Introduction à PHP

Attribution - Partage dans les Mêmes Conditions :
http://creativecommons.org/licenses/by-sa/3.0/fr/

Table des matières

I Contexte

II Rappel : XHTML

Corps

Exemple

```
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title>Exemple de fichier XHTML</title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    </head>
    <body>
        <p>Hello world !</p>
    </body>
</html>
```

HTML

Définition

HTML p.23 est un langage inventé à partir de 1989 pour coder des pages de contenu sur le Web. Il est standardisé par le *W3C* p.23.

Langage à balises

Définition

HTML est un langage à balises : il se fonde sur le mélange entre du contenu et des balises permettant de caractériser ce contenu. HTML utilise le formalisme *SGML* p.23 pour définir les balises et combinaisons de balises autorisées.

Extrait de code HTML

Exemple

```
<p>Ceci est un contenu, caractérisé par des <b>balises</b></p>
```

Les balises p et b ont une signification dans le langage HTML : Créer un paragraphe et mettre en gras.

Structure générale

Syntaxe

```
<html xmlns="http://www.w3.org/1999/xhtml">
    <head> ... </head>
    <body> ... </body>
</html>
```

Entête

Syntaxe

```
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title>...</title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    </head>
    <body> ... </body>
</html>
```

Corps

Syntaxe

```
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title>...</title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    </head>
    <body>
        <h1>...</h1>
        <h2>...</h2>
        <p>...</p>
    </body>
</html>
```

Syntaxe

```
<p>Un paragraphe de texte</p>
<p>Paragraphe contenant du texte, mot <b>gras</g> ou <i>italique</i>.</p>
<p><a href="page02.html">Un lien</a> vers une autre page</p>
<img src="img.png" alt="une image"/>
<h1>Titre de niveau 1</h1>
<h2>Titre de niveau 2</h2>
<h3>Titre de niveau 3</h3>
<table border="1">
 <tr><th>Titre colonne 1  </th><th>Titre colonne 2  </th><th>...</th></tr>
 <tr><td>Ligne 1 colonne 1</td><td>Ligne 1 colonne 2</td><td>...</td></tr>
 <tr><td>Ligne 2 colonne 1</td><td>Ligne 2 colonne 2</td><td>...</td></tr>
</table>
<ul>
 <li>Item de liste à puce</li>
 <li>Item de liste à puce</li>
</ul>
<ol>
 <li>Item de liste à ordonnée</li>
 <li>Item de liste à ordonnée</li>
</ol>
```

Complément

developer.mozilla.org : Les bases du HTMLL[1]

1. https://developer.mozilla.org/fr/docs/Apprendre/Commencer_avec_le_web/Les_bases_HTML

III Exercice : Appliquer la notion : Population

Soit la page HTML suivante, visualisée sous le navigateur web Firefox.

Population par département

Numéro	Nom	Population
01	Ain	529378
02	Aisne	552320
03	Allier	357110
04	Alpes-de-Haute-Provence	144809
05	Hautes-Alpes	126636
06	Alpes-Maritimes	1022710
07	Ardèche	294522
08	Ardennes	299166
09	Ariège	142834
10	Aube	301388

- Département le plus peuplé : **1022710**
- Département le moins peuplé : **126636**

Image 1 Page HTML visualisée avec un navigateur Web

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Exercice</title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
</head>
<body>
  <h1>Population par département</h1>
  <table border="1">
    <tr><th>Numéro</th><th>Nom</th><th>Population</th></tr>
    <tr><td>01</td><td>Ain</td><td>529378</td></tr>
    <tr><td>02</td><td>Aisne</td><td>552320</td></tr>
    <tr><td>...</td><td>...</td><td>...</td></tr>
  </table>
  <ul>
    <li>Département le plus peuplé : <b>Paris</b> (<i>2147857</i>)</li>
    <li>Département le moins peuplé : <b>Hautes-Alpes</b> (<i>126636</i>)</li>
  </ul>
</body>
</html>
```

Question 1

[solution n°1 p. 21]

Créez le fichier HTML source de cette page sur votre ordinateur en utilisant un éditeur de texte, comme `notepad++` sous Windows ou `gedit` sous Linux.

Question 2

[solution n°2 p. 21]

Déposez le fichier sur un VPS disposant d'un serveur installé afin qu'il soit accessible à l'adresse `http://mon.vps.org/samples/pop.html`.

IV Présentation et installation d'un serveur PHP

PHP est un langage interprété (un langage de script) exécuté du côté serveur (comme les scripts CGI, ASP, ...) et non du côté client (un script écrit en JavaScript ou une applet Java s'exécute au contraire sur l'ordinateur où se trouve le navigateur). La syntaxe du langage provient de celles du langage C, du Perl et de Java.

Ses principaux atouts sont :

- La gratuité et la disponibilité du code source (PHP est distribué sous licence GNU GPL)
- La simplicité d'écriture de scripts
- La possibilité d'inclure le script PHP au sein d'une page HTML (contrairement aux scripts CGi, pour lesquels il faut écrire des lignes de code pour afficher chaque ligne en langage HTML)
- La simplicité d'interfaçage avec des bases de données (de nombreux SGBD sont supportés, le plus utilisé avec ce langage est MySQL).
- L'intégration au sein de nombreux serveurs web (Apache...)

SGBD supportés par PHP Exemple

- MySQL
- Oracle
- PostgreSQL
- ...

V Exercice : Appliquer la notion

Question

[solution n°3 p. 21]

Créez un fichier `test.php` sur un VPS disposant d'un serveur PHP utilisant la fonction `phpinfo` pour renvoyer les informations techniques sur le module PHP installé sur le serveur.

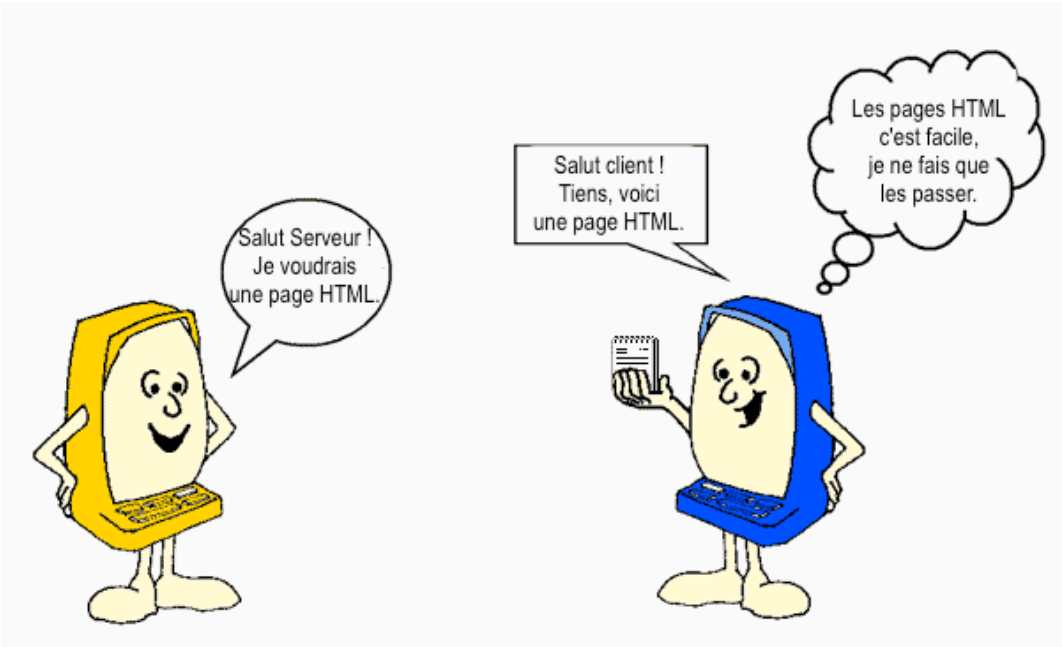
VI Fonctionnement d'un serveur PHP

L'interprétation du code par le serveur

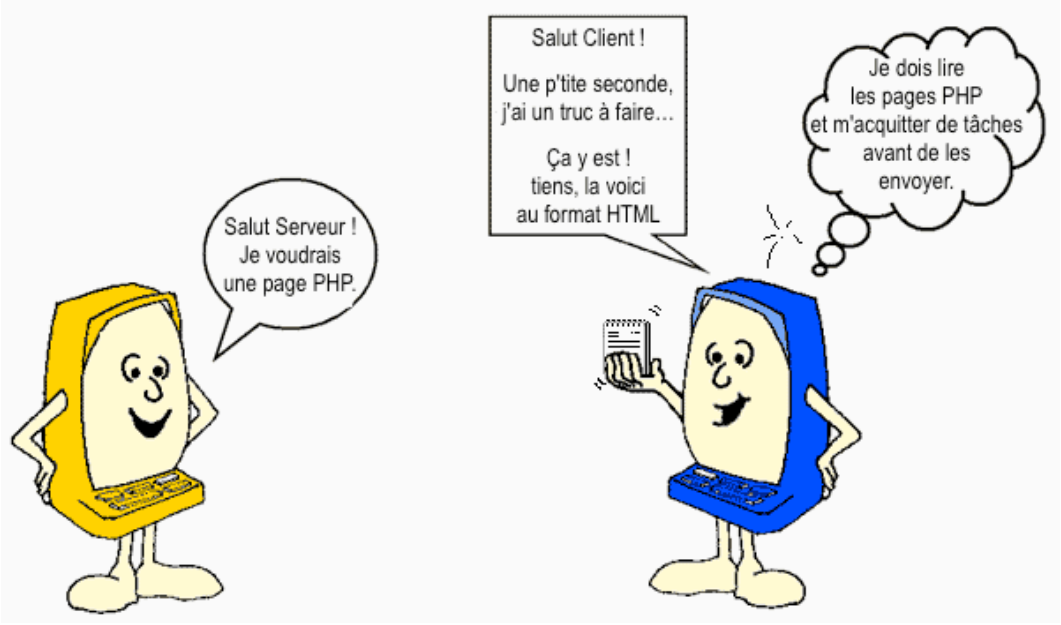
Un script PHP est un simple fichier texte contenant des instructions écrites à l'aide de caractères ASCII 7 bits (des caractères non accentués) incluses dans un code HTML à l'aide de balises spéciales et stocké sur le serveur. Ce fichier doit avoir une extension particulière (qui dépend de la configuration du serveur HTTP, en général ".php") pour pouvoir être interprété par le serveur.

Ainsi, lorsqu'un navigateur (le client) désire accéder à une page dynamique réalisée en php :

1. Le serveur reconnaît qu'il s'agit d'un fichier PHP
2. Il lit le fichier PHP
3. Dès que le serveur rencontre une balise indiquant que les lignes suivantes sont du code PHP, il "passe" en mode PHP, ce qui signifie qu'il ne lit plus les instructions: il les exécute.
4. Lorsque le serveur rencontre une instruction, il la transmet à l'interpréteur
5. L'interpréteur exécute l'instruction puis envoie les sorties éventuelles au serveur
6. A la fin du script, le serveur transmet le résultat au client (le navigateur)



Requête d'une page HTML (http://fr.html.net/tutorials/php/lesson1.php)



Requête d'une page PHP (http://fr.html.net/tutorials/php/lesson1.php)

Code PHP et clients Web

Remarque

Un script PHP est interprété par le serveur, les utilisateurs ne peuvent donc pas voir le code source.

Le code PHP stocké sur le serveur n'est donc jamais visible directement par le client puisque dès qu'il en demande l'accès, le serveur l'interprète ! De cette façon aucune modification n'est à apporter sur les navigateurs...

Hello world

Exemple

```php
<?php
echo "Hello world";
?>
```

Syntaxe

Pour que le script soit interprété par le serveur deux conditions sont nécessaires :

- Le fichier contenant le code doit avoir l'extension .php et non .html (selon la configuration du serveur Web)
- Le code PHP contenu dans le code HTML doit être délimité par les balises "<?php" et "?>"

Complément

Tutoriel PHP : http://fr.html.net/tutorials/php/

VII Exercice

Question

[solution n°4 p. 21]

Créer un fichier `hello.php` permettant d'afficher le texte *Hello World !*.

```php
<?php
  echo "Hello world !";
?>
```

VIII Envoi de texte au navigateur par le serveur PHP

Syntaxe

```
1 echo Expression;
```

Print

Remarque

La fonction `print` est iso-fonctionnelle avec `echo` et `printf` plus complexe permet en plus le formatage des données (peu utilisée).

IX Implantation du code PHP au sein du code HTML

L'importance de l'implantation du code PHP au sein du code HTML

Fondamental

Le code PHP peut être implanté au sein du code HTML. Cette caractéristique n'est pas à négliger car le fait d'écrire uniquement du code PHP là où il est nécessaire rend la programmation plus simple (il est plus simple d'écrire du code HTML que des fonctions `echo` ou `print`, dans lesquelles les caractères spéciaux doivent être précédés d'un antislash sous peine de voir des erreurs lors de l'exécution).

L'exemple le plus simple concerne les pages dynamiques dont l'en-tête est toujours le même: dans ce cas, le code PHP peut ne commencer qu'à partir de la balise `<body>`, au moment où la page peut s'afficher différemment selon une variable par exemple.

Mieux, il est possible d'écrire plusieurs portions de script en PHP, séparées par du code HTML statique car les variables/fonctions déclarées dans une portion de script seront accessibles dans les portions de scripts inférieures.

Hello world

Exemple

```
1 <html>
2 <head><title>Exemple</title></head>
3 <body>
4 <?php
5 echo "Hello world";
6 ?>
7 </body>
8 </html>
```

X Exercice

Question

[solution n°5 p. 21]

Transformer le fichier PHP suivant pour qu'il renvoie une page HTML.

```php
<?php
  echo "Hello world !";
?>
```

XI Syntaxe PHP

Manuel PHP en ligne — Fondamental

http://php.net/manual/

Exemple

```php
<?php
$i=0 ;
while($i<6) {
  echo $i ;
  $i=rand(1,6) ;
}
?>
```

Généralités — Attention

- Une instruction se termine par un `;`
- Les espaces, retours chariot et tabulation ne sont pas pris en compte par l'interpréteur
- Les commentaires sont écrits entre les délimiteurs `/*` et `*/` ou `//` sur une seule ligne.
- Le langage est *case-sensitive* (sauf pour les fonctions).

IDE — Complément

- Eclipse PDT (PHP Development Tools)
  http://www.zend.com/fr/community/pdt
- Zend Studio
  http://www.zend.com/fr/products/studio/

XII Variables en PHP

- Les variables ne sont pas déclarées
- Les variables commencent pas un $
- Les variables ne sont pas typées

Les variables en langage PHP peuvent être de trois types :

- Scalaires (entiers, chaîne, réels)
- Tableaux (un tableau pouvant être multidimensionnel et stocker des scalaires de types différents)
- Tableaux associatifs (indexés par des chaînes)

Exemple

```php
$Entier=1;
$Reel=1.0;
$Chaine="1";
$Tableau[0]=1
$Tableau[1]="1"
$TableauMulti[0][0]="1.0"
$TableauAssoc[Age]=18
```

isset()

Complément

```php
if (isset($var)) {
 echo $var;
}
```

La fonction `isset()` permet de tester qu'une variable existe et est affectée.

Complément

Formulaires HTML et PHP (cf. p.25)

XIII Structures de contrôle en PHP

Alternative IF

Syntaxe

```php
if (condition réalisée) {
 liste d'instructions
}
elseif (autre condition réalisée) {
 autre série d'instructions
}
 ...
else (dernière condition réalisée) {
 série d'instructions
}
```

Boucle FOR

Syntaxe

```php
for (compteur; condition; modification du compteur) {
 liste d'instructions
}
```

Boucle WHILE

Syntaxe

```php
while (condition réalisée) {
 liste d'instructions
}
```

Autres structures de contrôle

Complément

http://php.net/manual/fr/language.control-structures.php

XIV Exercice : Deux fois deux

Question

[solution n°6 p. 21]

Écrire un programme PHP qui permet d'afficher la table de multiplication des 2 en HTML.

Indice :

Penser à implanter le code PHP dans le code HTML

Indice :

On utilisera une boucle *for*.

XV Auto-évaluation

1. Exercice final

Exercice 1 : Quiz - Culture

[solution n°7 p. 21]

Exercice

...

A A

B B

Exercice 3 : Quiz - Méthode

[solution n°8 p. 22]

Exercice

...

A A

B B

Exercice 5 : Quiz - Code

[solution n°9 p. 22]

Exercice

...

A A

B B

2. Exercice : Défi

...

Solutions des exercices

Solution n°1

[exercice p. 6]

Solution n°2

[exercice p. 7]

Solution n°3

[exercice p. 9]

Solution n°4

[exercice p. 12]

Solution n°5

[exercice p. 15]

```
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title>Hello PHP !</title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    </head>
    <body>
        <p>
            <?php
                echo 'Hello world !';
            ?>
        </p>
    </body>
</html>
```

Solution n°6

[exercice p. 19]

```
<?php
for ($i=1; $i<10; $i++) {
  echo '<p>';
  echo "2 x $i = ", 2*$i;
  echo '</p>';
}
?>
```

Solution n°7

[exercice p. 20]

Exercice

...

A A

B B

Solution n°8

[exercice p. 20]

Exercice

...

A A

B B

Solution n°9

[exercice p. 20]

Exercice

...

A A

B B

Abréviations

HTML : HyperText Markup Language
SGML : Standard Generalized Markup Language
W3C : World Wide Web Consortium

Index

Contenus annexes

1. Formulaires HTML et PHP

Rappel

Requête GET ou POST par formulaire HTML (balise <form>) *(cf. p.26)*

Traiter les requêtes HTTP avec un serveur PHP *(cf. p.27)*

Page d'appel

Exemple

```html
<html>
  <body>
     <form method="get" action="test.php">
     <input type="text" size="20" name="MaVar" />
     <input type="submit" />
   </form>
</body>
</html>
```

Page appelée (test.php)

Exemple

```php
<?php
  echo $_GET["MaVar"];
?>
```

Code implanté

Remarque

La page retournée dans l'exemple n'est pas du HTML (mais du simple texte qu'un navigateur peut néanmoins afficher). Pour retourner du HTML, il faut implanter ce même code au sein d'une page HTML.

```html
<html>
  <body>
    <p>
    <?php
      echo $_GET["MaVar"];
    ?>
    </p>
  </body>
</html>
```

Cache ⚠ Attention

Les navigateurs disposent d'un cache, c'est à dire d'une copie locale des fichiers qui leur évite de recharger plusieurs fois un fichier identique. Lorsque l'on développe une application PHP, les fichiers changent fréquemment, il est alors nécessaire de **vider le cache** pour que le navigateur recharge bien la nouvelle version du code.

Sous Firefox, faire CTRL+F5.

2. Requête GET ou POST par formulaire HTML (balise <form>)

Rappel

Un exemple de fichier XHTML *(cf. p.27)*

Formulaire Az Définition

On appelle formulaire une interface permettant à un utilisateur humaine de saisir des données en vue dans une application informatique.

Contrôle Az Définition

On appelle contrôle un élément d'un formulaire permettant d'effectuer une action : saisir une donnée, exécuter une requête...

La balise `form` du langage HTML permet de :

- créer un formulaire avec des contrôles,
- envoyer le contenu du formulaire à un serveur web grâce à une requête GET ou POST.

Contrôle en HTML Exemple

- étiquette
- cases à cocher
- champs de saisie
- boutons radio
- listes à choix multiples
- ...

Formulaire Exemple

```
<form metho="get" action="test.php">
  <p><label>Nom</label> <input type="text" name="nom"></p>
  <p><label>Prénom</label> <input type="text" name="prenom"></p>
  <p><label>Age</label> <input type="text" name="age"></p>
  <p><input type="submit"></p>
</form>
```

Complément

https://developer.mozilla.org/fr/docs/Web/HTML/Element/Form

4. Traiter les requêtes HTTP avec un serveur PHP

Rappel

Requête GET ou POST par formulaire HTML (balise <form>) *(cf. p.26)*

Lorsqu'une requête HTTP envoie des données au serveur web, par exemple grâce à un lien <a> ou un formulaire `<form>` en HTML, les données envoyées doivent être traitées par un programme que l'on écrit spécifiquement sur le serveur.

Fondamental

Un serveur web/PHP peut gérer les données envoyées par une requête HTTP.

Lors de son chargement une page PHP contient un tableau de variables pour les données envoyées par la méthode GET et un autre pour les données envoyées par POST.

Syntaxe

On accède à ses données en utilisant la syntaxe :

`$_GET["var1"]`

`$_GET["var2"]`

ou

`$_POST["var1"]`

`$_POST["var2"]`

où `var1` et `var2` sont des noms de données dans la requête HTTP (par exemple le nom des contrôles dans le formulaire HTML à l'origine de la requête).

Exemple

```php
<?php
echo 'Hello ' . $_POST["name"] ;
?>
```

Complément

http://php.net/manual/en/reserved.variables.get.php

http://php.net/manual/en/reserved.variables.post.php