

TD 05 : Pages web et bases de données

I. HTML

Version actuelle : HTML 5.
Documentation : <http://www.w3schools.com/html>

1. Première page HTML

```
<!-- Fichier : index.html -->
<html>
  <head>
    <title>TD de BDD2</title>
  </head>
  <body>
    <h1>Bienvenue sur la page web du TD de BDD2</h1>

    <p>Bonjour la promo</p>
  </body>
</html>
```

Remarque : par défaut, le fichier index.html (ou index.php) est lu par le navigateur.

2. Balises indispensables

- h1, h2, ..., h6 : titres
- p : paragraphe
- br : aller à la ligne (auto-fermante)
- ul : liste à puces
 - li : élément de liste à puces
- a : lien
- img : image
- table : tableau
 - tr : ligne de tableau
 - td : case de tableau

3. Formulaire HTML

```
<form>
  Nom : <input type="text" name="nom" /><br/>
  Mot de passe : <input type="password" name="mdp" />
</form>
```

La balise input peut avoir comme type : button, checkbox, file, hidden, image, password, radio, reset, submit ou text.

4. Page HTML complète

- Déclaration de version de html

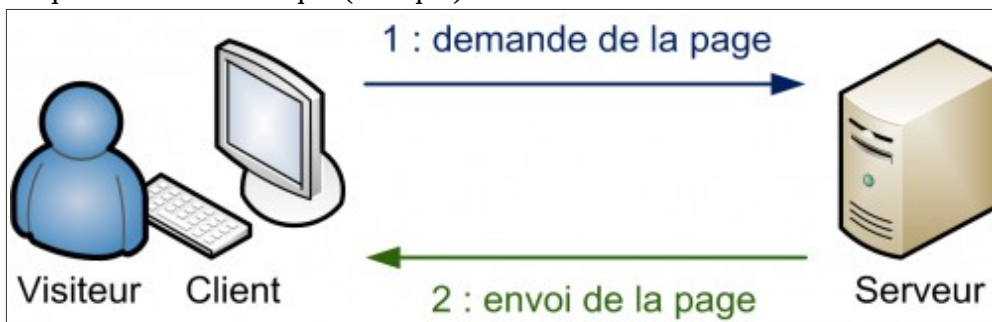
- Titre (et favicon)
- Méta-tags
- etc.

II. PHP génère du HTML

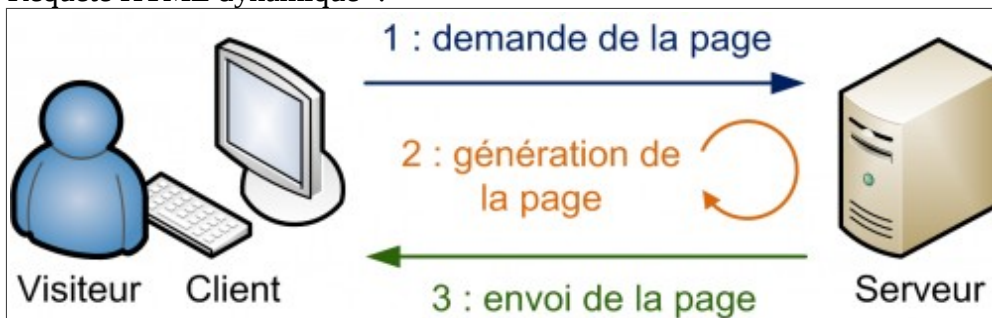
Version actuelle : PHP 5.
Documentation : <http://www.php.net/manual/fr/> ,
http://www.w3schools.com/php/php_intro.asp

1. Fonctionnement de PHP

- Requête HTML classique (statique)* :



- Requête HTML dynamique* :



1 : requête HTTP 2 : génération PHP 3 : envoi du code HTML/CSS

2. Implémentation

- balise PHP :

```
<?php /* Le code PHP se met ici */ ?>  
<? /* ou le code PHP se met ici */ ?>  
<? /* ou  
    sur plusieurs lignes  
    */  
?>
```

- se met n'importe où dans le code HTML :

```
<?php /* fichier index.php */ ?>  
<html>  
  <head>  
    <title>TD de BDD2</title>  
  </head>
```

* Schémas extraits de <http://www.siteduzero.com>

```
<body>
  <h1>Bienvenue sur la page web du TD de BDD2</h1>

  <?php
    echo '<p>Bonjour la promo</p>' ;
  ?>
</body>
</html>
```

- Type dynamique, structures de contrôle proche du C :

```
<? /* fichier index.php */ ?>
<?
  $mvariable_int = 1 ;
  $mvariable_bool = true ;
  $mvariable_texte = "Bonjour la promo" ;
?>
<html>
  <head>
    <title>TD de BDD2</title>
  </head>
  <body>
    <h1>Bienvenue sur la page web du TD de BDD2</h1>

    <?php
      if($mvariable_bool) {
        echo '<p>'.$mvariable_texte.'<br/>' ;
        echo 'Ce cours va durer ' . ($mvariable_int
          *2).'heures.</p>' ;
      }
    ?>
  </body>
</html>
```

Types

- Booléen (true/false)
- Entier
- Flottant
- Chaîne ('chaîne non interprétée', "\$variable")
- Tableau, indicé par des entiers ou par des chaînes : (tab[0], tab['SERVER'])

Structures de contrôle

- if (<cond>) <instr> ; else <instr.> ;
- while (<cond>) <instr> ;
- for (<expr1> ; <expr2> ; <expr3>) <instr> ;
- foreach (<tableau> as <variable>) ;
- switch (<var>) {
 case(<val1>) : <instr> ;
 case(<val2>) : <instr> ;
 default : <instr> ;
}

3. Passage d' « arguments » à une page :

- passage par URL (méthode GET) :

```
http://<monserveur.com>/index.php?text=Bonjour la promo
```

puis dans le code :

```
$mavARIABLE_texte = $_GET['text'];
```

- passage par formulaire (méthode POST) :

```
<form method="post" action="index.php">
  Nom : <input type="text" name="nom" /><br/>
  <input type="submit" value="Valider" />
</form>
```

puis dans le code :

```
$mavARIABLE_texte = $_POST['nom'];
```

4. Notions utiles

- `$_SERVER["REMOTE_ADDR"]` contient l'adresse IP du client (demandeur de la page).
- La fonction PHP `isset(<variable>)` permet de voir si une variable est initialisée.
- La concaténation de chaînes de caractères se fait par l'opérateur « . » (point).

5. Exercice

Écrire le code PHP/HTML qui demande à l'utilisateur d'entrer son nom et qui affiche ensuite un message d'accueil personnalisé ainsi que son adresse IP.

III. PHP comme interface entre SGBD et HTML

Documentation PHP/PostgreSQL :

<http://www.php.net/manual/fr/book.pgsql.php>

Documentation PHP/Oracle : <http://php.net/manual/fr/book.oci8.php> *

PHP permet de se connecter à une base de données et d'interagir avec elle. L'exemple suivant (fonctions spécifiques à PostgreSQL) en illustre l'utilisation :

```
<?php
// creation de la connection
$connection =
pg_connect('host=<host> dbname=<dbname> user=<login>
password=<pwd>') ;

// test de la connection
if (!$connection) {
  echo 'Echec de l'ouverture de la connection' ;
  exit ;
}

// definition de la requete
$query = 'SELECT nom, prenom, adresse
FROM personne
ORDER BY nom DESC' ;

// envoie de la requete a la base et execution
$sql_result = pg_query($connection,$query) ;

// nombre de colonne du resultat
$num_columns = pg_num_fields($sql_result) ;
```

```
// formatage de l'affichage du resultat
echo '<table>' ;
echo '<tr><th>Nom</th> <th>Prenom</th> <th>Adresse</th>' ;
// affichage ligne par ligne
while ($row = pg_fetch_array($sql_result)) {
    echo '<tr>' ;
    for ($i=0; $i < $num_columns; $i++) {
        echo '<td>'.$row[$i].'</td>' ;
    }
    echo '</tr>' ;
}
echo '</table>' ;

// liberation des ressources et fermeture de la connection
pg_free_result($sql_result) ;
pg_close($connection) ;

?>
```

1. Exercice : information de connection

La table suivante contient pour un utilisateur défini par son adresse IP, le nombre de fois qu'il a consulté le site ainsi que la date de sa dernière connexion :

```
info_connect(
    ip VARCHAR2(16),
    last_connect DATE,
    nb_connect INTEGER
)
```

Écrire un script PHP permettant d'afficher le contenu de la table `info_connect`.

2. Exercice : statistiques

Écrire un script PHP `compteur.php` permettant d'afficher le nombre de personnes différentes ayant consultées une page HTML (le nombre de visiteurs est déduit du contenu de la table `info_connect`). Une visite d'un utilisateur déjà enregistré entraînera la mise à jour des informations qui lui sont associés (nombre de visites et date de dernière connexion).

3. Exercice : base de données de projets

On décide d'enregistrer les projets à destination des étudiants dans une base de données. Une interface HTML permettra d'ajouter, d'afficher ou de supprimer les projets.

La table `projet` est définie de la façon suivante :

```
projet(
    idProjet INTEGER,
    auteur VARCHAR2(64),
    texte VARCHAR2(512),
    url VARCHAR2(256)
)
```

Écrire le formulaire HTML et le script PHP associé. Astuce : la balise html `<textarea>` permet de définir une zone de texte à plusieurs lignes.

IV. CSS pour mettre en page le HTML

Version actuelle : CSS 3
Documentation : <http://www.w3schools.com/css>

Le CSS (Cascading Style Sheet) permet de mettre en page le contenu (HTML). Pour associer une page CSS à une page HTML, il suffit d'ajouter la ligne suivante dans les metadata HTML :

```
<link href="<monfichier.css>" rel="stylesheet" type="text/css"
media="screen" />
```

Le fichier <monfichier.css> peut désormais contenir le code CSS associé à la page HTML.

Voici un exemple de CSS :

```
p { /* Tout texte dans un paragraphe sera rouge sur fond noir */
  color : red ;
  background : black ;
}

h2 { /* Le titre de niveau 2 aura une hauteur de 36 pixels. */
  font-size : 36px ;
}

/* Un lien inclus dans un menu deviendra rouge lorsque l'on passera
dessus avec la souris */
ul a:hover {
  background : yellow ;
}
```

Les sélecteurs (p ; h2 ; ul a:hover) définissent des cibles auxquelles on souhaite appliquer un certain style. Si plusieurs sélecteurs s'appliquent à une même balise html et définissent la même propriété (la couleur par exemple), alors le dernier lu (le plus bas dans le fichier) est appliqué.

De nombreuses possibilités sont offertes par le CSS, en particulier depuis sa version 3. Attention : tous les navigateurs n'implémentent pas de façon identique certains sélecteurs CSS.

V. Structuration du projet et du code

1. Une page web est souvent structurée en 4 parties : en-tête, menu, corps et pied de page. Sachant qu'en règle générale, il n'y a que le corps (ou presque) qui change lorsque le visiteur navigue sur le site, comment structurerez-vous votre code afin de le rendre lisible et d'éviter les copiers-collers dans différents fichiers ?
Astuce : la fonction PHP `include('<filename>')` permet de charger le contenu d'un fichier dans un autre).
2. Un certain site web est défini par son code source qui est réparti dans les fichiers suivants : fichiers de code html/php (un fichier pour chaque corps + un fichier pour chaque autre élément), librairies php, feuilles de style CSS. Ce site dispose également d'un grand nombre d'images. Comment structurerez-vous vos répertoires ?