

Annexe : Notions de Langage PHP

1 Généralités

Voici un exemple d'un fichier php.

```
<!DOCTYPE html>
<html lang='fr'>
  <head>
    <meta charset='utf-8' />
    <title>Exemple 1</title>
  </head>
  <body>
    <h1>Exemple 1</h1>
    Cette page est le résultat du code ci-après : <br />
     : <br />

    <?php
      echo "Premier exemple illustrant le langage <strong>php</strong>\n"
    ?>
  </body>
</html>
```

2 Les Variables

- Nom : commence toujours par \$
- Distinction entre minuscules et majuscules
- Déclaration inutile
- Fonctions très utiles :
 - `isset(<nomVar>)` : retourne TRUE si <nomVar> est définie, FALSE sinon
 - `empty(<nomVar>)` : retourne TRUE si <nomVar> est nulle, vide ou non définie, FALSE sinon
- Affectation : =
 - Numérique : `$age=20+$x;`
 - Chaîne : `$nom="Monsieur".$y;`
 - Booléen : `$marie=TRUE;`
- Comparaison :
== != < <= > >=
- Afficher le contenu d'une variable :

```
echo $nom;
echo "Je suis $nom et j'ai $age ans";
echo 'Je suis '.$nom.' et j\'ai '.$age.'ans';
echo " \$ $ \" \\";
```
- 2 façons de passer des valeurs d'une page à l'autre :
 - Utiliser l'URL de la page : `http://www.truc.fr/page.php?nom=Moi&age=20`
Le fichier `page.php` récupérera :
 - `$_GET['nom']` qui aura la valeur "Moi"
 - `$_GET['age']` qui aura la valeur 20Les valeurs sont passées par la méthode GET.
Ne pas passer de valeurs sensibles avec la méthode GET car elles sont visibles dans l'URL
 - Utiliser un formulaire

3 Structures conditionnelles

- Structure de base :

```
if (<condition>){
    <instructions_if>{
}
else { // Facultatif
    <instructions_else>{
}
```

- Opérateurs logiques ! && ||

- Structure à choix multiple :

```
switch (<nomVar>) {
    case <val1> :
        <instructions1>;
    case <val2> :
        <instructions2>;
    ...
    default: // Facultatif
        <instructions>;
}
```

Penser au **break** pour sortir d'un bloc

4 Structures itératives

- Structure while :

```
while(<condition>) {
    <instructions>;
}
```

- Structure for :

```
for (<init>;<condition>;<incr>) {
    <instructions>;
}
```

5 Tableaux

5.1 Tableaux "séquentiels"

- Indices : entiers (indice 1^{ère} case=0)
- Exemples :


```
$t[0]="BUT";
$t[1]="RT";
$t=array("BUT", "RT");
```
- Ajout d'une valeur dans la dernière case du tableau : `<nomTab>[]=<val>;`
- Taille d'un tableau : `count(<nomTab>)`
- Parcours d'un tableau :
 - Utilisation d'un for ...
 - Utilisation d'un foreach


```
foreach (<nomTableau> as <nomVar>) {
    <instructions>;
}
```

5.2 Tableaux "associatifs"

- Indices : chaînes de caractères
- Couples : clé => valeur
- Clés = valeurs uniques
- Exemples :


```
$t['B']="BUT";
$t['R']="RT";
$t=array('B'=>"BUT", 'R'=>"RT");
```
- Taille d'un tableau : `count(<nomTab>)`
- Parcours d'un tableau :


```
foreach (<nomTableau> as <cle>=><val>) {
    <instructions>;
}
```
- Autres tableaux associatifs : `$_GET`, `$_POST`, ...

6 Fonctions

- Écriture d'une fonction :

```
function <nomFct> (<listeParam>){  
    <instructions>;  
    return <varResultat>; // Facultatif  
}
```

- Appel d'une fonction qui ne retourne rien :

```
<nomFct> (<listeParam>);
```

- Appel d'une fonction qui retourne un résultat :

```
<nomVar> = <nomFct> (<listeParam>);
```

- Variables locales :

- Variables qui apparaissent pour la première fois dans une fonction
- Connues et utilisables uniquement dans la fonction

- Variables globales :

- Variables initialisées hors de toute fonction
- Connues et utilisables dans toutes les fonctions
- Valeurs stockées dans un tableau associatif : `$GLOBALS` (partie sur les variables super globales)
 - Clé = nom de la variable globale
 - Valeur = valeur de la variable globale

- La fonction `include` :

- Permet d'inclure le contenu d'un fichier `.php`
- Très utile pour :
 - Structuration de toutes les pages d'un site
 - Déclaration des fonctions, des variables globales, ...
- Syntaxe : `include(<nomFic.php>);`
 - Le nom du fichier est une chaîne de caractères, il doit donc être entre guillemets.
 - Le contenu de `<nomFic.php>` sera inséré à la place de l'instruction `include`.

- La fonction `header` :

- Permet de rediriger vers une autre page
- Utilise les en-têtes HTTP
 - ⇒ la fonction doit être appelée au tout début
- Syntaxe :

```
header('Location: <nomFic.php>');  
exit();
```
- Ne pas oublier de mettre le `exit`
- Le nom du fichier, `<nomFic.php>`, peut-être : Un chemin relatif, Un chemin absolu et Une URL (avec éventuellement passage de paramètres)

- Quelques fonctions utiles sur les chaînes :

- `strlen(<uneCh>)` retourne la longueur de la chaîne passée en paramètre
- `str_replace(<quoi>,<par>,<uneCh>)` retourne une chaîne où toutes les occurrences de `<quoi>` dans `<uneCh>` auront été remplacées par `<par>`.
- `strtolower(<uneCh>)/strtoupper(<uneCh>)` retourne une chaîne où toutes les caractères de `<uneCh>` auront été mis en minuscules/majuscules
- `explode(<sep>,<uneCh>)` explode `<uneCh>` en fonction de `<sep>` et range chaque "morceau" dans le tableau qui sera retourné
- `htmlentities(<uneCh>)` retourne une chaîne dans laquelle certains caractères de `<uneCh>` seront transformés en HTML (ex : `&` => `&` ;)

- Manipulation des dates :

- Dates manipulées sous la forme de leur **timestamp** (= nombre de secondes écoulées depuis le 01/01/1970)

- Fonction `time()` retourne le **timestamp** correspondant à la date/heure courante

- Fonction `mktime(<h>,<min>,<s>,<d>,<m>,<y>)` retourne le **timestamp** correspondant à la date spécifiée en paramètre

Fonction `date(<format>,<timestamp>)` retourne une chaîne correspondant au **<timestamp>** mis en forme suivant **<format>** qui utilise (entre autres) :

s : secondes - **i** : minutes - **H** : heure

d : jour - **m** : mois - **y** ou **Y** : année

7 Formulaires

- Informations saisies dans le formulaire HTML seront "envoyées" à un script Php.

Il y aura donc 2 fichiers :

- Un fichier `.html` (ou `.php`) contenant au moins le code HTML du formulaire.

- Un fichier `.php` pour le traitement des informations en provenance du formulaire

- Code HTML :

```
<form method='post' action='script.php'>
  <p>
    <!-- Ici les champs du formulaire -->
  </p>
</form>
```

Il faut obligatoirement un champ button de type `submit`.

- Le script `script.php` récupérera les données dans un tableau (`$_POST`).

Voici l'exemple d'un fichier HTML contenant le formulaire

```
<!DOCTYPE html>
<html lang='fr'>
  <head>
    <meta charset='utf-8' />
    <title>Formulaire</title>
  </head>

  <body>
    <h1>Exemple d'un formulaire</h1>
    Cette page est le résultat du code ci-après : <br />
     : <br />

    Ce formulaire se trouve sur la page php_ex.html. <br />
    Le prénom saisi sera ensuite envoyé au script bonjour.php. <br />
    <form method='post' action='bonjour.php'>
      <p>
        <label for='qui'>Prénom</label>
        <input type='text' name='qui' id='qui' /><br /><br />
        <button type="submit">Go</button>
      </p>
    </form>
  </body>
</html>
```

8 Variables Super-Globales

- Variables créées automatiquement par PHP au chargement d'une page .php
- Connues et accessibles dans toutes les pages du site
- Leur nom commence toujours par \$_ et est écrit en majuscules (\$_GET, \$_POST,...)
- Ce sont des tableaux associatifs :
 - \$GLOBALS : toutes les variables globales
 - \$_GET : pour passer les variables par l'URL
 - \$_POST : pour passer les variables sans passer par l'URL
 - \$_SERVER : récupère des informations du serveur
 - \$_ENV : toutes les variables d'environnement
 - \$_FILES : pour envoyer des fichiers sur le serveur
 - \$_SESSION : partager des variables sur tout le site
 - \$_COOKIE : pour stocker les cookies enregistrés