

Programmation web

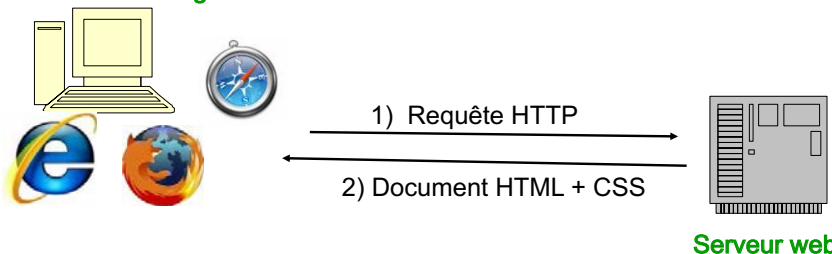
Jean-Marie PECATTE
IUT Paul Sabatier
jean-marie.pecatte@iut-tlse3.fr

Le langage PHP



Schéma de principe client / serveur

Client avec navigateur



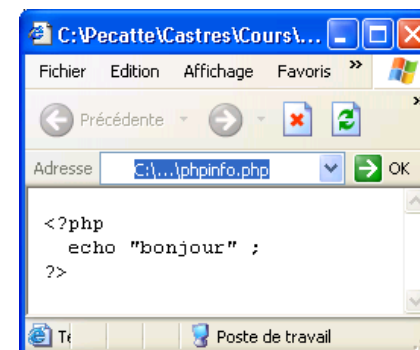
- Le serveur web ne fait rien sauf renvoyer le fichier demandé
- Le navigateur reçoit le fichier HTML (et éventuellement le fichier CSS), l'interprète et l'affiche

Mon premier programme PHP

phpinfo.php

```
<?php
echo "bonjour" ;
?>
```

Ouverture du fichier directement dans un navigateur (sans utiliser de serveur) → pas de http dans l'adresse ?



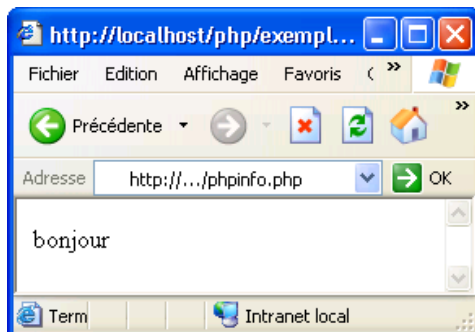
Les navigateurs ne comprennent pas le PHP !!!

Mon premier programme PHP

phpinfo.php

```
<?php
echo "bonjour" ;
?>
```

Ouverture du fichier dans le navigateur mais en utilisant un serveur web (->http dans l'adresse)

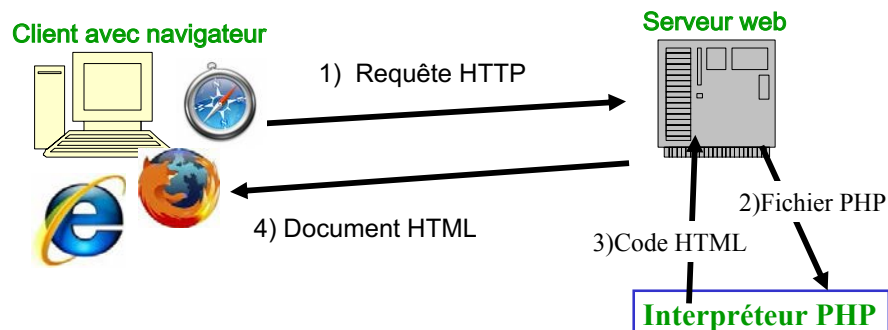


Nécessité d'un serveur web avec un interpréteur PHP
-> adresse : http://...

Mon premier programme PHP

- Extension du fichier ?
PHP (éventuellement .PHP4, .PHP3, ...)
pas HTM (HTML) : le code PHP présent dans ce type de fichier ne sera pas interprété.
- Si l'environnement est bien configuré et les règles précédentes respectées, le code PHP ne doit pas être visible dans le navigateur (ni dans le code source du document)

Schéma de principe client / serveur



- Le serveur web interprète le fichier PHP et renvoie uniquement du code HTML
- Le navigateur reçoit le fichier HTML, l'interprète et l'affiche
- Pas de différence pour le navigateur

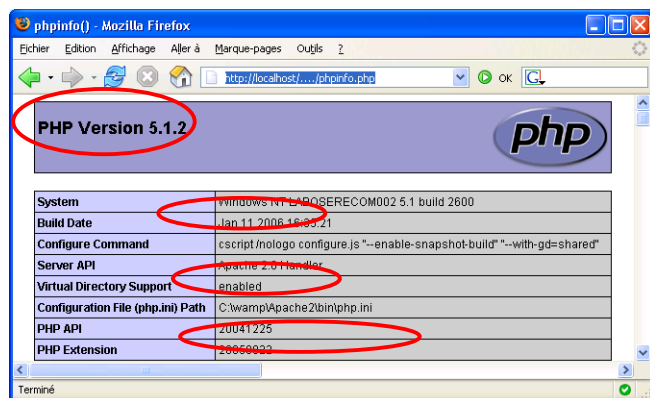
Environnement PHP

Comment savoir si PHP fonctionne, s'il est installé ?
Comment connaître la version de PHP installé ainsi que la liste des modules ?

Exécuter le script suivant :

phpinfo.php

```
<?php
phpinfo();
?>
```



Traitement des formulaires en PHP

Traitement des formulaires en PHP

- Les données saisies dans des formulaires HTML peuvent être traitées par un pgm PHP

Etape 1 : créer le formulaire en HTML

⇒ Donner un nom à chaque contrôle (**attribut name**)

⇒ Indiquer le nom du script PHP à exécuter

(**attribut action de l'élément form**)

⇒ Indiquer le mode de passage des valeurs saisies (**attribut mode de l'élément form**)

Etape 2 : écrire le script PHP

⇒ Utiliser le tableau \$_POST pour récupérer les valeurs saisies

⇒ Le tableau contient une case pour chaque attribut name

Traitement des formulaires en PHP

form.html

```
<form method="post" action="recup.php">
  <fieldset>
    <legend>coordonnes</legend>
    <label>nom </label> <input type="text" name="nom" />
    <input type="submit" value="envoyer" />
  </fieldset>
</form>
```

Ouverture dans un navigateur

coordonnes

nom

Saisie et validation du formulaire (*dans le navigateur*)

coordonnées

nom



Recherche du pgm pour traiter le formulaire (*valeur de l'attribut action de la balise <form>*)

[http:// ... / recup.php](http://.../recup.php)

Création automatique du tableau PHP \$_POST qui contient toutes les valeurs saisies dans le formulaire (*autant de cases que d'attributs name dans le formulaire en html*)

nom
\$_POST

M. Dupont			
-----------	--	--	--

Fichier **recup.php**

```
<?php
```

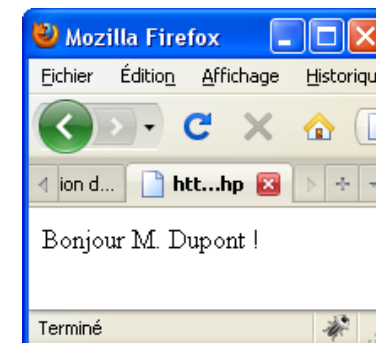
```
// récupérer le nom saisi (zone de texte)
```

```
$name = $_POST["nom"];
```

```
echo "Bonjour ". $name. " !";
```

```
?>
```

Valeur de l'attribut name

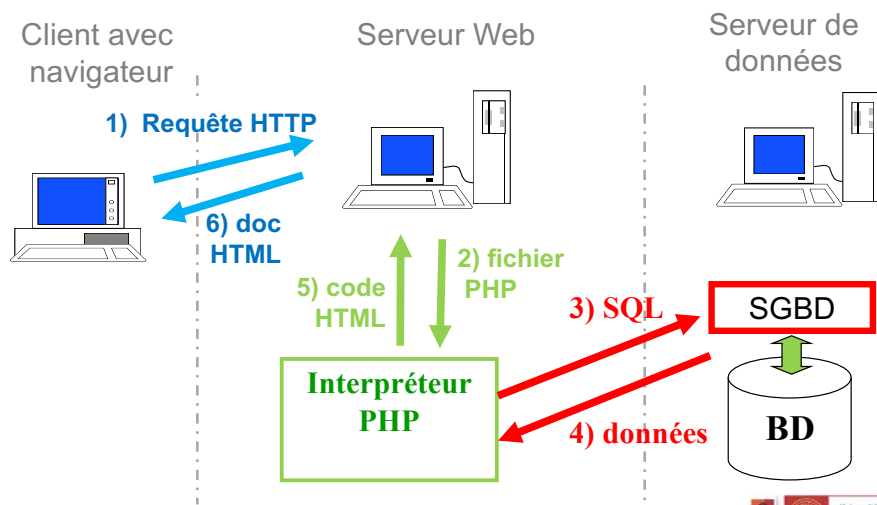


PHP & Bases de données

PHP & BD

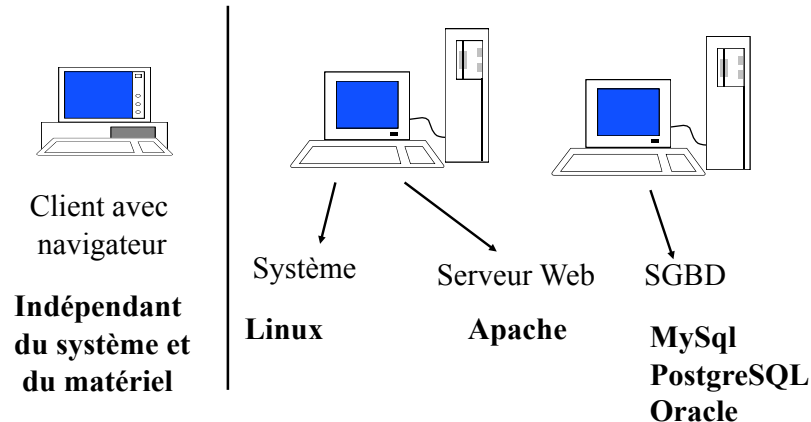
- La quantité de données utilisée par certains sites web nécessite l'utilisation d'une base de données
- Il faut donc disposer d'un SGBD (mysql, postgresql, oracle, ...) installé sur un serveur (pas forcément le même que le serveur web)
- Interactions PHP & BD ?
 - L'interpréteur PHP envoie la requête SQL au SGBD
 - Le SGBD exécute la requête
 - L'interpréteur PHP récupère le résultat
 - L'interpréteur PHP traite le résultat (génère de HTML)

Bases de données et Web



J.-M. Pécatte – IUT Paul Sabatier

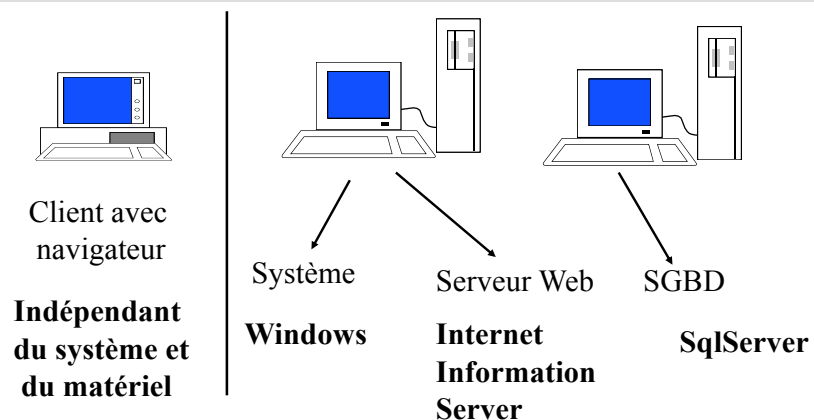
Logiciels & matériels utilisés



Langage de script → PHP

J.-M. Pécatte – IUT Paul Sabatier

Logiciels & matériels utilisés



Langage de script → ASP

J.-M. Pécatte – IUT Paul Sabatier

PHP & BD

- Les 5 étapes pour l'écriture d'un script
 1. Se connecter au SGBD
 2. Sélectionner la BD
 3. Envoie d'une requête
 4. Récupération du résultat
 5. Se déconnecter
- Les exemples présentés se feront en utilisant la bibliothèque PDO (PHP Data Objects) qui permet de rendre l'application indépendante du SGBD.
- PDO est une bibliothèque orientée objet et utilise le mécanisme des exceptions.

J.-M. Pécatte – IUT Paul Sabatier

- Création d'un objet instance de la classe qui tente de réaliser les étapes 1 (connexion) et 2 (choix de la BD)

```
$bdd = new PDO(DSN, USER, PASSWD);
```

Le DSN (Data Source Name) est la chaîne de connexion spécifique au SGBD

```
$bdd = new PDO("mysql:host=sql.free.fr;  
dbname=test", "dupont", "truc" );
```

- Etape 5 : se déconnecter
pas de fonction PDO

déconnexion automatique en fin de script

- Etape 1 & étape 2 : 4 paramètres nécessaires dans toutes les pages et variables suivant les hébergeurs :

→Création d'un fichier de configuration

config.inc.php

```
$server = " sql.free.fr" ;  
$user = "dupont" ;  
$passwd = "truc" ;  
$database = "test" ;
```

→ inclure le fichier de config

```
include "config.inc.php"
```

- Etape 1 & étape 2 :

fichier connect.php

```
include "config.inc.php"  
try { // tentative de connexion avec gestion des exceptions  
    $bdd = new PDO("mysql:host=$server;dbname=$database",  
                  $user, $passwd);  
} catch (Exception $e) {  
    die('Erreur : ' . $e->getMessage()); // die arrête le script  
}
```

→ inclure le fichier de connexion au début de tout fichier PHP qui utiliser une BD :

```
include "config.inc.php"
```

- Etape 3** : Exécution d'une commande SQL

Il faut distinguer 2 cas :

- la requête **ne renvoie pas de données** (toutes les commandes sauf SELECT, comme par exemple INSERT, UPDATE, DELETE, ...)

il suffit d'exécuter la requête

```
$bdd->exec("requête SQL")
```

- la requête **renvoie des données** vers l'application (commande SELECT)

il faut exécuter la requête **\$bdd->query("requête SELECT")**

mais en plus, il faut récupérer le résultat qui est l'ensemble des tuples de la requête

Etape 3 & 4 :

exécution d'une requête SQL autre que SELECT

// suppression de l'étudiant nommé Dupont

```
$nb = $bdd->exec("delete from etudiant  
                where nom = 'Dupont' ");
```

// si la requête est correcte, \$nb contient le nombre de tuples
supprimés, sinon c'est un booléen qui permet de savoir que
la requête n'est pas correcte

```
if ($nb) echo $nb." étudiant(s) supprimé(s)"  
else echo "Pb lors de la suppression";
```

Etape 3 & 4: exécution d'une commande Select

query renvoie un résultat

- Ce résultat contient un ensemble d'enregistrements
- Il n'est possible d'accéder qu'à un seul enregistrement à la fois
- Une fonction **fetch**_... permette de récupérer l'enregistrement courant (au début le premier) et de passer au suivant
- Pour parcourir tous les enregistrements, il faut utiliser une boucle.
- La fonction **fetch**_... renvoie faux si il n'y a plus d'enregistrements

Etape 3 & 4: exécution d'une commande Select

// liste des étudiants dans l'ordre alphabétique

```
$rs = $bdd->query("SELECT nom, prenom  
                FROM etudiant ORDER BY nom, prenom");
```

// boucle pour afficher tous les étudiants

```
while ($etud = $rs->fetch()) {  
    // $etud est un tableau associatif qui contient autant de cases  
    que le select d'attributs  
    echo $etud["nom"] . " " . $etud["prenom"];  
}
```

Quelques exemples

- **Exemple1** : Tableau HTML contenant la liste des étudiants

```
<?php
include "connect.php"
$rs = $bdd->query("SELECT nom, prenom
FROM Etudiant ORDER BY nom, prenom");
// récupération des étudiants 1 par 1
echo '<table>'; // début de tableau HTML
while ($etud = $rs->fetch(PDO::FETCH_ASSOC)) {
    echo '<tr><td>'; echo $etud["nom"]; // 1ère colonne
    echo '</td><td>'; echo $etud["prenom"]; // 2ème colonne
    echo '</td></tr>';
}
echo '</table>'; // fin de tableau HTML
?>
```

J.-M. Pécatte – IUT Paul Sabatier

- **Exemple 2** : zone de liste d'un formulaire contenant la liste des étudiants pour en choisir un

La zone de liste affiche les nom et prénom des étudiants mais la valeur renvoyé est le code étudiant

```
<select name="etudiant">
<option value="13"> jean dupont </option>
<option value="21"> pierre durand </option>
<option value="7"> leo leroy </option>
...
</select>
```

Les valeurs (code et nom de l'étudiant) doivent être récupérées dans la BD

J.-M. Pécatte – IUT Paul Sabatier

Accès à la base pour construire le formulaire

```
<?php
include "connect.php";
$rs = $bdd->query("SELECT code, nom, prenom
FROM Etudiant ORDER BY nom, prenom");
// zone de liste etudiant
echo '<select name="etudiant">';
while ($etud = $rs->fetch(PDO::FETCH_OBJ)) {
    echo '<option value="'. $etud["code"].' ">';
    echo $etud->prenom.' ' . $etud["nom"];
    echo '</option>';
}
echo '</select>';
?>
```

J.-M. Pécatte – IUT Paul Sabatier

Formulaire de gestion de données

- Ex : suppression d'un étudiant

J. Dupont

```
<form action="delete.php" method="post">
```

delete.php

```
// récupère le code de l'étudiant à supprimer
$code_etud = $_POST["etudiant"];
// exécution de la commande DELETE paramétrée par le code etudiant
$ok = $bdd->exec("delete from etudiant where code=" . $code_etud);
```

J.-M. Pécatte – IUT Paul Sabatier