# Cours 2 : Introduction au langage PHP

Module I-C3 : Applications Web
IUT R&T 2e année

David Mercier



## PHP : Hypertext Preprocessor
### Introduction

**Motivation**

- Langage de scripts généraliste et Open Source, spécialement conçu pour le développement d'applications web.
- Il peut être intégré facilement au (X)HTML.
  - <?php ... ?>
- Grande force de PHP : nombreuses fonctions natives
  - Support de nombreuses bases de données (PostgreSQL, MySQL, Oracle, ...)
  - Support de nombreux protocoles (LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM,...)
- Syntaxe voisine du langage C et typage voisin du Perl

**Autres remarques**

- Choix d'utiliser la programmation procédurale ou objet (PHP 5 introduit un modèle objet complet)
- Histoire : créé par Rasmus Lerdorf (1995, Personal Home Page), très populaire.

**Utilisation traditionnelle : langage de script exécuté côté serveur.**



## PHP : Hypertext Preprocessor
### Architecture 3 tiers

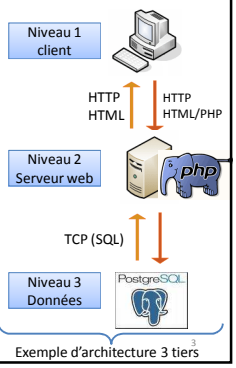
L'architecture 3 tiers est composée de trois niveaux

- Couche présentation
- Couche métier
- Couche accès aux données

- !Attention!: « tier » en anglais = niveau/étage

Extension du modèle client/serveur = architecture 2 tiers

Cas particulier de l'architecture n tiers




Exemple d'architecture 3 tiers



## PHP : Hypertext Preprocessor
### Exemple 1 : une première page PHP (1/2)

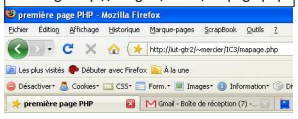
mapage.php — **En pratique ici : à mettre dans votre /home/~user/public_html/**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xml?ns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-15"/>
    <title> première page PHP </title>
</head>
<body>
    <?php echo "<p> Bonjour tout le monde ! Ceci est déjà un script PHP </p>"; ?>
</body>
</html>
```



## PHP : Hypertext Preprocessor
### Exemple 1 : une première page PHP (2/2)

Affichage http://iut-gtr2/~user/mapage.php




Code source généré (plus de PHP)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
        "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-15"/>
        <title> première page PHP </title>
</head>
<body>
        <p> Bonjour tout le monde ! Ceci est déjà un script PHP </p>
</body>
</html>
```

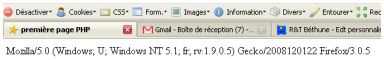


## PHP : Hypertext Preprocessor
### Exemple 2 : détection du navigateur utilisé par un visiteur

- Information stockée dans une variable réservée (superglobale) de PHP : $_SERVER['HTTP_USER_AGENT'].
- $_SERVER contient toutes les informations relatives au serveur web
- **Variables PHP commencent toutes par un signe dollar.**

```
<body>
    <?php echo $_SERVER['HTTP_USER_AGENT']; ?>
</body>
```

Affichage possible




Mozilla/5.0 (Windows; U; Windows NT 5.1; fr; rv:1.9.0.5) Gecko/2008120122 Firefox/3.0.5

## PHP : Hypertext Preprocessor

### Exemples de variables prédéfinies

| Variable | Description |
|---|---|
| $_SERVER['DOCUMENT_ROOT'] | Racine du serveur |
| $_SERVER['HTTP_ACCEPT_LANGUAGE'] | Langage accepté par le navigateur |
| $_SERVER['HTTP_HOST'] | Nom de domaine du serveur |
| $_SERVER['HTTP_USER_AGENT'] | Type de navigateur |
| $_SERVER['PATH_INFO'] | Chemin WEB du script |
| $_SERVER['PATH_TRANSLATED'] | Chemin complet du script |
| $_SERVER['REQUEST_URI'] | Chemin du script |
| $_SERVER['REMOTE_ADDR'] | Adresse IP du client |
| $_SERVER['REMOTE_PORT'] | Port de la requête HTTP |
| $_SERVER['QUERY_STRING'] | Liste des paramètres passés au script |
| $_SERVER['SERVER_ADDR'] | Adresse IP du serveur |
| $_SERVER['SERVER_ADMIN'] | Adresse de l'administrateur du serveur |
| $_SERVER['SERVER_NAME'] | Nom local du serveur |
| $_SERVER['SERVER_SIGNATURE'] | Type de serveur |
| $_SERVER['REQUEST_METHOD'] | Méthode d'appel du script |

## PHP : Hypertext Preprocessor

### Les types

**Types de données**

- Non nécessaire de déclarer, mais il faut les connaître
- Booléens ('boolean'), entiers ('integer'), flottants ('float', 'double'), chaînes ('string')

```
<body>
    <?php
    $bool = TRUE;           #un booléen
    $bool = FALSE;          //un autre booléen
    $a = 12;                /*un entier*/
    $b = 0x1A;              //nb hexadécimal
    $c = 012;               // nb octal
    $d = 1.2;               // flottant
    $e = 5e7;               // flottant
    $s1 = "livre";          //une chaine
    $s2 = "son $s1 est bon";    //une chaine
    $s3 = 'son $s1 est bon';    //une chaine
    $s4 = "\$s1 contient \" $s1 \".";
    echo $s1."<br />";
    echo $s2."<br />";
    echo $s3."<br />";
    echo $s4."<br />";
    ?>
</body>
```

Affichage :

## PHP : Hypertext Preprocessor

### Concaténer deux chaînes

```
</head>
<body>
    <?php
        $nom = "Toto le champion";
        echo "<p>Bonjour ".$nom."!</p>";
    ?>
</body>
</html>
```

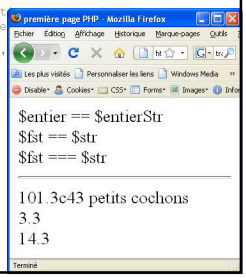
```
</head>
<body>
        <p>Bonjour Toto le champion!</p>
</body>
</html>
```



## PHP : Hypertext Preprocessor

### Transtypage (cast)

```
<?php
    $entier = 10;
    $entierStr = "10";
    $str = "$entier";
    $fst = (string) $entier; // CAST

    if ($entier==$entierStr) echo '$entier == $entierStr<br />';
    if ($entier===$entierStr) echo '$entier === $entierStr<br />';
    if ($fst == $str) echo '$fst == $str <br />';
    if ($fst === $str) echo '$fst === $str <br />';
?>
    <hr />
<?php
    $a = 10;
    $b = TRUE;
    $c = FALSE;
    $d = ".3c4";
    $e = "3 petits cochons";

    echo $a.$b.$c.$d.$e.'<br />';
    echo $d+$e.'<br />';
    echo $a+$b+$c+$d+$e.'<br />';
?>
```




## PHP : Hypertext Preprocessor

### Fonctions utiles liées aux types

Certaines fonctions sont nécessaires pour pouvoir connaître/modifier le type d'une variable :

- isset($v) : renvoie true si la variable existe
- is_int($v), is_array($v) : connaître si une variable est de type entier, tableau…
- gettype($v) : renvoie le type de la variable sous forme de chaîne de caractères.
- unset($v) : détruit la variable.
- Il y en a plein d'autres…

**Excellente documentation !** http://www.php.net/manual/fr/



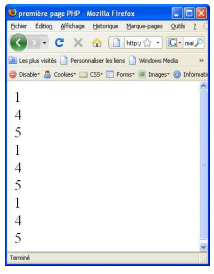
## PHP : Hypertext Preprocessor

### Tableaux classiques

```
<?php
    $a[0] = 1;
    $a[1] = 4;
    $a[2] = 5;

    $b[] = 1;
    $b[] = 4;
    $b[] = 5;

    $c = array(1,4,5);

    for($i=0;$i<count($a);$i++) {
        echo $a[$i]."<br />";
    }
    foreach($b as $val) {
        echo $val."<br />";
    }
    for($i=0;$i<count($c);$i++) {
        echo $c[$i]."<br />";
    }
?>
```

HTML :

```
1<br />4<br />5<br />1<br />4<br />5<br />1<br />4<br />5<br />
```



PHP : Hypertext Preprocessor

Tableaux associatifs

```php
<?php
    $t = array("ville"=>"Bordeaux",
               "cp"=>33000,
               "adresse"=>"1 Place Pey-Berland",
               "tel"=>"05.00.00.00.00");

    $t[] = "Un nouveau élément";

    foreach($t as $key=>$val) {
        echo "$key : $val"."<br />";
    }
?>
```

PHP : Hypertext Preprocessor

Tableaux : fonctions de tri

**Tableaux**

- sort(), rsort() *décroissant*

**Tableaux associatifs**

- asort(), arsort() sur la valeur
- ksort() sur la clé



PHP : Hypertext Preprocessor

Structure de contrôle : if

```
</head>
<body>
    <?php
    $nombre = 11;
    if ($nombre >= 0 && $nombre < 10) {
        echo $nombre.' est compris entre 0 et 9';
    }
    elseif ($nombre >= 10 && $nombre < 20) {
        echo $nombre.' est compris entre 10 et 19';
    }
    else {
        echo $nombre.' est plus grand que 19';
    }
    ?>
</body>
</html>
```

```
</head>
<body>
11 est compris entre 10 et 19
</body>
</html>
```



PHP : Hypertext Preprocessor

Structure de contrôle : switch

```php
<?php
$nom = "Toto";
switch ($nom) {
    case "Toto" :
        echo 'Votre nom est Toto.';
        break;
    case "David" :
        echo 'Votre nom est David.';
        break;
    default :
        echo 'Je ne sais pas qui vous êtes !!!';
}

if ($nom == "Toto") {
    echo 'Votre nom est Toto.';
}
elseif ($nom == "David") {
    echo 'Votre nom est David.';
}
else {
    echo 'Je ne sais pas qui vous êtes !!!';
}
?>
```

Remarque :
Test d'égalité sur chaîne de caractères…

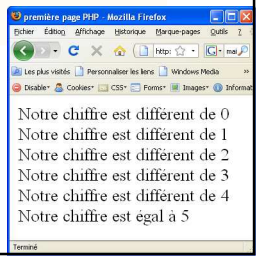
```
<body>
Votre nom est Toto.Votre nom est Toto.
</body>
```



PHP : Hypertext Preprocessor

Boucle for

```php
<?php
    $chiffre = 5;

    for ($i=0; $i < $chiffre; $i++) {
        echo 'Notre chiffre est différent de '.$i.'<br />';
    }

    echo 'Notre chiffre est égal à '.$i;
?>
```

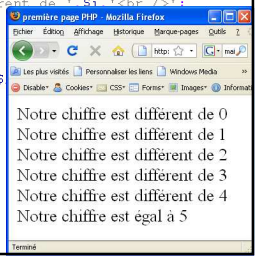





PHP : Hypertext Preprocessor

Boucle while

```php
<?php
    $chiffre = 5;

    $i = 0;
    while ($i < $chiffre) {
        echo 'Notre chiffre est différent de '.$i.'<br />';
        $i = $i + 1;
    }

    echo 'Notre chiffre est égal à '.$i;
?>
```



PHP : Hypertext Preprocessor

fonctions

```php
<?php
    function compteur1($i) {
        /* PASSAGE PAR VALEURS */
        $i++;
    }
    function compteur2(&$i) {
        /* PASSAGE PAR ADRESSES */
        $i++;
    }
    $a=5;
    echo '1. $a='.$a."<br />";
    compteur1($a);
    echo '2. $a='.$a."<br />";
    compteur2($a);
    echo '3. $a='.$a."<br />";
?>
```

PHP : Hypertext Preprocessor

La fonction date()

**string date ( string $format [, int $timestamp ] )**

Retourne une date sous forme d'une chaîne, au format donné par le paramètre ***format*** , fournie par le paramètre ***timestamp*** ou la date et l'heure courantes si aucun timestamp n'est fourni.

```php
echo date("d")."<br />"; // 01-31
echo date("D")."<br />"; // Mon à sun
echo date("l")."<br />"; // Monday...
echo date("m")."<br />"; // 01-12
echo date("M")."<br />"; // Jan à Dec
echo date("F")."<br />"; // January ...
echo date("Y")."<br />"; // 2009 ...
etc ... pour heures minutes ...
```

Voir : http://fr3.php.net/manual/fr/function.date.php

**Pour avoir les dates en français utiliser strftime() et setlocale() :**

```
$error = @setlocale('LC_TIME', 'fr_FR.utf8','fra');
setlocale('french');
```

PHP : Hypertext Preprocessor

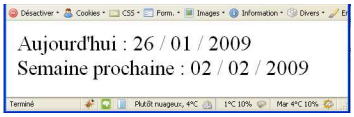
La fonction time()

**int time (void)**

Retourne l'heure courante, mesurée en secondes depuis le début de l'époque UNIX, (1er janvier 1970 00:00:00 GMT).

```php
<?php
$nextWeek = time() + (7 * 24 * 60 * 60);
// 7 jours; 24 heures; 60 minutes; 60secondes
echo 'Aujourd\'hui :       '. date('d / m / Y') ."<br/>";
echo 'Semaine prochaine : '. date('d / m / Y', $nextWeek) ."<br/>";
?>
```

Au moment de la création de ce transparent →



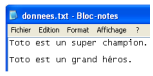



PHP : Hypertext Preprocessor

Lecture d'un fichier ligne par ligne

```php
<?php
$ligne = @fopen("donnees.txt", "r")
if ($ligne) {
    while (!feof($ligne)) {
        $buffer = fgets($ligne, 4096);
        echo $buffer."<br />";
    }
    fclose($ligne);
}
?>
```

Documentation : http://www.php.net/manual/fr/




bool **feof** ( resource *$handle* )

Teste la fin du fichier.

string **fgets** ( resource *$handle* [, int *$length* ] )

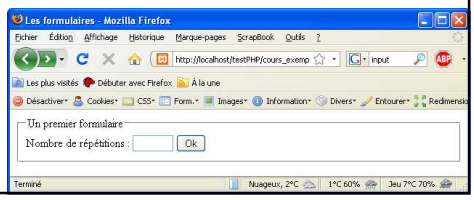
Récupère la ligne courante sur laquelle se trouve le pointeur du fichier.

PHP : Hypertext Preprocessor

Formulaire 1

mapage8_form1.php

```php
<form method="post" action="mapage8_form1_repBonjour.php" />
    <fieldset>
        <legend> Un premier formulaire </legend>
        <label for="repet">Nombre de répétitions</label> :
        <input type="text" size="5" name="nbre_nom" id="nbre_id" />
        <input type="submit" value=" Ok " />
    </fieldset>
</form>
```






PHP : Hypertext Preprocessor

Formulaire 1

mapage8_form1.php

```php
<form method="post" action="mapage8_form1_repBonjour.php" />
    <fieldset>
        <legend> Un premier formulaire </legend>
        <label for="repet">Nombre de répétitions</label> :
        <input type="text" size="5" name="nbre_nom" id="nbre_id" />
        <input type="submit" value=" Ok " />
    </fieldset>
</form>
```

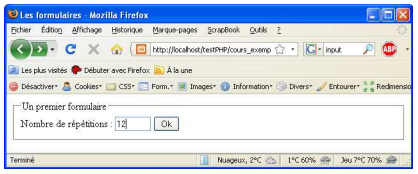
mapage8_form1_repetBonjour.php

```php
<?php
    for($i=1;$i<=$_POST['nbre_nom'];$i++){
        echo "$i : Bonjour ! <br />";
    }
?>
```

PHP : Hypertext Preprocessor

Formulaire 1

```
<form method="post" action="mapage8_form1_repBonjour.php" />
    <fieldset>
        <legend> Un premier formulaire </legend>
        <label for="repet">Nombre de repetitions</label> :
        <input type="text" size="5" name="nbre_nom" id="nbre_id" />
        <input type="submit" value=" Ok " />
    </fieldset>
</form>
```




```
<?php
    for($i=1;$i<=$_POST['nbre_nom'];$i++){
        echo "$i : Bonjour ! <br />";
    }
?>
```

PHP : Hypertext Preprocessor

Formulaire 2

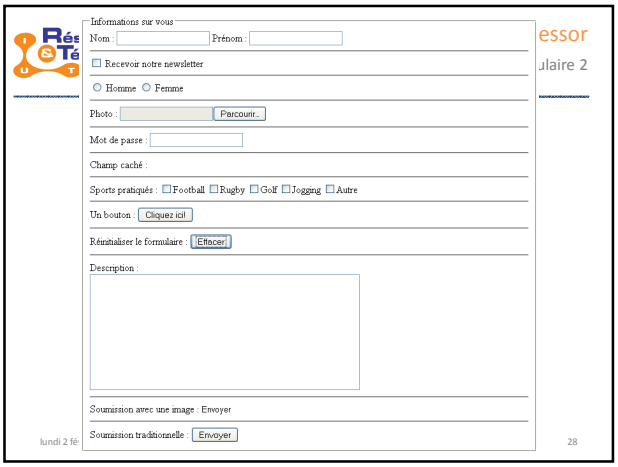
```
<form method="post" action="mapage8_form2_action.php">
    <fieldset>
        <legend>Informations sur vous</legend>
        <label for="nom">Nom</label> :
        <input type="text" name="nom" id="nom" />
        <label for="prenom">Prénom</label> :
        <input type="text" name="prenom" id="prenom" />
        <hr />
        <input type="checkbox" name="newsletter" id="newsletter" />
        <label for="newsletter">Recevoir notre newsletter</label>
        <hr />
        <input type="radio" name="sexe" id="homme" />
        <label for="homme">Homme</label>
        <input type="radio" name="sexe" id="femme" />
        <label for="femme">Femme</label>
        <hr />
        <label for="photo">Photo</label> :
        <input type="file" name="photo" id="photo" />
        <hr />
        <label for="pass">Mot de passe</label> :
        <input type="password" name="pass" id="pass" />
        <hr />
        <label for="champCacher">Champ caché</label> :
        <input type="hidden" name="champCacher" id="champCacher" />
        <hr />
```



PHP : Hypertext Preprocessor

Formulaire 2

```
        <label for="sport">Sports pratiqués</label> :
        <input type="checkbox" name="sports[]" value="football" />Football
        <input type="checkbox" name="sports[]" value="rugby" />Rugby
        <input type="checkbox" name="sports[]" value="golf" />Golf
        <input type="checkbox" name="sports[]" value="jogging" />Jogging
        <input type="checkbox" name="sports[]" value="autre" />Autre
        <hr />
        <label for="bouton">Un bouton</label> :
        <input type="button" name="bouton" id="bouton" value="Cliquez ici!" />
        <hr />
        <label for="initialize">Réinitialiser le formulaire</label> :
        <input type="reset" name="init" id="init" />
        <hr />
        <label for="description">Description</label> : <br />
        <textarea rows="10" cols="50" name="description" id="description"></textarea>
        <hr />
        <label for="image">Soumission avec une image</label> :
        <input type="image" name="image" id="image" src="/image/soumission_image.jpg" />
        <hr />
        <label for="soumission">Soumission traditionnelle</label> :
        <input type="submit" name="soumission" id="soumission" />
    </fieldset>
</form>
```






PHP : Hypertext Preprocessor

Formulaire 2

action

```
<?php
    foreach($_POST as $k => $val){
        echo "$k => $val <br />";
        if(is_array($val)){
            echo "Le tableau ".$val." compte ".count($val)." élément(s) qui sont : ";
            for($i=0;$i<count($val);$i++){
                echo "$val[$i] ";
            }
            echo "<br />";
        }
    }
?>
```

À essayer !

Voir aussi la fonction **print_r**



PHP : Hypertext Preprocessor

Formulaire 2

action

```
<?php
    foreach($_POST as $k => $val){
        echo "$k => $val <br />";
        if(is_array($val)){
            echo "Le tableau ".$val." compte ".count($val)." élément(s) qui sont : ";
            for($i=0;$i<count($val);$i++){
                echo "$val[$i] ";
            }
            echo "<br />";
        }
    }
?>
```

À essayer !

Voir aussi la fonction **print_r**

## PHP : Hypertext Preprocessor

### Les Sessions : motivations

Permettent la **sauvegarde de données confidentielles ou importantes sur le serveur** et non sur le client comme pour les cookies (fichiers texte courts voir $_COOKIE et setcookie())

Exemples d'emploi :

- Authentifier un visiteur;
- Garder des informations d'un visiteur tout au long de sa présence dans votre application;
- Gérer le panier d'achat d'un internaute sur un site marchand;
- Mettre en place des formulaires en plusieurs parties
- …



## PHP : Hypertext Preprocessor

### Les Sessions : principe

**session_start()** appelée **au début (avant le moindre code (X)HTML)** de chaque page utilisant les variables de sessions.

Une fois la session initialisée, on peut accéder aux variables de sessions grâce au tableau **$_SESSION[]**. C'est dans ce tableau que vous mettrez les variables auxquelles vous voulez accéder de page en page.

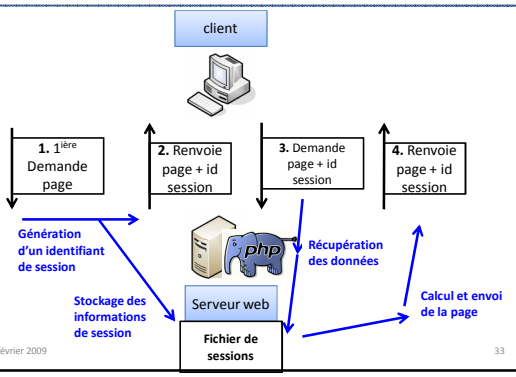
Les informations de sessions sont stockées dans des fichiers sur le serveur, à chaque session correspond un fichier. Chaque session est désignée par un nom et un identifiant.

- Par défaut le nom est généralement **PHPSESSID**, l'identifiant est lui unique, différent pour chaque session par exemple : 84af0e0b337f35960b36cc3b37ef2ddc.

Les informations de sessions sont donc stockées sur le serveur, mais celui-ci a besoin de connaître l'identifiant de la session en cours pour pouvoir récupérer ces informations.

## PHP : Hypertext Preprocessor

### Les Sessions : Grandes lignes






## PHP : Hypertext Preprocessor

### Les Sessions : cas des cookies acceptés

Identifiant de la session stocké dans un cookie dont le nom par défaut est PHPSESSID

Lorsque la fonction session_start() est appelée :

- si aucun cookie correspondant à une session n'existe, alors le serveur génère un identifiant de session et un cookie est créé.
- si un cookie existe déjà (c'est-à-dire si une session est déjà ouverte), alors le serveur récupère les variables de sessions correspondantes à l'identifiant qui se trouve dans le cookie déjà présent.



## PHP : Hypertext Preprocessor

### Les Sessions : cas des cookies **non** acceptés

Pour que le serveur puisse récupérer l'identifiant de session malgré l'absence du cookie, il faut lui passer cet identifiant dans nos url.

La constante **SID**, qui est une chaîne de caractères et qui est définie si la session a commencée, est égale à une chaîne vide si un cookie de session approprié a été envoyé et à "<nom_de_la_session>=<identifiant_de_la_session>" sinon.

- Si on refuse les cookies, la constante SID vaut donc par exemple quelque chose comme : "PHPSESSID=84af0e0b337f35960b36cc3b37ef2ddc".

session_name() session_id()



## PHP : Hypertext Preprocessor

### Exemple de début de page comportant une session

```
<? php
   /* initialisation de la session : ce code est avant tout code HTML, donc
   avant le DTD en particulier*/
   session_start();

   // écrire 'TOTO' dans la variable 'mavariable'
   $_SESSION['mavariable'] = 'TOTO';

   $tab = array('un', 'dos', tres');
   $_SESSION['tableau'] = $tab;
?>
```

PHP : Hypertext Preprocessor

Fermer une session

PHP efface la session de lui-même au bout d'un certain temps, néanmoins vous pouvez forcer la destruction de la session avec la fonction **session_destroy()**.

Remarque : la fonction session_destroy() ne fait qu'effacer le fichier de données, elle n'efface pas les variables présentes dans $_SESSION[]. Pour éviter tout risque de confusion, effacer $_SESSION[] aussi : unset($_SESSION);



PHP : Hypertext Preprocessor

Utilisation du SGDB PostgreSQL

**PostgreSQL** est un système gratuit de gestion de bases de données relationnel objet (**SGBDRO**). Fonctionnalités comparables à celles des ses concurrents propriétaires comme Oracle.

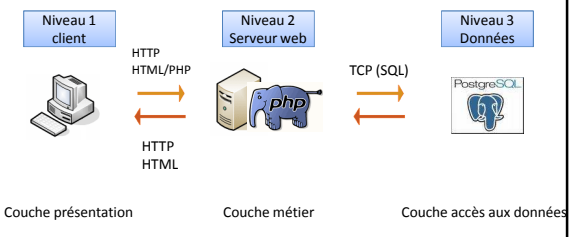
PostgreSQL offre notamment :

- La possibilité d'imposer des **contraintes à l'insertion des données**
- Les **contraintes d'intégrités** (garantie d'intégrité des données)
- La possibilité de créer des **vues**
- La possibilité d'**hériter** des tables et des types de données
- Les **procédures stockées** (programmation coté serveur)
- Les **transactions** (plusieurs étapes en une seule opération)
- Les **triggers** (gestion des événements)

PostgreSQL de plus en plus conforme aux normes ANSI SQL 92 (SQL 2), SQL 99 (SQL 3) et SQL:2003.

PHP : Hypertext Preprocessor

Rappel : architecture 3 tiers






PHP : Hypertext Preprocessor

Connexion au serveur

Connexion à un serveur PostgreSQL :

resource_c pg_connect(string )

http://www.php.net/manual/fr/

**Ressource de connexion** nécessaire aux autres fonctions postgreSQL

```
<?php
// connexion à une base de données nommée "marie"
$conn = pg_connect("dbname=marie");

// connexion à une base de données nommée "marie" sur l'hôte "localhost" sur le port "5432"
$conn2 = pg_connect("host=localhost port=5432 dbname=marie");

// connexion à une base de données nommée "marie" sur l'hôte "mouton" avec un
// nom d'utilisateur et un mot de passe
$conn3 = pg_connect("host=mouton port=5432 dbname=marie user=agneau password=foo");

// connexion à une base de données nommée "test" sur l'hôte "mouton" avec un
// nom d'utilisateur et un mot de passe
$conn_string = "host=mouton port=5432 dbname=test user=agneau password=bar";
$conn4 = pg_connect($conn_string);
?>
```

PHP : Hypertext Preprocessor

Exécuter une requête SQL

Pour exécuter une requête SQL sur la base de données :

resource_r pg_query(resource_c, string)

```
<?php
$conn = pg_connect("...");

$result = pg_query ($conn, "SELECT nom FROM joueur");

/* renvoie
 une ressource de résultat si la requête s'est exécutée correctement,
 FALSE sinon
*/
?>
```



PHP : Hypertext Preprocessor

Lire le résultat d'une requête

Il est pratique d'utiliser :

array pg_fetch_array(resource_r (,...))

```
/* renvoie
 Un tableau contenant la ligne demandée fonction d'une ressource de résultat ,
 FALSE s'il ne reste plus de lignes
*/

Par exemple, affichage du premier élément de chaque ligne :

while ($row = pg_fetch_array($result)) {
        echo "$row[0]  <br />\n";
}
```

Documentation :

http://www.php.net/manual/fr/

## PHP : Hypertext Preprocessor

Un exemple sur notre serveur PostgreSQL iut-gtr2

```
<?php
$conn = pg_connect("host=iut-gtr2 port=5432 dbname=ligue1 user=duchmol password=bdrt00");
if (!$conn) {
        echo "Connexion à la base impossible\n ";
        exit;
}
$result = pg_query($conn, "SELECT nom FROM joueur");
if (!$result) {
        echo "La requête n'a pu être exécutée.\n" ;
        exit;
}
while ($row = pg_fetch_array($result)) {
        echo "Joueur: $row[0] ";
        echo "<br />\n";
}
pg_close ($conn) ;
?>
```

## PHP : Hypertext Preprocessor

Connexion au serveur PostgreSQL (iut-gtr2)

```
<?php
$conn = pg_connect("host=iut-gtr2 port=5432 dbname=ligue1 user=duchmol password=bdrt00");
if (!$conn) {
        echo "Connexion à la base impossible\n";
        exit;
}
$result = pg_query($conn, "SELECT nom FROM joueur");
if (!$result) {
        echo "La requête n'a pu être exécutée.\n" ;
        exit;
}
while ($row = pg_fetch_array($result)) {
        echo "Joueur: $row[0] ";
        echo "<br />\n";
}
pg_close ($conn) ;
?>
```

## PHP : Hypertext Preprocessor

Gestion de l'échec de la connexion

```
<?php
$conn = pg_connect("host=iut-gtr2 port=5432 dbname=ligue1 user=duchmol password=bdrt00");
if (!$conn) {
        echo "Connexion à la base impossible\n ";
        exit;
}
$result = pg_query($conn, "SELECT nom FROM joueur");
if (!$result) {
        echo "La requête n'a pu être exécutée.\n" ;
        exit;
}
while ($row = pg_fetch_array($result)) {
        echo "Joueur: $row[0] ";
        echo "<br />\n";
}
pg_close ($conn) ;
?>
```

## PHP : Hypertext Preprocessor

Exécution d'une requête SQL

```
<?php
$conn = pg_connect("host=iut-gtr2 port=5432 dbname=ligue1 user=duchmol password=bdrt00");
if (!$conn) {
        echo "Connexion à la base impossible.\n";
        exit;
}
$result = pg_query($conn, "SELECT nom FROM joueur");
if (!$result) {
        echo "La requête n'a pu être exécutée.\n" ;
        exit;
}
while ($row = pg_fetch_array($result)) {
        echo "Joueur: $row[0] ";
        echo "<br />\n";
}
pg_close ($conn) ;
?>
```

## PHP : Hypertext Preprocessor

Gestion de l'échec d'exécution de la requête

```
<?php
$conn = pg_connect("host=iut-gtr2 port=5432 dbname=ligue1 user=duchmol password=bdrt00");
if (!$conn) {
        echo "Connexion à la base impossible.\n";
        exit;
}
$result = pg_query($conn, "SELECT nom FROM joueur");
if (!$result) {
        echo "La requête n'a pu être exécutée.\n" ;
        exit;
}
while ($row = pg_fetch_array($result)) {
        echo "Joueur: $row[0] ";
        echo "<br />\n";
}
pg_close ($conn) ;
?>
```

## PHP : Hypertext Preprocessor

Affichage de chaque ligne du résultat de la requête

```
<?php
$conn = pg_connect("host=iut-gtr2 port=5432 dbname=ligue1 user=duchmol password=bdrt00");
if (!$conn) {
        echo "Connexion à la base impossible.\n";
        exit;
}
$result = pg_query($conn, "SELECT nom FROM joueur");
if (!$result) {
        echo "La requête n'a pu être exécutée.\n" ;
        exit;
}
while ($row = pg_fetch_array($result)) {
        echo "Joueur: $row[0] ";
        echo "<br />\n";
}
pg_close ($conn) ;
?>
```

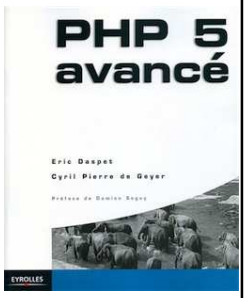
PHP : Hypertext Preprocessor

Fermeture de la base

```php
<?php
$conn = pg_connect("host=iut-gtr2 port=5432 dbname=ligue1 user=duchmol password=bdrt00");
if (!$conn) {
        echo "Connexion à la base impossible.\n";
        exit;
}
$result = pg_query($conn, "SELECT nom FROM joueur");
if (!$result) {
        echo "La requête n'a pu être exécutée.\n" ;
        exit;
}
while ($row = pg_fetch_array($result)) {
        echo "Joueur: $row[0] ";
        echo "<br />\n";
}
pg_close ($conn) ;
?>
```

PHP : Hypertext Preprocessor

Pour aller plus loin : références

Une mine d'informations :

http://fr3.php.net/manual/fr/