pour déclarer un tableau vide en PHP :

```
$tab = array();  
$tab = [];  
$tab = (array) null;
```

- Pour déclarer un tableau ayant des valeurs :

```
$tab = array(1,2,3,4,5);  
$tab = [1,2,3,4,5];
```

- Pour déclarer un tableau associatif:

```
$tab = array("un" => 1, "deux" => 2, "trois" => 3);  
$tab = ["un" => 1, "deux" => 2, "trois" => 3];
```

- Pour compter le nombre d'entrées dans un tableau:

```
echo count($tab);
```

Secret sur les tableaux.... Chuuuut!

- Ajouter des éléments à la fin d'un tableau :

```
$couleurs = array("magenta", "cyan", "jaune");  
$couleurs[] = "brun";  
array_push( &array: $couleurs, var: "rose");  
for($i = 0; $i < count($couleurs); $i++)  
{  
    echo $couleurs[$i]."<br/>";  
}
```

magenta
cyan
jaune
brun
rose

- Pour boucler dans un tableau associatif :

```
$autos = array("autoCool"=>"Bugatti", "autoOk"=>"Honda", "autoEurk"=>"Peugeot");  
  
foreach ($autos as $auto => $marque)  
{  
    echo "Clé : ".$auto." Valeur : ".$marque."<br/>";  
}
```

Clé : autoCool Valeur : Bugatti
Clé : autoOk Valeur : Honda
Clé : autoEurk Valeur : Peugeot

Je suis tanné de me répéter...

- Grâce à PHP, il vous est possible de réutiliser des pages de code déjà faites!
- Pour ce faire, il faut inclure les pages les unes dans les autres :
 - **Include** : va produire un avertissement si le fichier n'a pas pu être chargé ;
 - **Require** : va arrêter le script si le fichier n'a pas pu être chargé.

Je suis tanné de me répéter... (exemple)

```
<div id="header">
  <?php
    echo "Ceci provient du fichier 'header.php'";
  ?>
</div>
```

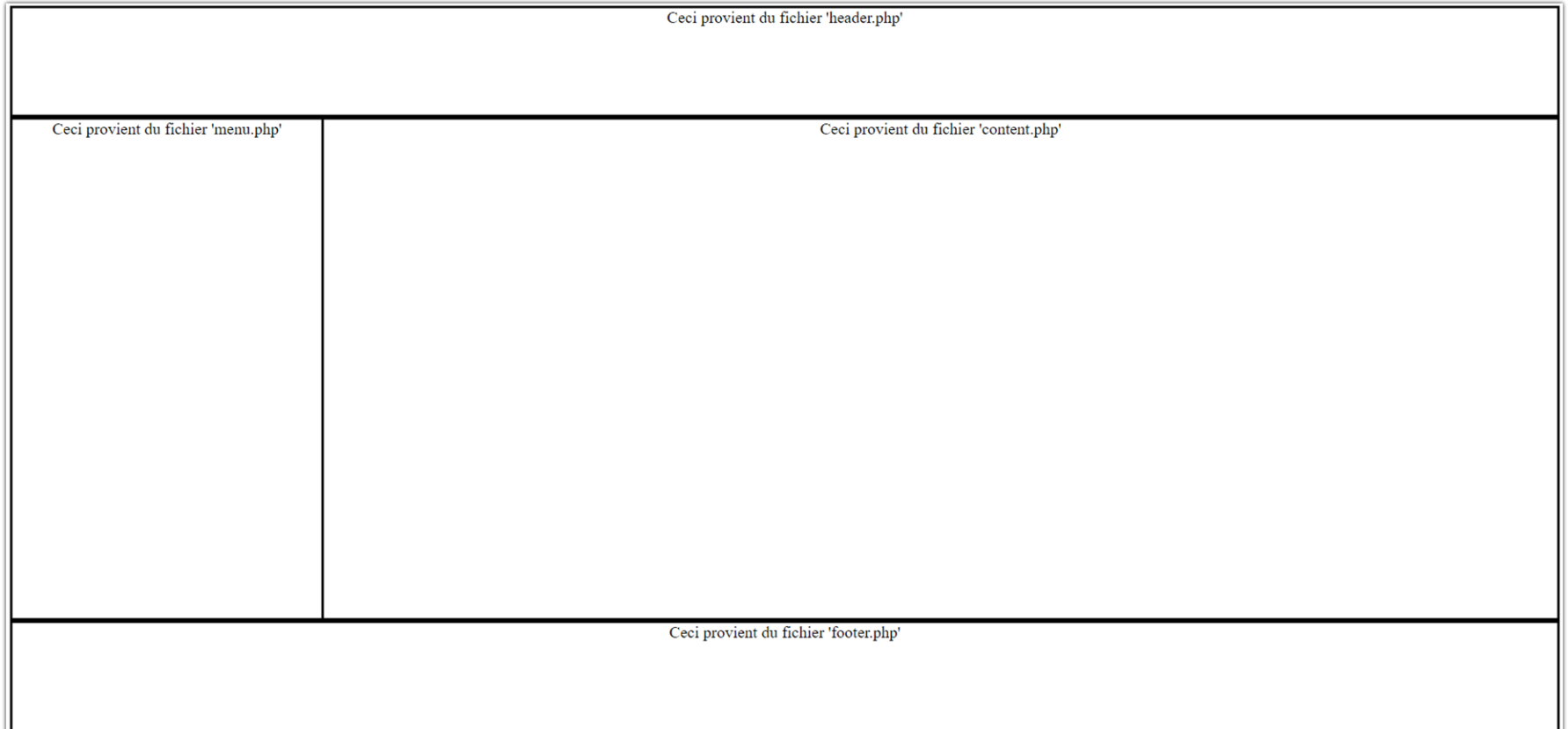
```
<div id="menu">
  <?php
    echo "Ceci provient du fichier 'menu.php'";
  ?>
</div>
```

```
<div id="content">
  <?php
    echo "Ceci provient du fichier 'content.php'";
  ?>
</div>
```

```
<div id="footer">
  <?php
    echo "Ceci provient du fichier 'footer.php'";
  ?>
</div>
```

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Include EVERYTHING</title>
    <link rel="stylesheet" type="text/css" href="Style.css"/>
  </head>
  <body>
    <?php
      include 'header.php';
      include 'menu.php';
      require 'content.php';
      include 'footer.php'
    ?>
  </body>
</html>
```

MAGIE!



Rendre Cookie Monster heureux

- Il vous est possible de conserver des données dans un petit fichier sur la machine de l'utilisateur. Ce petit fichier est accédé par le navigateur lorsque la page est chargée ;
- Ce sont les COOKIE ;
- La déclaration d'un COOKIE doit TOUJOURS être faite avant l'étiquette "<html>" ;
- **setcookie(name, value, expire);**
 - Name: le nom du COOKIE ;
 - Value: sa valeur;
 - Expire: La date, en secondes, à laquelle il expire.

Création et utilisation de COOKIE

```
<?php
$cookie_name = "utilisateur";
$cookie_value = "Cookie Monster";
/* Date d'aujourd'hui en seconde +
 * Nombre de seconde dans 1 journée *
 * 30 jours
 */
$expire = time() + 86400*30;
setcookie($cookie_name, $cookie_value, $expire);
```

```
if(isset($_COOKIE["utilisateur"]))
{
    echo "Bienvenu cher ".$_COOKIE["utilisateur"];

    /*Faisons expirer le COOKIE
    * Date et heure d'aujourd'hui
    * Il y a une heure
    */
    setcookie($cookie_name, value: "", expire: time() - 3600);
}
```

Moins lourd qu'un cookie?

- Il est possible de garder de l'information à travers plusieurs pages sans avoir à la stocker sur l'ordinateur de l'utilisateur: il s'agit de la variable SESSION ;
- Elle existe à travers toutes les pages d'une application, et ce jusqu'à ce que l'utilisateur ferme la page web ;
- Pour démarrer une session, il faut mettre la ligne suivante au tout début du code, avant n'importe quelle balise HTML: `session_start();`

101 de comment réussir sa session

```
<?php
// Fichier sessionMaker.php
session_start();

$_SESSION["couleurDeFond"] = "red";

echo "La session est prête!";
```

La session est prête!

```
<?php
//fichier sessionUser.php
session_start();

echo $_SESSION["couleurDeFond"];
```

red

```
<?php
//fichier sessionDeleter.php
session_start();

session_unset();

session_destroy();

echo "Ta session est foutue...";
```

Ta session est foutue...

Du PHP sans changer de page, c'est possible!

- Si vous avez besoin de questionner un serveur WEB pour avoir des données, mais que vous ne voulez pas rafraîchir ou changer de page, vous pouvez utiliser du [JavaScript](#) pour faire des requêtes HTTP ;
- Pour avoir les données, elles doivent être affichées dans un fichier (XML, HTML, résultats d'un **ECHO** en PHP, etc.)

Du PHP sans changer de page, c'est possible!

```
//Créer l'objet qui permet de faire
//Des requêtes HTTP
var xhttp = new XMLHttpRequest();

//Déterminer l'action à faire une
//Fois la requête exécutée
xhttp.onreadystatechange = function(){
    // Vérifier que la requête a fonctionné
    if(this.readyState === 4 && this.status === 200){
        //Afficher le contenu de la réponse
        document.getElementById("demo").innerHTML = xhttp.responseText;
    }
};

//Préparer la requête (le bool est pour synchrone ou non)
xhttp.open("GET", "exemple.php?id=1", true)
//Envoyer la requête
xhttp.send();
```