

# Formation à PHP



Juin 2003  
Formation du Grand-Ouest  
Version 2.1

Pascal AUBRY  
IFSIC - Université de Rennes 1  
Pascal.Aubry@univ-rennes1.fr  
<http://perso.ifsic.univ-rennes1.fr/aubry>

L'utilisation de ce document dans un but de formation de groupe, y compris dans le cadre universitaire, est soumise à une autorisation explicite et préalable de son auteur.



# Table des matières

<b>Introduction</b>	<b>1</b>
Le protocole HTTP	
Des pages statiques aux pages dynamiques	
<b>Page dynamiques : les solutions côté serveur</b>	<b>7</b>
Les premiers pas	
Le protocole CGI	
PHP et les autres (FastCGI, Perl, ASP, JSP, servlets, Zope, ...)	
<b>Présentation de PHP</b>	<b>17</b>
PHP et HTML	
Outils/Projets	
Avantages/inconvénients	
Des pages aux composantes	
<b>Le langage PHP</b>	<b>22</b>
Syntaxe de base	
Variables & constantes	
Expressions & opérateurs	
Structures de contrôle	
Fonctions	
Inclusion de fichiers	
<b>Techniques de programmation</b>	<b>41</b>
Bibliothèques	
Formulaires HTML (rappels)	
Paramètres CGI	
Classes/Objets	
<b>Techniques avancées</b>	<b>53</b>
Accès au système de fichiers	
Gestion des erreurs	
Modification de l'entête HTTP	
Sorties non HTML, Redirection, Cookies, Authentification	
Bases de données	
<b>Aller plus loin...</b>	<b>76</b>
Gestion de sessions	
Téléchargement de fichiers	
Gestion des connexions HTTP	
Exécution de programmes externes	
Envoi d'un courrier électronique	
Création d'image	



# Introduction

Le protocole HTTP  
Des pages statiques aux pages dynamiques



Formation à PHP version 2.1 - Copyright© 1999 - 2003 Pascal AUBRY - IFSIC - Université de Rennes 1

## Le World Wide Web

- Système d'information hyper-média réparti
  - texte, images, images animées, son, vidéo ...
- Informations stockées sur les serveurs
  - [www.jres.org](http://www.jres.org), [www.ifsic.univ-rennes1.fr](http://www.ifsic.univ-rennes1.fr), [www.ulaval.ca](http://www.ulaval.ca)
- Interrogation des serveurs
  - tels que Apache, IIS, Netscape, Roxen, AOL, iPlanet, Lotus Domino, NCSA httpd, goAhead
  - par clients tels que Netscape, Internet Explorer, Mosaic, Lynx, Opera, Amaya



Formation à PHP version 2.1 - Copyright© 1999 - 2003 Pascal AUBRY - IFSIC - Université de Rennes 1

# Les principes du Web

- Modèle client-serveur
- Le client envoie des requêtes au serveur
  - transfert de fichiers
  - exécution de programmes sur le serveur
  - mise à jour de fichiers
  - ...
- Objets manipulés repérés par leur URL
- Utilisation du protocole HTTP

Formation PHP version 2.1 - Copyright © 1999 - 2003 Pascal AUBRY - F3SC - Université de Rennes 1



# Le protocole HTTP

- Définit le langage utilisé pour les échanges entre client et serveur Web
  - version 0.9
    - simple protocole de transfert de données (GET et réponse)
  - version 1.0
    - restée un Internet Draft (RFC 1945)
  - actuellement version 1.1
    - RFC 2616 (juin 1999)
- Pas de session permanente entre client/serveur

Formation PHP version 2.1 - Copyright © 1999 - 2003 Pascal AUBRY - F3SC - Université de Rennes 1



# Déroulement d'une requête HTTP



**protocole  
sans état**

- Demande de connexion
- Attente de la réponse du serveur
- Établissement de la connexion
- Envoi d'une requête (URL)
- Réponse du serveur
- Affichage de la réponse
- Fermeture de la connexion



# Exemple de transaction HTTP

```
% telnet www.ifsic.univ-rennes1.fr 80 .....connexion au serveur web
Trying 148.60.4.30...
Connected to apollon.univ-rennes1.fr.
Escape character is '^'.
GET /index.html HTTP/1.1.....demande de transfert
Host: apollon.univ-rennes1.fr .....nom du serveur
From: pa@ifsic.univ-rennes1.fr .....adresse demandeur (optionnelle)
                                     (ligne blanche = fin de l'entête HTTP de la requête)

HTTP/1.1 200 OK .....réponse du serveur
Date: Tue, 02 Jun 2001 14:11:17 GMT
Server: Apache/1.3b6
Last-Modified: Mon, 07 Apr 2001 10:39:08 GMT .....informations sur la ressource
ETag: "b3dd-524-33b78ccc"
Content-Length: 1316 .....taille de la ressource
Accept-Ranges: bytes
Content-Type: text/html .....type MIME
                                     (ligne blanche = fin de l'entête HTTP de la réponse)
<DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"> .....(contenu)
<HTML>
...
</HTML>
Connection closed by foreign host. ....fermeture de la connexion
```



## Les nouveautés de HTTP 1.1

- Identification obligatoire du *hostname* par le client
  - « *virtual hosting* »
- Négociation du contenu
  - documents français/anglais, HTML/PDF
- Connexions persistantes
  - amélioration temps de chargement de pages avec images
- Prise en charge des mandataires (*proxies*)
- Nouveaux codes réponse
- Nouvelles méthodes



## Les méthodes HTTP

- Demander une ressource
  - **GET** : rapatrier une ressource
  - **POST** : envoyer des données et rapatrier une ressource
- Avoir de l'information sur une ressource
  - **HEAD** : connaître ses caractéristiques
  - **OPTIONS** : connaître les options qui lui sont applicables
- Mettre à jour une ressource à distance
  - **PUT** : la créer ou remplacer son contenu
  - **DELETE** : la détruire
- Déboguer
  - **TRACE** : tracer les mandataires (*proxies*)





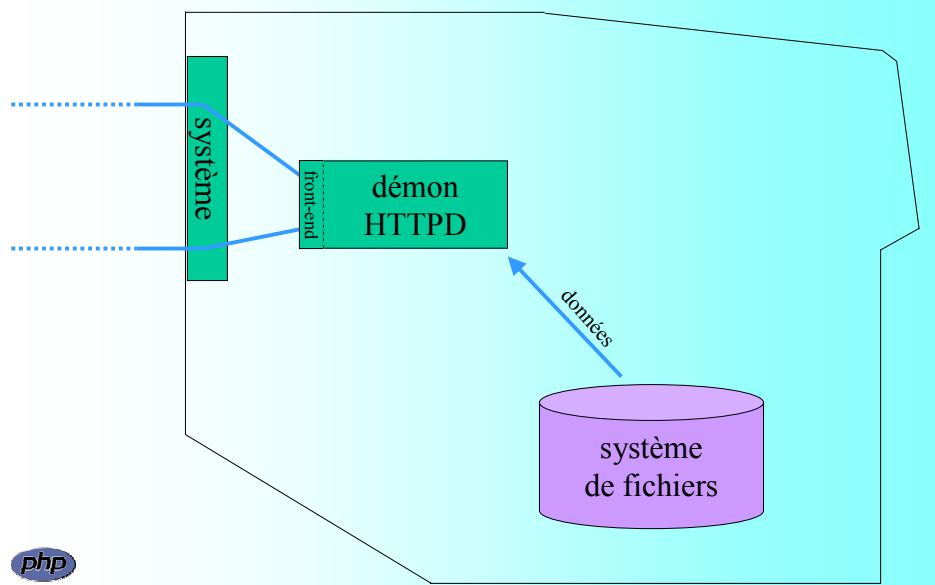
# Les codes de réponse sous HTTP

- 10x : information
- 20x : succès                    200 OK
- 30x : redirection            301 Moved permanently
- 40x : erreur du client        401 Unauthorized  
   402 Payment required  
   403 Forbidden  
   404 Not found
- 50x : erreur du serveur      500 Internal server error

Formation à PHP version 2.1 - Copyright © 1999 - 2002 Pascal AUBRY - FSC - Université de Rennes 1



# Délivrer un document statique



Formation à PHP version 2.1 - Copyright © 1999 - 2002 Pascal AUBRY - FSC - Université de Rennes 1



## Pourquoi développer sur le web ?

- Universalité du protocole HTTP
  - indépendance vis-à-vis des clients
  - pérennité des applications
- Simplicité de l'interface
- Parce que ça faisait moderne ;-)



## Web dynamique : qui fait quoi ?

- Le serveur exécute, le client reçoit
  - SSI, XSSI, CGI, FastCGI, PHP, ASP, JSP
  - indépendance vis-à-vis du client (navigateur)
  - interactivité limitée
- Le serveur envoie, le client exécute
  - JavaScript embarqué (DHTML), Applet Java
  - dépendance vis-à-vis du client
  - plus d'interactivité



# Pages dynamiques : solutions côté serveur

Les premiers pas  
Le protocole CGI  
PHP et les autres  
(FastCGI, Perl, PHP, Servlets, JSP, ASP)



Formation à PHP version 2.1 - Copyright© 1999 - 2003 Pascal AUBRY - IFSIC - Université de Rennes 1

## SSI, XSSI : le premier pas vers la dynamique

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html>
<head><title>Exemple XSSI</title></head>
<body bgcolor="#FFFFFF">
<!--#config errmsg="erreur de syntaxe dans un (x)SSI" -->
<!--#set var="x" value="test"--><!--#echo var="x"--><br>
Fichier modifié le : <!--#echo var="LAST_MODIFIED" --> <br>
Nom du serveur : <!--#echo var="SERVER_NAME" --> <br>
<!--#if expr="$HTTP_USER_AGENT = 'Mozilla/2.0 (compatible;
MSIE 3.01; Windows 95)'" -->
  Il est temps de se mettre à jour !
<!--#endif -->
</body>
</html>
```



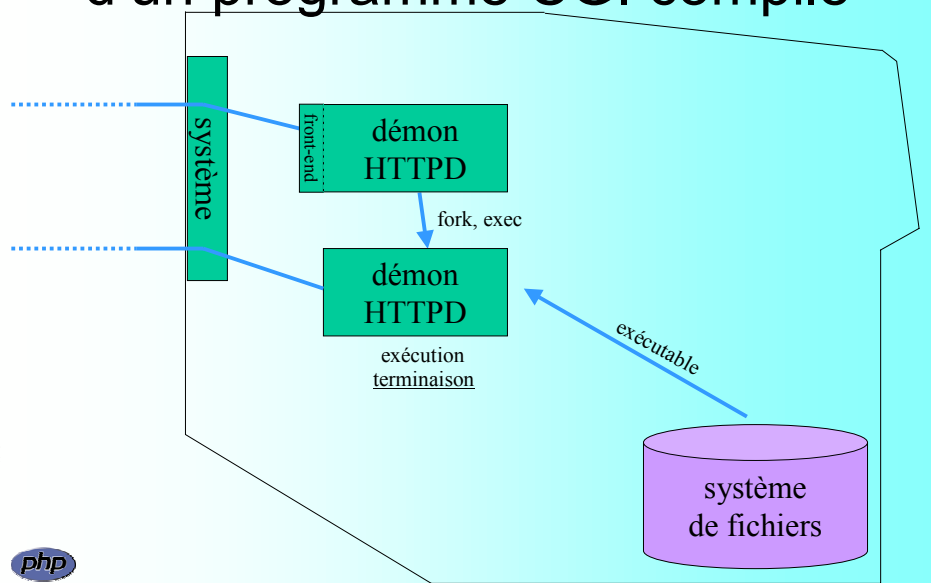
Formation à PHP version 2.1 - Copyright© 1999 - 2003 Pascal AUBRY - IFSIC - Université de Rennes 1

# CGI : Common Gateway Interface

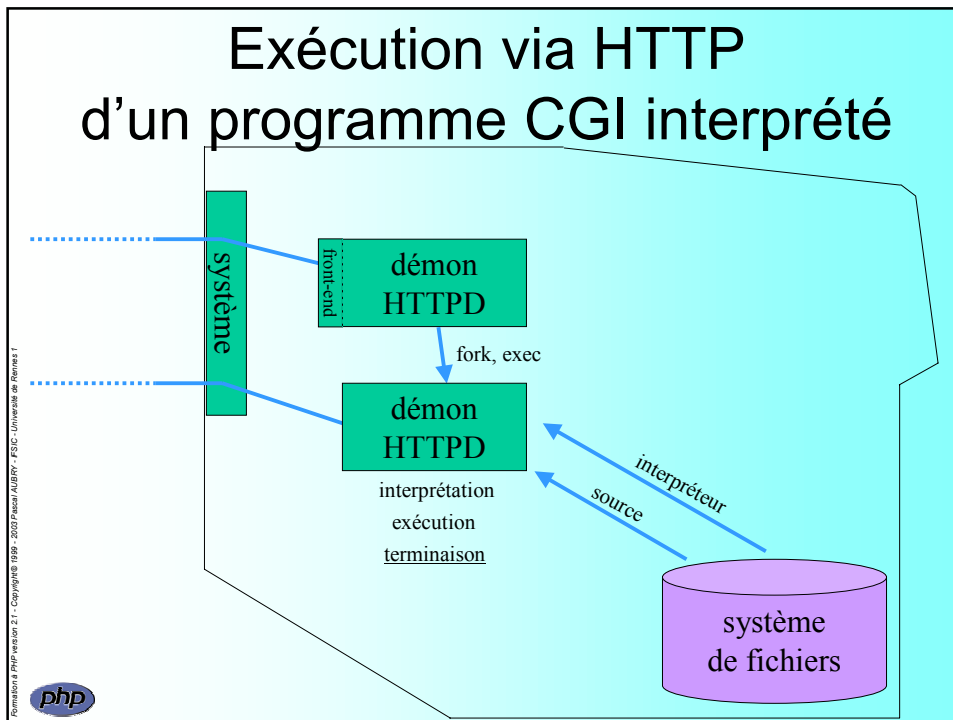
- Un standard pour l'interface entre applications et serveurs d'informations
- Permet de passer des paramètres aux requêtes
  - dans l'URL avec la méthode GET  
`http://serv.dom.org/cgi-bin/script?arg1=val1&arg2=val2`
  - comme des données avec la méthode POST
- Exécution d'un programme sur le serveur
  - Les informations renvoyées au client sont statiques
  - Des requêtes successives permettent le dynamisme



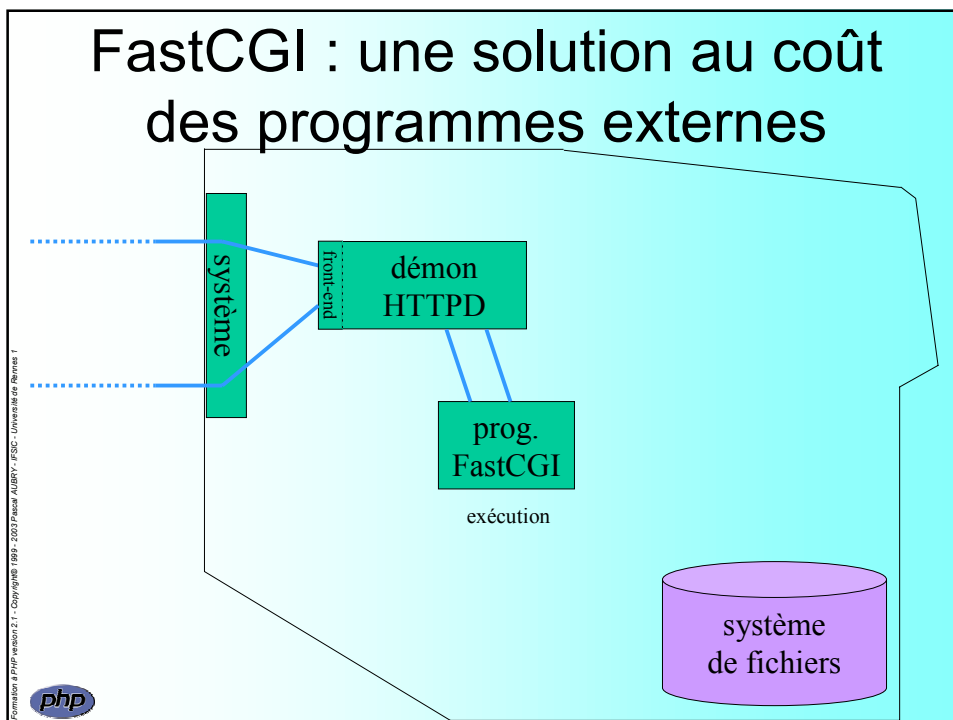
## Exécution via HTTP d'un programme CGI compilé



## Exécution via HTTP d'un programme CGI interprété



## FastCGI : une solution au coût des programmes externes



## FastCGI

- Performance, simplicité (comme CGI)
- Migration facile (de l'existant CGI)
- Isolation du démon httpd
  - sécurité en cas de crash
  - Indépendance de l'architecture du serveur
- Distribution possible de la charge

Formation à PHP version 2.1 - Copyright 1999 - 2002 Pascal Aubry - FSC - Université de Rennes 1



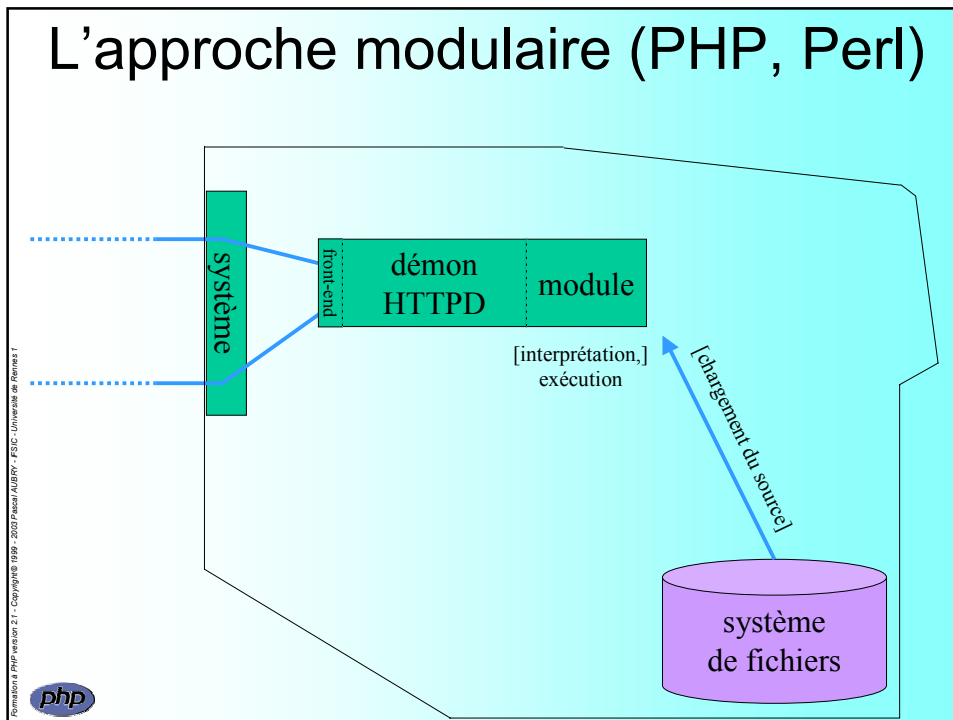
## Scripts côté serveur : les quatre leaders du marché

- Perl
- PHP (Php Hypertext Preprocessor)
- JSP (Java Server Pages)
- ASP (Active Server Pages)

Formation à PHP version 2.1 - Copyright 1999 - 2002 Pascal Aubry - FSC - Université de Rennes 1

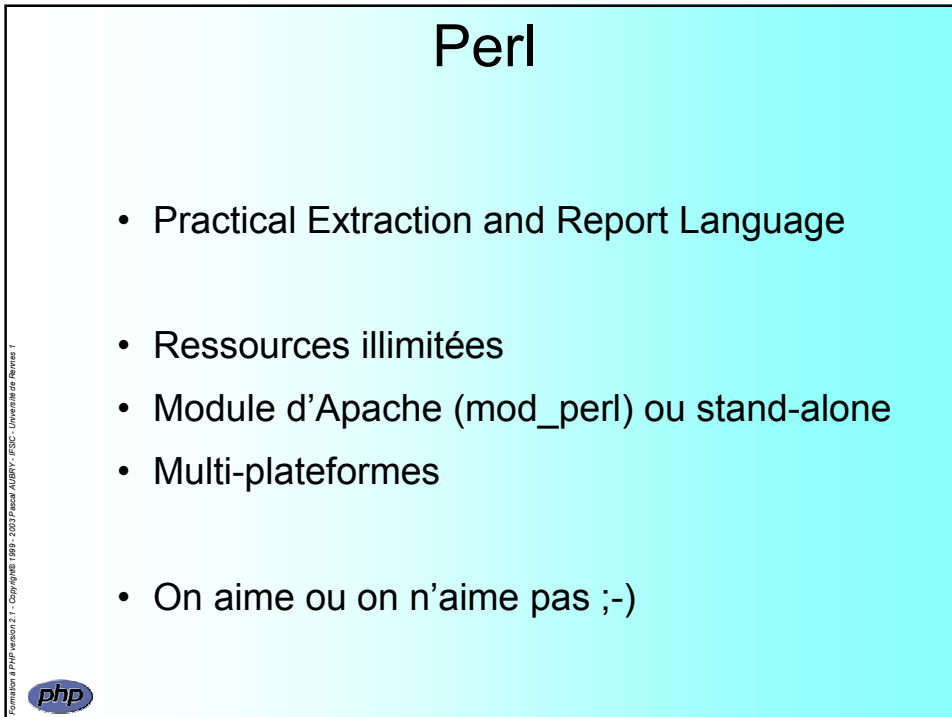


# L'approche modulaire (PHP, Perl)



## Perl

- Practical Extraction and Report Language
- Ressources illimitées
- Module d'Apache (mod\_perl) ou stand-alone
- Multi-plateformes
- On aime ou on n'aime pas ;-)



# PHP

- Le plus simple
- Multi-plateformes
- Un langage non généraliste (comme Java et Perl) mais néanmoins très riche (bibliothèques)
- Un produit libre



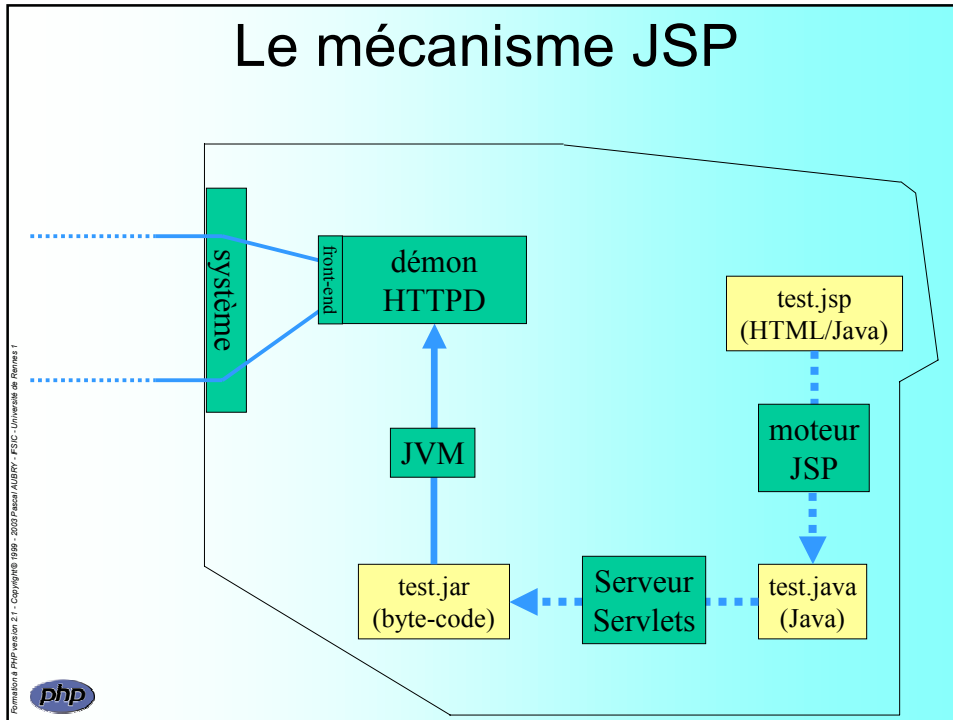
# Java

- Les servlets
  - Programme Java exécuté côté serveur
  - Transformation en byte-code avant exécution
  - Servies par un serveur dédié (ex : Tomcat)
- JSP (Java Server Pages)
  - Pages HTML avec Java embarqué
  - Transformation en Servlets





## Le mécanisme JSP



## ASP

- Tourne nativement sur IIS (MicroSoft)
- JavaScript ou VBScript
- ASP+ :
  - support langages compilés (VB, C++, C#)
  - compilation intermédiaire JS et VBS
  - mécanisme de cache des objets compilés
  - configuration format XML
  - contrôle accru des formulaires
- portabilité limitée ([www.chilisoft.com](http://www.chilisoft.com))

## Les langages de CGI

- Les shells
  - sh, tcsh, bash
- Les langages compilés
  - C, C++, Pascal, ...
- Perl
- Java (JSP, Servlets)
- PHP
- Python
- VBScript, JavaScript (ASP)

Formation à PHP version 2.1 - Copyright © 1999 - 2002 Pascal AUBRY - FSC - Université de Rennes 1



## Un site web : des responsabilités, des métiers

- Technique (ingénieur)
  - fonctionnement du service
  - mise à disposition d'outils
  - développement
- Éditoriale (rédacteur)
  - mise à jour du contenu
- Graphique (infographiste)
  - présentation du site
- Pénale (juriste)

Formation à PHP version 2.1 - Copyright © 1999 - 2002 Pascal AUBRY - FSC - Université de Rennes 1



## Pourquoi un environnement de développement ?

- Pour gérer l'accès concurrent
  - les droits sur les systèmes de fichiers sont insuffisants
  - on utilise souvent une base de données et/ou une interface web
- Pour séparer les tâches
  - graphisme, interfaçage Homme/Machine
  - développement
  - mise à jour du contenu
- Pour intégrer plusieurs technologies
  - COM, CORBA, EJB, C/C++, Java, XML



## Les environnements de développement

- BEA WebLogic (Java)
- Lutris Enhydra (Java, open source)
- Netscape Application Server
- Jbuilder (Java)
- Zope (Python)
- Midgard (PHP, open source)
- Allaire/Macromedia ColdFusion 5 (CFML)
- Vignette Content Suite V6
- ...



## Une comparaison rapide...

	[Java VB Script (+ASP)	Java (+JSP)	Perl (+mod_perl)	PHP	Python (+Zope)
Installation	★★★★★	★★★	★★★★★	★★★★★	★★★★★
Apprentissage	★★★	★★	★	★★★★★	★★★★★
Puissance	★★★	★★★★★	★★★	★★★	★★★
Portabilité	★	★★★★★	★★★★★	★★★	★★★
Outils	★★★★★	★★★★★	★★	★★★	★★★
Ressources	★★★	★★★★★	★★★★★	★★★★★	★★★

...et subjective !



Formation à PHP version 2.1 - Copyright © 1999 - 2002 Pascal Aubry - F3SC - Université de Rennes 1

## Comment choisir ?

- 1. Évaluer le ou les projets à mettre en œuvre
- 2. Évaluer ses moyens
  - humains
  - financiers
- 3. Tenir compte de l'existant
  - les habitudes
  - les compétences
  - les préférences
- 4. Se faire soi-même une idée des produits



Formation à PHP version 2.1 - Copyright © 1999 - 2002 Pascal Aubry - F3SC - Université de Rennes 1

# Présentation de PHP

Outils / projets  
Avantages / inconvénients  
Des pages aux composants



Formation à PHP version 2.1 - Copyright© 1999 - 2003 Pascal AUBRY - IFSIC - Université de Rennes 1

## PHP en quelques mots

- Un langage
  - qui comprend CGI
  - complet
  - simple
  - hérité de Perl, C et sh
- Un module d'Apache
  - performant
  - existe aussi en stand-alone
  - en pleine évolution
    - au départ pour les pages personnelles ;-)
    - aujourd'hui pour les applications sur le web

Formation à PHP version 2.1 - Copyright© 1999 - 2003 Pascal AUBRY - IFSIC - Université de Rennes 1



## Outils

- Mise au point
  - debugger intégré dans la distribution PHP
- Édition
  - PhpEd, PhpCoder, Xemacs
- Développement
  - Midgard, PHPGem, phpDBForm, phpDB
- Documentation
  - Doxygen, PHPXref, PhpDocumentor
- Passerelle ASP
  - asp2php

Formation à PHP version 2.1 - Copyright © 1999 - 2003 Pascal AUBRY - F3SC - Université de Rennes 1



## Projets

- Parmi les plus connus...
  - IMP (interface IMAP/SMTP)
  - W-Agora (forums)
  - phpMyAdmin (administration MySQL)
  - PHPGem (création de formulaires)
  - IRM (gestion de parcs matériels et logiciels)
  - Bookmarker (gestion de favoris)
  - PHP3Domo (interface Majordomo)
  - xml2tree (gestion de documents XML)
  - SEARCpHp (moteur de recherche)
  - phpGroupWare (travail coopératif)
  - ...

Formation à PHP version 2.1 - Copyright © 1999 - 2003 Pascal AUBRY - F3SC - Université de Rennes 1



# Avantages

- Portable (multi-plateforme)
- Simple
  - apprentissage rapide
  - nombreux exemples
- Rapide
  - cycle de développement très court
  - exécution rapide (module d'Apache, Zend)
- Riche (bibliothèques)
- Évolutif
- Libre (GPL)

Formation à PHP version 2.1 - Copyright © 1999 - 2003 Pascal AUBRY - FSC - Université de Rennes 1



# Inconvénients

- Interactivité limitée
  - HTTP
  - CGI
- Protection du code

Formation à PHP version 2.1 - Copyright © 1999 - 2003 Pascal AUBRY - FSC - Université de Rennes 1



## Un programme PHP, c'est...

- Un fichier HTML...
- dans lequel on trouve des tags PHP :

```
<html>
<head><title>TEST</title></head>
<body>
  <p>
    Il est <?echo date("H:i");?>.
  </p>
</body>
</html>
```



## Dans la pratique... (surtout pour les gros projets)

- Un programme PHP est un ensemble d'instructions PHP qui affichent :
  - du code HTML
  - ou autre chose :
    - ascii
    - PostScript, PDF
    - GIF, JPEG, PNG, ...
- L'approche « pages » est désormais supplantée par l'approche « composants »





## Une page HTML avec des instructions PHP

```
<html>
  <head>
    <title>
      Bonjour !
    </title>
  </head>
  <body>
    <p>
      sur le serveur, il est exactement
      <? echo date("H:i:s") ; ?>
    </p>
  </body>
</html>
```



## Un programme PHP

```
class sortie {
  function sortie($titre) // constructeur
  { $this->titre = $titre ; }
  function debut()
  { echo "<html><head>$this->titre</head><body>" ; }
  function fin()
  { echo "</body></html>" ; }
}

$s = new sortie("Bonjour !") ;
$s->debut() ;
echo "<p>sur le serveur, il est exactement "
    .date("H:i:s")
    ."</p>" ;
$s->fin() ;
```



# Le langage PHP

## Bases

Variables / Constantes  
Expressions / Opérateurs  
Structures de contrôle  
Fonctions / Inclusions de fichiers



Formation à PHP version 2.1 - Copyright© 1999 - 2003 Pascal AUBRY - IFSIC - Université de Rennes 1

## Syntaxe de base

- Séparateur d'instructions : point-virgule

```
<? instruction1 ;  
    instruction2 ?>
```

- Commentaires

– à la C :

```
/* commentaire */
```

– à la C++ :

```
// fin de ligne commentée
```

– à la shell :

```
# fin de ligne commentée
```



Formation à PHP version 2.1 - Copyright© 1999 - 2003 Pascal AUBRY - IFSIC - Université de Rennes 1

## Entiers et réels

- Entiers

```
1234      # décimal
-123     # négatif
0123     # octal (=83)
0x12     # hexadécimal (=18)
```

- Flottants (réels)

```
1.234
1.2e3     # (=1200)
```



## Chaînes de caractères

- Entre guillemets :

- variables remplacées
- `\n`, `\r`, `\t`, `\\`, `\$`, `\"`, `\167`, `\x5f`

- Entre apostrophes :

- variables non remplacées
- `\\`, `\'`

- Chaînes «here doc» :

```
$str = <<<EOS
...
EOS
```

- Indices de 0 à l-1 (`$str[0]` à `$str[$l-1]`)



## Conversion de chaînes

- On prend le premier mot de la chaîne
  - si mot numérique [-]n[.n][en], la valeur
  - sinon 0

- Exemples :

1 + "10.5"	11.5
1 + "-1.3e3"	-1299
1 + "bob-1.3e3"	1
1 + "bob3"	1
1 + "10 cochons"	11
"10.0 cochons" + 1	11
"10.0 cochons" + 1.0	11.0



## Tableaux

- Scalaires ou associatifs  
`$tab[12]`, `$tab["foo"]`, `$tab[$n]`
- Multi-dimensionnels  
`$coord["top"][$x][$y]`
- Création :  
`array()`, `list()`, assignation directe
- Accès :  
`count()`, `each()`, `next()`, `prev()`
- Tri :  
`asort()`, `arsort()`, `ksort()`, `usort()`, ...



## Tableaux (création)

- Directe

- `$stab[0] = "bonjour" ;`
- `$t["café"] = 2 ;`
- `$x[] = "nouvelle entrée" ;`

- Avec la fonction `array()`

- `$dejeuner = array(`
  - `"entrée" => "salade",`
  - `"plat" => "choucroute",`
  - `"boisson" => "eau",`
  - `"dessert" => "flan",`
  - `"café" => TRUE) ;`



## Tableaux (accès, tri)

- Direct

- `$stab["personnel"][4]`

- Parcours

- `while ( list($cle,$valeur) = each($dejeuner) )`  
`echo "$cle : $valeur<br>" ;`
- Foreach (cf plus loin)

- Tri

- `ksort($dejeuner) ;`  
`reset($dejeuner) ;`  
`while ( list($cle,$valeur) = each($dejeuner) )`  
`echo "$cle : $valeur<br>" ; // trié selon la clé`



# Typage

- Le type d'une expression est déterminé par le contexte dans lequel elle est utilisée
- Il peut être forcé  
`settype()`
- Type casting traditionnel  
`$entier = (integer) 1.1 // (=1)`
- Il n'existe pas de type booléen dédié
  - 0 est faux,
  - non nul est vrai



# Variables

- Nom =  $\$(\text{lettre} | \_)(\text{lettre} | \text{chiffre} | \_)^*$   
`$vingtaine = 20 ; // ok`  
`$20n = 20 ; // erreur`  
`$_20n = 20 ; // ok`
- Case-sensitive
- Références (~alias, ~pointeur)  
`$foo = "bonjour" ;`  
`$bar = &$foo ;`  
`$bar = "au revoir" ;`  
`echo $foo ; // au revoir`



# Variables prédéfinies

- Visibles par `phpinfo()`
- Apache
- Environnement
  - selon l'O.S.
  - attention ! . remplacés par des \_
- PHP (dont paramètres CGI)

Formation à PHP version 2.1 - Copyright© 1999 - 2003 Pascal AUBRY - F3C - Université de Rennes 1



## 1. Configuration du serveur

<b>System</b>	Linux php.ifsic.univ-rennes1.fr 2.4.2-2 #1 Sun Apr 8 20:41:30 EDT 2001 i686 unknown
<b>Build Date</b>	Sep 10 2001
<b>Configure Command</b>	<code>./configure '--prefix=/usr' '--with-config-file-path=/etc' '--disable-debug' '--enable-pic' '--enable-shared' '--enable-inline-optimization' '--with-apxs=/usr/sbin/apxs' '--with-exec-dir=/usr/bin' '--with-regex=system' '--with-gettext' '--with-gd' '--with-jpeg-dir=/usr' '--with-png' '--with-zlib' '--with-db2' '--with-db3' '--with-gdbm' '--enable-debugger' '--enable-magic-quotes' '--enable-safe-mode' '--enable-sockets' '--enable-syssem' '--enable-sysvshm' '--enable-track-vars' '--enable-yp' '--enable-ftp' '--enable-wddx' '--without-mysql' '--without-oracle' '--without-oci8' '--with-xmli' '--enable-trans-sid'</code>
<b>Server API</b>	Apache
<b>Virtual Directory Support</b>	disabled
<b>Configuration File (php.ini) Path</b>	/etc
<b>ZEND_DEBUG</b>	disabled
<b>Thread Safety</b>	disabled

This program makes use of the Zend scripting language engine:  
Zend Engine v1.0.4, Copyright (c) 1998-2000 Zend Technologies

Formation à PHP version 2.1 - Copyright© 1999 - 2003 Pascal AUBRY - F3C - Université de Rennes 1



## Portée des variables

- Par défaut, une variable est locale
- Pour accéder à une variable globale depuis une fonction :
  - `global $a ;`
  - `$GLOBALS["a"]`



## Super-variables (PHP 4.3)

- Toujours considérées comme des variables globales, sans préciser leur portée (pas de global)
  - `$_GLOBALS` : toutes les variables globales
  - `$_SERVER` : les variables transmises par Apache
  - `$_GET` : les paramètres CGI passés en GET
  - `$_POST` : les paramètres CGI passés en POST
  - `$_COOKIE` : les cookies
  - `$_FILES` : les fichiers uploadés
  - `$_ENV` : les variables d'environnement
  - `$_REQUEST` : `$_GET` + `$_POST` + `$_COOKIE`
    - Cf `variables_order`
  - `$_SESSION` : les variables de session





## Portée des variables (exemple)

```
$a = "bonjour" ;  
$b = "hello" ;  
  
function f($x)  
{  
    global $b ;  
  
    echo $a ;           // rien  
    echo $b ;           // hello  
    echo $GLOBALS["a"] ; // bonjour  
}
```



## Variables de variables (exemple)

```
$a = "bonjour" ;  
  
$$a = "monsieur" ;  
  
echo "$a ${$a}" ;    // bonjour monsieur  
  
echo "$a $bonjour" ; // idem
```



## Propriétés des variables

- Existence
  - `isset()`
- Type
  - `gettype()`
  - `is_long()`, `is_double()`, `is_string()`
  - `is_array()`, `is_object()`
- Contenu
  - `empty()`



## Constantes

- Variables à assignation unique
  - `define("MA_CONSTANTE", 45) ;`
- Quelques constantes prédéfinies
  - `__CLASS__`, `__FUNCTION__`, `__FILE__`, `__LINE__`, `PHP_VERSION`, `TRUE`, `FALSE`, ...

```
function affiche_erreur($fichier,$ligne,$message)
{
    echo "<p>$fichier, ligne $ligne : $message</p>" ;
}

affiche_erreur(__FILE__, __LINE__, "aie !") ;
```



# Expressions

- En PHP, tout est expression (comme en C)
- Conséquences :
  - on peut trouver 100 décimales de  $\pi$  en une seule ligne de code
  - on peut facilement perdre en lisibilité ;-)

```
int a=10000,b,c=8400,d,e,f[8401],g;
main()
{
    for(;b-c;) f[b++]=a/5;
    for(;d=0,g=c*2;c -= 14,printf("%.4d",e+d/a),e=d%a)
        for (b=c;d += f[b]*a,f[b]=d% --g,d /=g--, --b; d *=b) ;
}
```



# Opérateurs(1)

- Arithmétiques
  - +, -, \*, / (quotient), % (reste)
- Binaires (bit à bit)
  - & (et), | (ou), ^ (ou excl.), ~ (non), << et >> (décalages)
- Comparaison
  - == (égal), === (identique), !=, >, <, <=, >=



## Opérateurs(2)

- Incrémentation (++), décrémentation (--)

```
$a = 5 ;  
$b = 2 ;  
$c = (2 * --$a) % $b-- ;
```

- Logiques

```
if ( ($a && !$b) || $c )  
    ...
```

- Caractères

. (composition)



## Opérateurs(3)

- Exécution

```
$sortie = `ls -al` ;  
echo "<pre>$sortie</pre>" ;
```

- Assignations composées

+=, -=, \*=, /=, %=, .=, &=, |=, ^=, ~=, <<=, >>=

- Alternative

```
expr_bool ? res_si_vrai : res_si_faux
```



## Opérateurs (associativité)

- Tous les opérateurs sont associatifs à gauche, sauf exception ;-)
- Associatifs à droite  
`print, !, ~, ++, --, (type), [, @`
- Non associatifs  
`==, !=, ===, new`



## Opérateurs (priorité)

- Par ordre de priorité décroissante

<code>,</code>	<code>^</code>
<code>or</code>	<code>&amp;</code>
<code>xor</code>	<code>== != ===</code>
<code>and</code>	<code>&lt;&lt;= &gt;&gt;=</code>
<code>print</code>	<code>&lt;&lt; &gt;&gt;</code>
<code>= += -= *= /=</code>	<code>+ - .</code>
<code>.= %=</code>	<code>* / %</code>
<code>&amp;=  = ^= ~=</code>	<code>! ~++ --</code>
<code>&lt;&lt;= &gt;&gt;=</code>	<code>(type) @</code>
<code>? :</code>	<code>[</code>
<code>  </code>	<code>new</code>
<code>&amp;&amp;</code>	
<code> </code>	



## Structures de contrôle

- Conditionnelle
- Alternative / alternative multiple
- Choix multiple
- Boucles
  - « tant que »
  - « répéter »
  - « pour »
  - « pour chaque »

Formation à PHP version 2.1 - Copyright © 1999 - 2002 Pascal AUBRY - F3SC - Université de Rennes 1



## Alternative (multiple)

```
if (condition1)
{
    /* . . . */
}
elseif (condition2)
{
    /* . . . */
}
elseif . . .
. . .
else
{
    /* . . . */
}
```

Formation à PHP version 2.1 - Copyright © 1999 - 2002 Pascal AUBRY - F3SC - Université de Rennes 1



## Tant que / répéter

```
while (condition)
{
    /* . . . */
}

do
{
    /* . . . */
}
while (condition) ;
```



## Boucle « pour »

```
for ( expr1 ; expr2 ; expr3 )
{
    /* . . . */
}
```

- est équivalent à

```
expr1 ;
while ( expr2 )
{
    /* . . . */
    expr 3 ;
}
```



## Boucle « pour chaque » (1)

```
foreach ( arr_expr as $valeur )
{
    echo "valeur : $valeur<br>\n" ;
}
```

- est équivalent à :

```
reset(arr_expr) ;
while (list(,$valeur) = each(arr_expr))
{
    echo "valeur : $valeur<br>\n" ;
}
```



## Boucle « pour chaque » (2)

```
foreach ( arr_expr as $cle => $valeur )
{
    echo "clé : $cle<br>\n" ;
    echo "valeur : $valeur<br>\n" ;
}
```

- est équivalent à :

```
reset(arr_expr) ;
while (list($cle,$valeur) = each(arr_expr))
{
    echo "clé      : $cle<br>\n" ;
    echo "valeur : $valeur<br>\n" ;
}
```





## Choix multiple

```
switch (expr)
{
  case val1:
    /* . . . */
    break ;
  case val2:
    /* . . . */
    break ;
  . . .
  default:
    /* . . . */
}
```



## Ruptures de séquence

- **continue**
  - arrêter l'itération courante
  - passer à la suivante
- **break [n]**
  - arrêter l'itération courante
  - sortir de la boucle courante  
(ou des n boucles imbriquées)



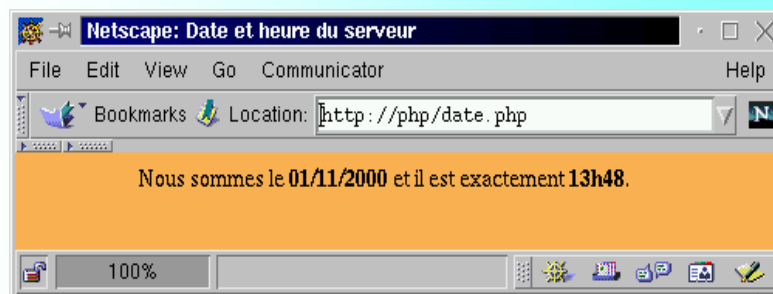
## Fonctions (exemple)

```
function exemple ($arg_1,  
                 $arg_2,  
                 ...  
                 $arg_n)  
{  
    echo "dans exemple<br>\n" ;  
    return $val_ret ;  
}
```

- Notes :
  - peut retourner une valeur
  - peut être appelée sans affectation (comme en C)



### 2. Utilisation de fonctions



## Fonctions : valeur par défaut des arguments

```
function augmente($valeur,$ajout=1)
{
    return $valeur + $ajout ;
}

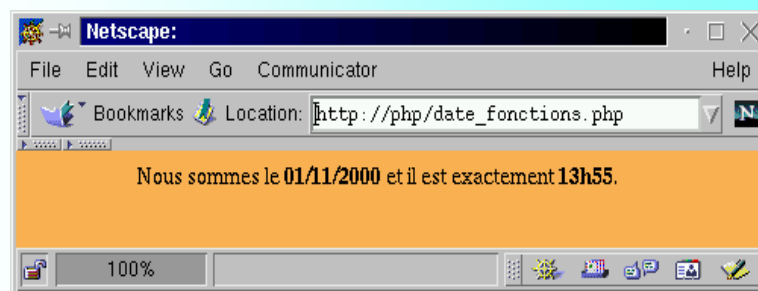
$x = 5 ;

echo augmente($x,4) . "<br>" ;           // 9
echo augmente($x) . "<br>" ;           // 6
```

- argument vide  $\neq$  argument absent



### 3. Écriture de fonctions



## Fonctions : passage des arguments par référence

```
$a = $b = 0 ;           // initialisation de $a et $b

function f($p1,&$p2)    // déclaration de la fonction f
{
    $p1 = $p2 = 5 ;
}

f($a,$b) ;            // appel de la fonction f

// affichage des valeurs de $a et $b
echo "\$a=$a, \$b=$b<br>\n" ;
```

Formation à PHP version 2.1 - Copyright © 1999 - 2002 Pascal AUBRY - FSC - Université de Rennes 1

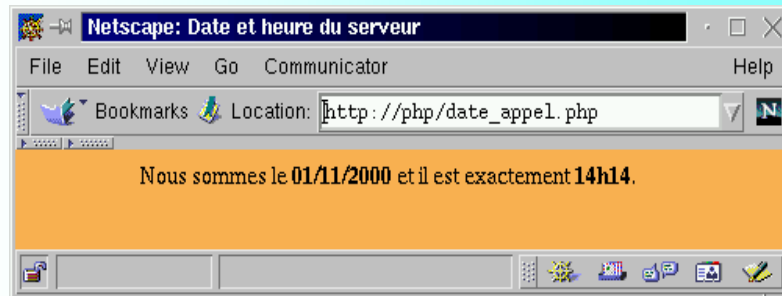


## Inclusion de fichiers

- **require**("fichier.inc") ;
  - inclut le fichier fichier.inc
  - évaluation en pre-processing
- **include**("fichier.inc") ;
  - inclut le fichier fichier.inc
  - évaluation à chaque fois
- Inclusion unique
  - **include\_once**()
  - **require\_once**()

Formation à PHP version 2.1 - Copyright © 1999 - 2002 Pascal AUBRY - FSC - Université de Rennes 1





Formation à PHP version 2.1 - Copyright© 1999 - 2003 Pascal AUBRY - IFSIC - Université de Rennes 1



# Techniques de programmation

Bibliothèques

Formulaires HTML (rappels)

Paramètres CGI

Programmation Orientée Objet



Formation à PHP version 2.1 - Copyright© 1999 - 2003 Pascal AUBRY - IFSIC - Université de Rennes 1

## Bibliothèques (1)

- Bases de données
- Vérification d'orthographe (aspell)
- Précision arbitraire (bcmath)
- Calendrier
- Date/heure
- dbm (Berkeley data base manager)
- Système de fichier
- Chargement dynamique de fonctions
- Exécution de programmes externes



## Bibliothèques (2)

- HTTP
- Manipulation d'images (gd)
- IMAP (Internet Mail Access Protocol)
- LDAP (Lightweight Directory Access Protocol)
- Courrier électronique
- Mathématiques
- Réseau (socket, ...)
- PDF (Portable Document Format)



## Bibliothèques (3)

- Expressions régulières
- Sémaphores, mémoire partagée
- SNMP (Simple Network Management Protocol)
- Chaînes de caractères
- Cryptage, codage
- Compression, décompression
- XML (eXtended Markup Language)
- Paiement électronique



## Bibliothèques (4)

- Interface aux bases de données
  - Adabas D
  - filePro
  - Informix
  - mSQL
  - MySQL
  - Oracle 8
  - PostgreSQL
  - ODBC
  - dBase
  - Hyperwave
  - InterBase
  - MS SQL Server
  - Oracle 7
  - SyBase
  - Solid



## Formulaires HTML (rappels)

- Exemple :
  - ```
<form action="prog.php" method="get">  
  <input type="submit" name="go">  
</form>
```
- Action : URL cible
- Method : dans la pratique GET ou POST



## Entrées des formulaires HTML (1)

- Boutons d'action
  - ```
<input type="submit" name="nom" value="valeur">
```
  - ```
<input type="reset" name="nom" value="valeur">
```
  - ```
<input type="image" name="nom" src="source">
```
- Entrées textuelles
  - ```
<input type="text" name="nom" value="valeur">
```
  - ```
<input type="hidden" name="nom" value="valeur">
```
  - ```
<input type="password" name="nom" value="valeur">
```
- Boîtes de saisie
  - ```
<textarea name="nom" rows="r" cols="c">  
  texte...  
</textarea>
```





## Entrées des formulaires HTML (2)

- Listes à choix

```
<select [multiple] name="nom">
  <option value="valeur">texte</option>
  ...
</select>
```

- Boutons radio

```
<input type="radio" name="nom"
       value="valeur" [selected] >
```

- Boutons à cocher

```
<input type="checkbox" name="nom"
       value="valeur" [checked] >
```



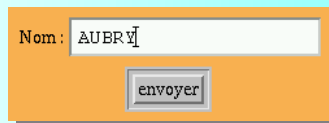
## Paramètres CGI (1)

- Ne sont plus des variables globales (PHP 4.3)
- Se trouvent dans les super-variables `$_GET` et `$_POST` (et `$_REQUEST`)
- Peuvent être des tableaux



## Paramètres CGI (2)

```
<form action="exemple.php" method="get">  
Nom : <input type="text" name="nom"><br>  
<input type="submit" name="bouton"  
      value="envoyer">  
</form>
```



Nom : AUBRY

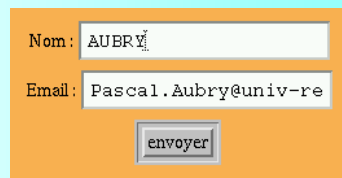
envoyer

- Dans exemple.php
  - \$REQUEST["bouton"] vaut "envoyer"
  - \$REQUEST["nom"] vaut ce qu'il y a dans le dialogue associé



## Paramètres CGI (3)

```
<form action="tab.php" method="get">  
Nom : <input type="text" name="p[nom]"><br>  
Email : <input type="text" name="p[email]"><br>  
<input type="submit" name="bouton"  
      value="envoyer">  
</form>
```



Nom : AUBRY

Email : Pascal.Aubry@univ-re

envoyer

- Dans tab.php
  - \$REQUEST["bouton"] vaut "envoyer"
  - \$REQUEST["p"]["nom"] et \$REQUEST["p"]["email"] valent ce qu'il y a dans les dialogues associés



## Paramètres CGI (4)



```
<form action="selection.php" method="get">
  <select multiple name="selection[]">
    <option value="x">Mlle X</option>
    <option value="y">Mme Y</option>
    <option value="z">M. Z</option>
  </select>
  <input type="submit" name="bouton"
    value="envoyer">
</form>
```

- Dans selection.php
  - `$REQUEST["bouton"]` vaut "envoyer"
  - `$REQUEST["selection"]` a pour taille le nombre d'options sélectionnées et pour valeurs les options sélectionnées



## Paramètres CGI (5)

```
<form action="image.php" method="post">
  Cliquez sur l'image :<br>
  <input type="image" src="image.gif" name="bout">
</form>
```



- Dans image.php
  - `$REQUEST["bout_x"]` et `$REQUEST["bout_y"]` ont pour valeur les coordonnées du point de l'image sur lequel l'utilisateur a cliqué



# Exemple

```
<form action="test_cgi.php" method="GET">
  Texte :<br>
  <input type="text" name="texte_court" value="blabla"><hr>
  Sélection simple :<br>
  <select name="sel_simple">
    <option value="1">Choix 1</option>
    <option value="2">Choix 2</option>
    <option value="3">Choix 3</option>
  </select><hr>
  Sélection multiple :<br>
  <select multiple name="sel_multiple[]">
    <option value="1">Choix 1</option>
    <option value="2">Choix 2</option>
    <option value="3">Choix 3</option>
  </select><hr>
  <input type="reset" name="bouton_reset" value="Annuler">
  <input type="submit" name="bouton_submit" value="Valider">
</form>
```



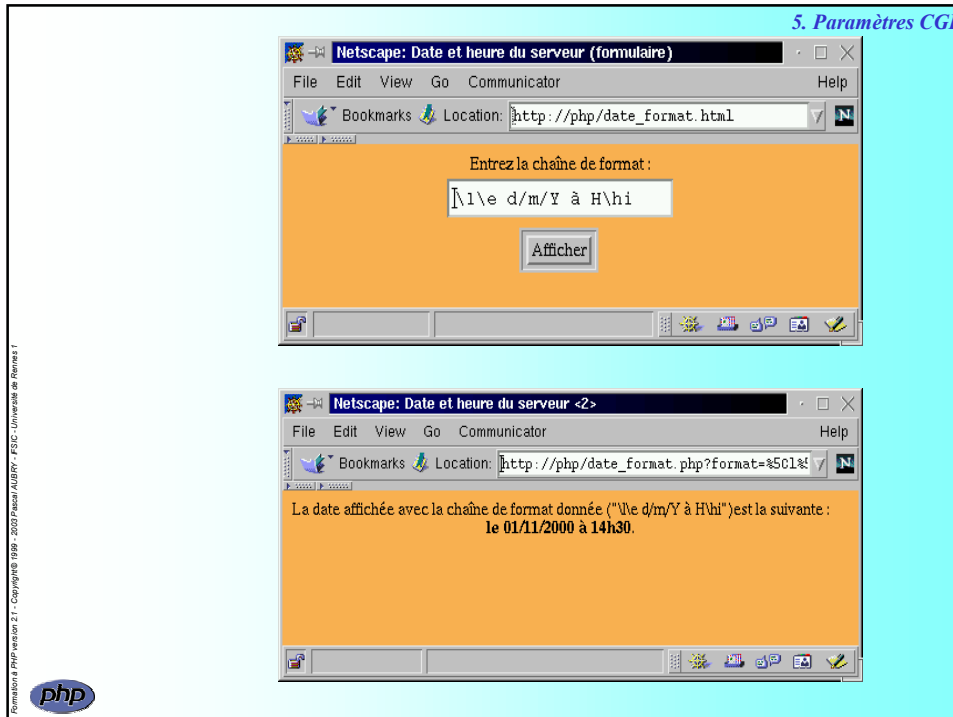
# Exemple

**\$\_REQUEST**

- `<input type="text" name="texte_court" value="blabla">` → `'texte_court' => "blabla"`
- `<select name="sel_simple">...</select>` → `'sel_simple' => "2"`
- `<select multiple name="sel_multiple[]">...</select>` → `'sel_multiple' => { [0] => "1", [1] => "3" }`
- `<input type="reset" name="bouton_reset" value="Annuler">` → `'bouton_reset' => "Annuler"`
- `<input type="submit" name="bouton_submit" value="Valider">` → `'bouton_submit' => "Valider"`

The browser window shows the form with the text field containing 'blabla', the simple selection dropdown set to 'Choix 2', and the multiple selection checkboxes for 'Choix 1' and 'Choix 3' checked. The 'Annuler' and 'Valider' buttons are visible at the bottom.





## Classes (1)

- Une classe est un ensemble de variables et de fonctions travaillant sur ces variables
- Un objet est une instance particulière d'une classe
- POO de base, mais, grâce à l'héritage
  - réutilisabilité du code
  - modularité

## Classes (2)

- Possibilité de surcharger des méthodes (fonctions) d'une classe
- Attention !
  - pas d'appel automatique des constructeurs des classes ancêtres !
  - Pas de destructeur
- Dans une méthode, `$this` désigne l'objet dont on exécute la méthode



## Classes (définition)

- Exemple : la classe `exemple` maintient un entier `n` affiché et incrémenté à chaque appel de sa méthode `plus()`.

```
class exemple {  
  
    var $n ;                               // attributs  
  
    function exemple($init=1) // constructeur  
    { $this->n = $init ; }  
  
    function plus() // méthode  
    { echo "$this->n<br>" ; $this->n ++ ; }  
  
} // fin de la classe exemple
```



## Classes (héritage)

- Exemple : la classe `sur_exemple` possède une méthode supplémentaire `moins()` qui décrémente l'attribut

```
class sur_exemple extends exemple { // héritage

function sur_exemple($init=1) // constructeur
{ $this->exemple($init) ; }
// ! pas d'appel automatique

function moins() // méthode
{ echo "$this->n<br>" ; $this->n -- ; }

} // fin de la classe sur_exemple
```

- Pas d'héritage multiple

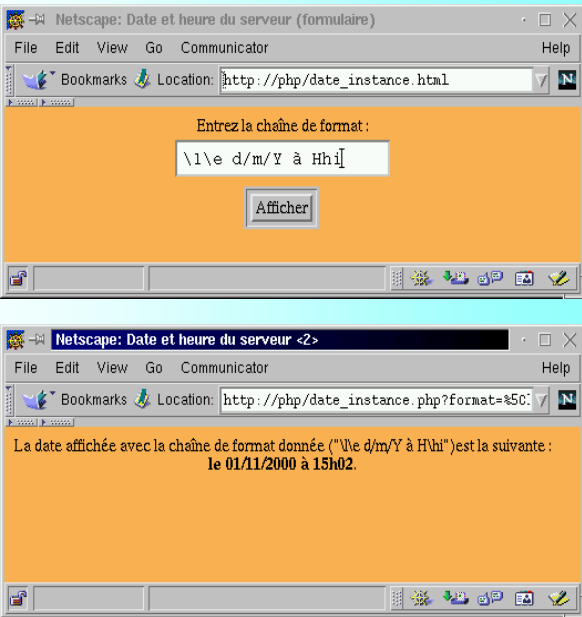


## Classe (instanciation)

```
<?
    $e = new sur_exemple(4) ;
    $e->plus() ; // 4
    $e->moins() ; // 5
    $e->plus() ; // 4
?>
```

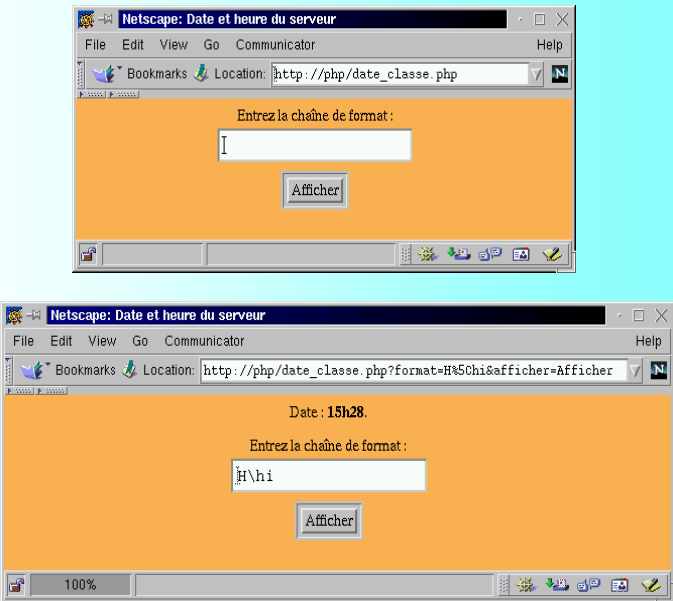


## 6. Création d'une classe



The screenshot shows two windows of Netscape Communicator. The top window, titled "Date et heure du serveur (formulaire)", displays a form with the text "Entrez la chaîne de format :" and an input field containing the string "\1\ e d/m/Y à H\hi". Below the input field is a button labeled "Afficher". The bottom window, titled "Date et heure du serveur <2>", shows the result of the form submission. The location bar indicates the URL "http://php/date\_instance.php?format=%5C". The main content area displays the text: "La date affichée avec la chaîne de format donnée (" \1\ e d/m/Y à H\hi") est la suivante : le 01/11/2000 à 15h02." A small "php" logo is visible in the bottom left corner of the slide.

## 7. Héritage



The screenshot shows two windows of Netscape Communicator. The top window, titled "Date et heure du serveur", displays a form with the text "Entrez la chaîne de format :" and an empty input field. Below the input field is a button labeled "Afficher". The bottom window, titled "Date et heure du serveur", shows the result of the form submission. The location bar indicates the URL "http://php/date\_classe.php?format=H%5Chi&afficher=Afficher". The main content area displays the text: "Date : 15h28." followed by "Entrez la chaîne de format :" and an input field containing the string "H\hi". Below the input field is a button labeled "Afficher". A small "php" logo is visible in the bottom left corner of the slide.



# Techniques avancées

Accès au système de fichiers

Gestion des erreurs

Protocole HTTP

(sorties non HTML, redirections, authentification, cookies)

Bases de données

Gestion de sessions



Formation à PHP version 2.1 - Copyright© 1999 - 2003 Pascal AUBRY - IFSIC - Université de Rennes 1

## Système de fichiers

- Toutes les fonctions classiques sont implémentées dans les bibliothèques
- Gestion des volumes
  - `diskfreespace()`
- Gestion des répertoires
  - `mkdir()`, `opendir()`, `readdir()`, `closedir()`, `dir`
- Gestion des fichiers
  - `file_exists()`, `file_is_xxx()`, `fopen()`, `fclose()`, `fputs()`, `rewind()`, `rename()`, `touch()`, `chmod()`, `flock()`, ...



Formation à PHP version 2.1 - Copyright© 1999 - 2003 Pascal AUBRY - IFSIC - Université de Rennes 1

## Parcours d'un répertoire

- Affichage des entrées d'un répertoire

```
$d = dir("/etc") ;  
echo "chemin : $d->path<br>\n" ;  
while ( $entry = $d->read() )  
{  
    echo "$entry<br>\n" ;  
}  
$d->close() ;
```



## Parcours d'un fichier local

- Affichage des utilisateurs du serveur

```
$data = file("/etc/passwd") ;  
foreach ( $data as $ligne )  
{  
    $user = strtok($ligne,":") ;  
    $passwd = strtok(":") ;  
    $uid = strtok(":") ;  
    $gid = strtok(":") ;  
    $gecos = strtok(":") ;  
    $homedir = strtok(":") ;  
    $shell = strtok(":") ;  
    echo "$user (uid=$uid) => $gecos<br>\n" ;  
}
```



# La forme et le fond

- On veut séparer le contenant du contenu
  - pour pouvoir changer le « look » de l'application sans modifier l'application elle-même
  - pour faire travailler différentes personnes (développeurs, infographistes) en même temps sur un projet
- Les solutions
  - utiliser des feuilles de style (CSS, pas toujours suffisant)
  - adopter un environnement de développement qui offre cette possibilité (Zope, Midgard, ...)
  - se servir de « templates »



# Exemple de template

```
<html>
<head><title>__TITRE__</title></head>
<body>
<h1>__TITRE__</h1>
<form action=__ACTION__ method="POST">
  date : <input type="text" name="jour" size="2" value="__JOUR__">
  / <input type="text" name="mois" size="2" value="__MOIS__">
  / <input type="text" name="annee" size="4" value="__ANNEE__">
  <select name="type">
    <option value="">-- tous --</option>
    __TYPES__
  </select>
  <input type="submit" name="go" value="chercher">
</form>
<hr>
<table border="1">
<tr><th>Chaîne</th><th>Horaire</th><th>Émission</th><th>Type</th></tr>
  __RESULTATS__
</table>
</body>
</html>
```



# Utilisation d'un template

```
function affichage($titre,$jour,$mois,$annee,$types,$resultats)
{
    // lecture du template dans un tableau
    foreach ( file("modeles/modele4.html") as $ligne )
    {
        // remplacement des tokens du template par les valeurs à afficher
        $ligne = str_replace("__TITRE__",$titre,$ligne) ;
        $ligne = str_replace("__ACTION__",$_SERVER["PHP_SELF"],$ligne) ;
        $ligne = str_replace("__JOUR__",$jour,$ligne) ;
        $ligne = str_replace("__MOIS__",$mois,$ligne) ;
        $ligne = str_replace("__ANNEE__",$annee,$ligne) ;
        $ligne = str_replace("__TYPES__",$types,$ligne) ;
        $ligne = str_replace("__RESULTATS__",$resultats,$ligne) ;
        echo $ligne ;
    }
}
```



Formation à PHP version 2.1 - Copyright© 1999 - 2003 Pascal AUBRY - F3SC - Université de Rennes 1

# Place aux infographistes...

- Une image cliquable pour soumettre les requêtes :

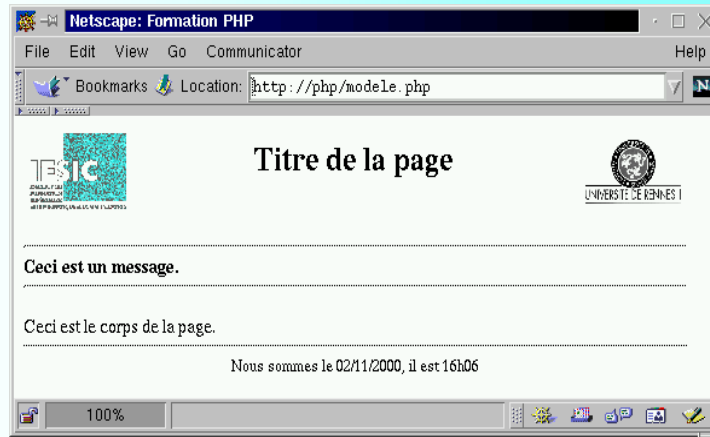
```
<input type="image"
        src="images/submit.png"
```

- Un fond d'écran créé avec Gimp

- Des polices et des couleurs agréables



Formation à PHP version 2.1 - Copyright© 1999 - 2003 Pascal AUBRY - F3SC - Université de Rennes 1



Formation à PHP version 2.1 - Copyright © 1999 - 2003 Pascal AUBRY - FSC - Université de Rennes 1



## Gestion des erreurs (1)

- Il y a plusieurs niveaux d'erreurs
  - gérés par la fonction `error_reporting()`
- Les erreurs rencontrées par PHP et non filtrées par `error_reporting` sont envoyées sous forme de messages directement sur la sortie standard des programmes
- On peut loguer les erreurs (`error_log()`)
- On peut générer des erreurs (`trigger_error()`)

Formation à PHP version 2.1 - Copyright © 1999 - 2003 Pascal AUBRY - FSC - Université de Rennes 1



## Gestion des erreurs (2)

- L'évaluation de toute expression précédée de l'opérateur @ ne génère aucun message
  - utiliser @ quand une erreur est envisageable
  - tester `$php_errormsg`

- Exemple

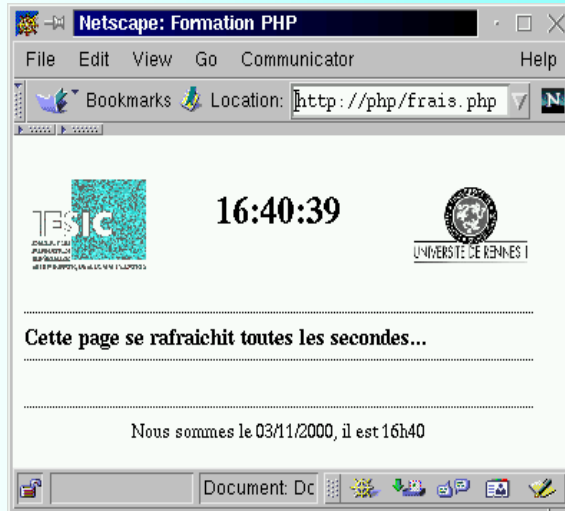
```
$c = @mysql_connect("sqlhost","guest") ;
if ( !$c )
    echo "connexion impossible :
$php_errormsg<br>" ;
else
{
    echo "Ça marche !<br>" ;
    $c = mysql_close() ;
}
```



## Modification de l'entête HTTP (RFC 2068)

- PHP génère une entête classique, à laquelle on peut ajouter des lignes
  - `header("Content-Type: application/postscript") ;`
  - `header("Cache-control: no-cache, must-revalidate") ;`
  - `header("Pragma: no-cache") ;`
- On peut également utiliser un tag `<meta...>` dans le code HTML
  - `echo "<meta http-equiv=\"Refresh\" content=\"10[;url=http://www.php.net]\">";`
- Attention : l'envoi d'une entête HTTP doit se faire avant toutes les autres sorties





Formation à PHP version 2.1 - Copyright © 1999 - 2003 Pascal AUBRY - FSC - Université de Rennes 1



## Sorties non HTML

- La sortie d'un programme PHP peut être n'importe quoi
- L'entête HTTP indique au client le format des données renvoyées par défaut text/html

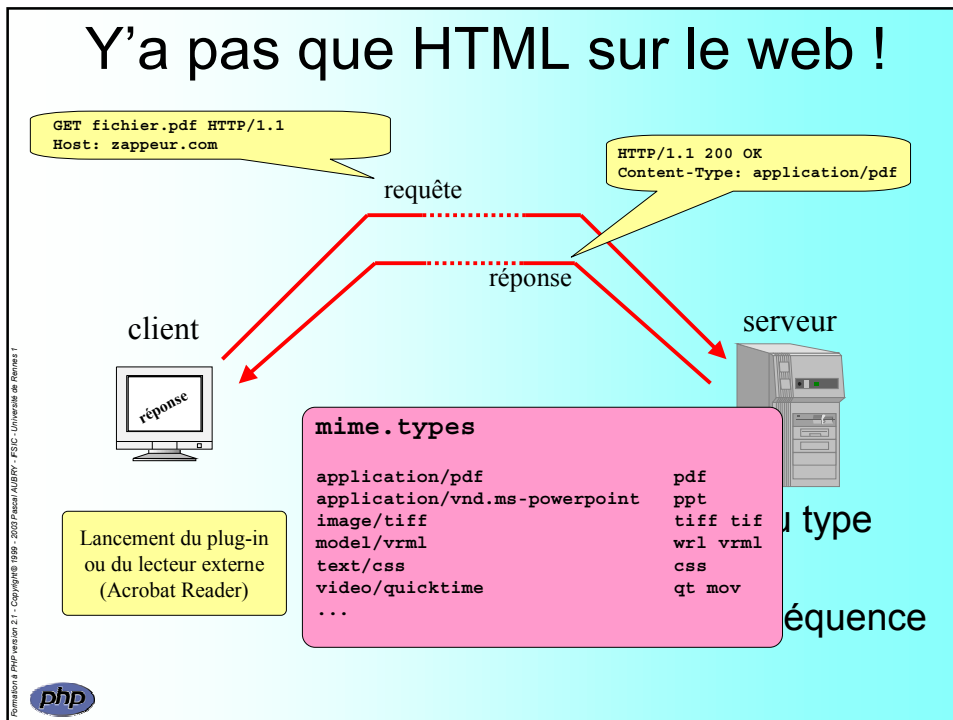
- Exemple (pour renvoyer du PostScript)

```
<? header("Content-Type: application/postscript") ;  
    readfile("/tmp/fichier.ps") ; exit ; ?>
```

Formation à PHP version 2.1 - Copyright © 1999 - 2003 Pascal AUBRY - FSC - Université de Rennes 1



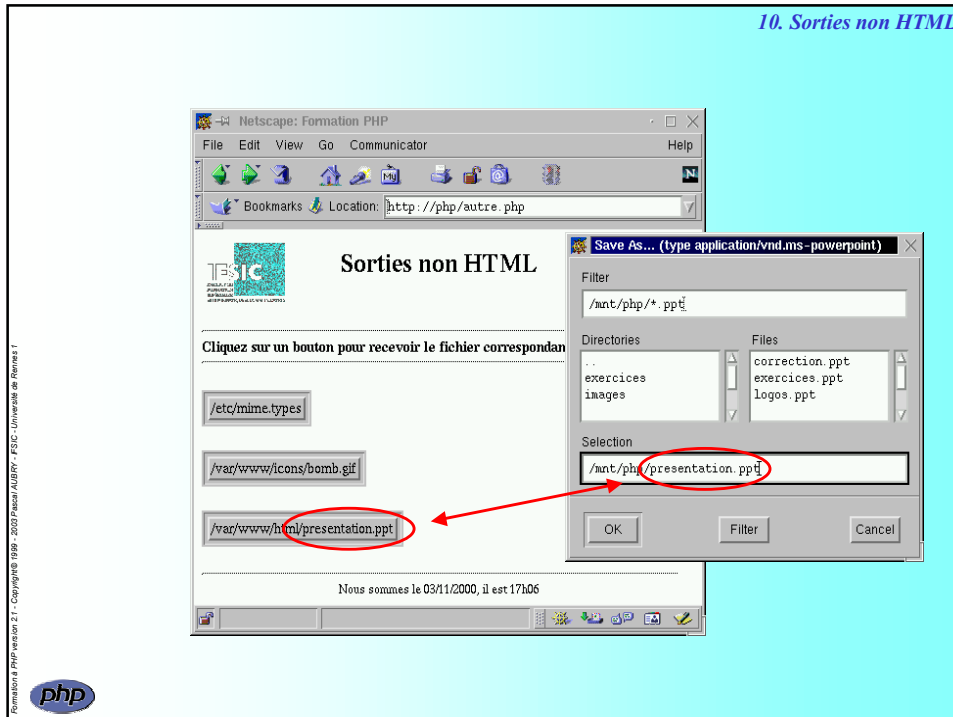
# Y'a pas que HTML sur le web !



## Sorties non HTML (exemple)

```
<?
strtok($SERVER["REMOTE_ADDR"], ".") ;
strtok(".");
strtok(".");
$last_ip_byte = strtok(".");
header("Content-Type: application/x-ns-proxy-
autoconfig");
echo "function FindProxyForURL(url,host) { return " ;
echo (
( $last_ip_byte % 2 )
? "\"www-cache1:3128\" "
: "\"www-cache2:3128\" " ) ;
echo "}" ;
?>
```

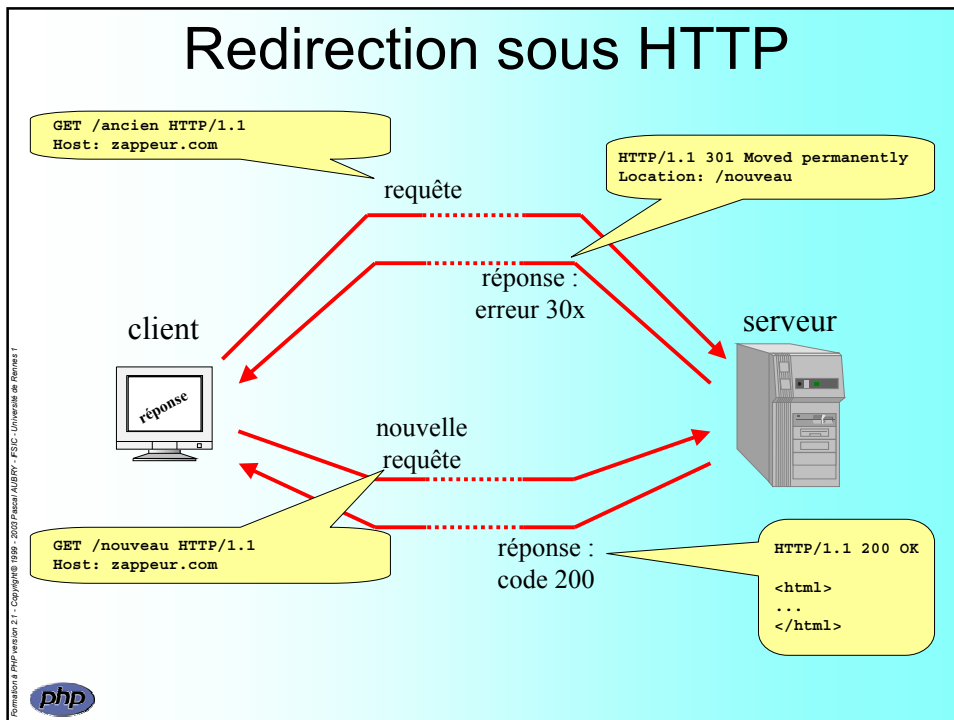




## Redirection

- Il est parfois utile de rediriger une requête vers une autre URL
  - en cas d'erreur
  - en cas d'une authentification réussie
- La redirection peut servir à forcer la transmission des cookies
- Exemples
  - `header("Location: /interdit.html") ;`
  - `header("Location: ".$SERVER["PHP_SELF"].'/autre.php') ;`

# Redirection sous HTTP



# Forcer le protocole HTTPS (redirection immédiate)

```
If ( $_SERVER['HTTPS'] != "on" )  
{  
    header("Location: https://" .  
        $_SERVER["SERVER_NAME"]  
        . ":443"  
        $_SERVER["REQUEST_URI"]);  
    exit ;  
}
```

*// à partir d'ici, on sait que l'on est sous HTTPS*

php

Formation à PHP version 2.1 - Copyright © 1999 - 2002 Pascal AUBRY - F3SC - Université de Rennes 1

## Forcer le protocole HTTPS (redirection différée)

```
if ( $_SERVER['HTTPS'] != "on")
{
    $new_url = "https://"
        . $_SERVER["SERVER_NAME"]
        . ":443";
    header("Refresh: 2;url=$new_url");
    echo "<h2>zappeur.com utilise maintenant
        le protocole HTTPS</h2>
        Vous allez être redirigé vers l'URL
        suivante dans quelques instants :
        <a href=\"$new_url\">$new_url</a>." ;
    exit ;
}
```



## Les cookies

- Ce sont des lignes particulières de l'entête HTTP qui permettent de véhiculer des informations entre client et serveur
- Caractéristiques :
  - nom (UTILISATEUR)
  - valeur (Pascal.Aubry)
  - date de validité (Sunday, 04-Nov-01 23:12:40 GMT)
  - domaine/serveur (www.zappeur.com)
  - chemin (/public)



# Les cookies

GET /rep/fichier.html HTTP/1.1  
Host: zappeur.com

HTTP/1.1 200 OK  
Set-Cookie: h\_cookie=00; \  
path=/rep/; \  
expires=Friday, 09-Nov-01 05:23:12 GMT

client

réponse

Alerte de sécurité

Pour avoir une expérience de navigation plus personnalisée, autorisez-vous ce site Web à enregistrer temporairement un petit fichier (appelé cookie) sur votre ordinateur ? Le fichier ne peut être utilisé que par le site Web.

Ne plus afficher cet avertissement

Qui Non Plus d'infos

Informations de cookie

Nom	h_cookie
Domaine	ifisc.univ-rennes1.fr
Chemin d'accès	/
Date d'expiration	ven., nov. 09 2001 05:23:12 Sécursisé <input type="checkbox"/>
Données	00

serveur

GET /rep/fichier2.htm  
Cookie: h\_cookie=00

php

Formation à PHP version 2.1 - Copyright© 1999 - 2002 Pascal AUBRY - FISC - Université de Rennes 1

# Les cookies (limitations)

- La taille
  - 4Ko par cookie
- Le nombre
  - 300 cookies par navigateur
  - 20 cookies par serveur (pour un même client)
- Les clients peuvent « refuser » des cookies

php

Formation à PHP version 2.1 - Copyright© 1999 - 2002 Pascal AUBRY - FISC - Université de Rennes 1

## Cookies (utilisation)

- Ils sont accessibles à travers une super-variables
  - \$\_COOKIE (ou \$\_REQUEST)
- Initialisation d'un cookie

```
SetCookie("my_cookie",  
         "valeur du cookie",  
         time()+3600,           // validité  
         "/magazin/",         // chemin  
         ".ifsic.univ-rennes1.fr", // domaine  
         1) ;                  // SSL
```
- Effacement d'un cookie
  - utiliser `SetCookie` avec une valeur vide
  - redonner les même paramètres



## Cookies (avertissement)

- Le cookie est transmis par le serveur au client en même temps que la page HTML
- Conséquence : le serveur ne voit le cookie initialisé (ou avec une nouvelle valeur) que lors de la requête HTTP suivante
- On utilise souvent une redirection



# Cookies (exemple)

- On veut associer un identifiant (unique) à un visiteur (par exemple pour connaître son parcours sur le site)

```
if ( empty($user_cookie) )
{
    $user_cookie = uniqid(rand()) ;
    SetCookie("user_cookie",
            $user_cookie,
            time()+3600,"/",
            ".univ-rennes1.fr") ;
    header("Location: /") ;
}
...
```



Formation à PHP version 2.1 - Copyright© 1999 - 2003 Pascal AUBRY - IFSIC - Université de Rennes 1

11. Cookies

The screenshot shows a Netscape browser window displaying a page titled "Cookies". The page content includes the text "Le nombre de passage sur cette page est 1." and "Liste des cookies :". A dialog box titled "Netscape: Question" is overlaid on the page, asking for permission to set a cookie. The dialog box text reads: "The server php.ifsic.univ-rennes1.fr wishes to set a cookie that will be sent to any server in the domain .ifsic.univ-rennes1.fr. The name and value of the cookie are: nb\_cookie=1. This cookie will persist until Fri Nov 3 19:04:36 2000. Do you wish to allow the cookie to be set?". A second dialog box is also visible, showing the same message but with the value "nb\_cookie=2". The browser's address bar shows "http://php.ifsic.univ-rennes1.fr/cookie.php".



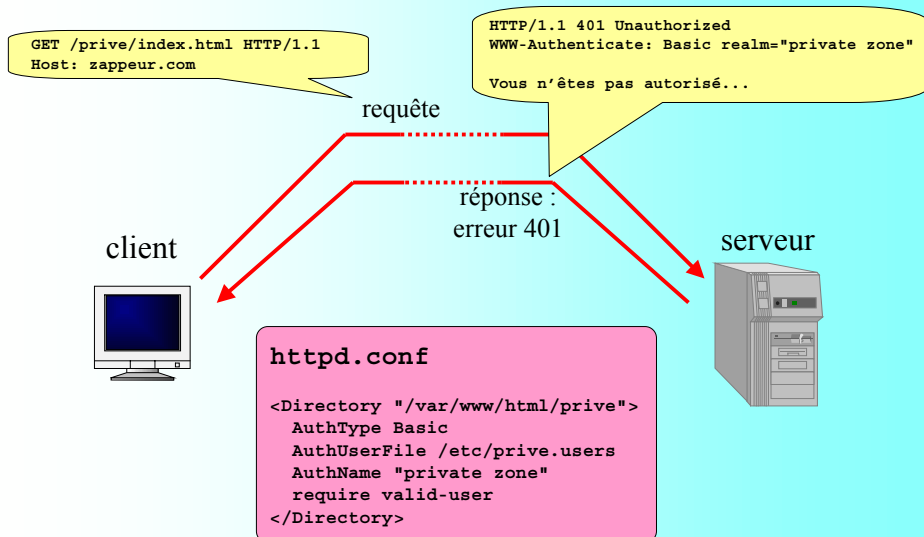
Formation à PHP version 2.1 - Copyright© 1999 - 2003 Pascal AUBRY - IFSIC - Université de Rennes 1

# L'authentification sur le web

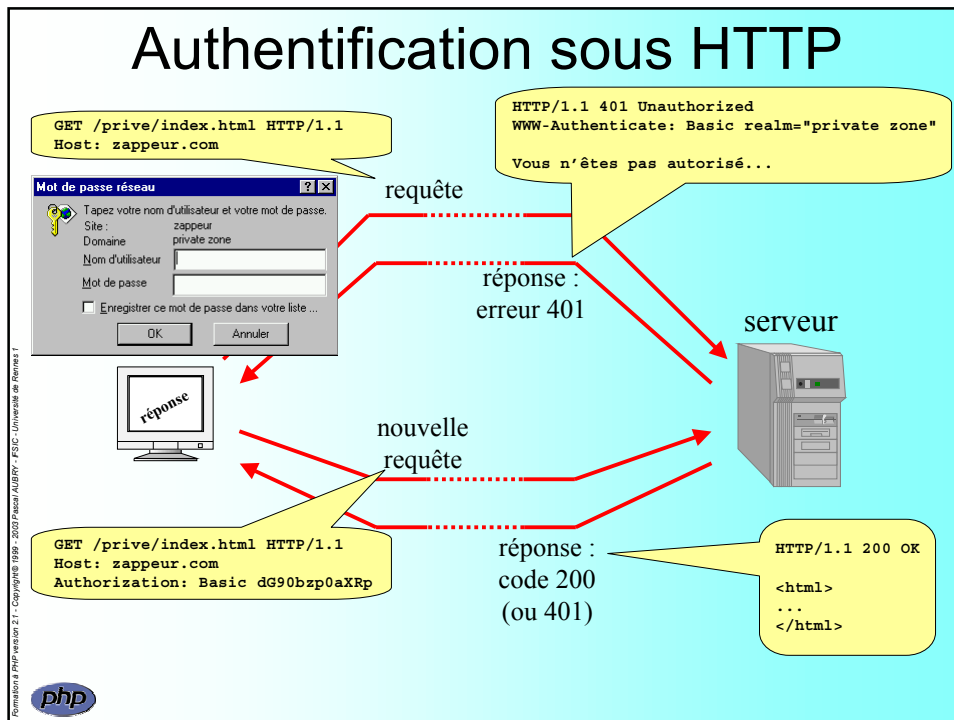
- Transport des informations d'authentification
  - protocole HTTP (royaume)
  - cookies
  - SSL (Secure Socket Layout)
- Vérification
  - par le démon HTTP ou par l'application
- Référence
  - fichiers, base de données, codage « en dur »



# Authentification sous HTTP



# Authentification sous HTTP



## Vérification de l'authentification

- Par le démon httpd
  - authentification HTTP simple
    - sur un fichier de mots de passe (htpasswd)
    - sur une base de données (mod\_auth\_mysql)
    - sur un annuaire (auth\_ldap)
  - authentification SSL
    - sur des certificats (X509)
- Par applicatif
  - sur n'importe quoi !



# Protéger le site avec HTTP

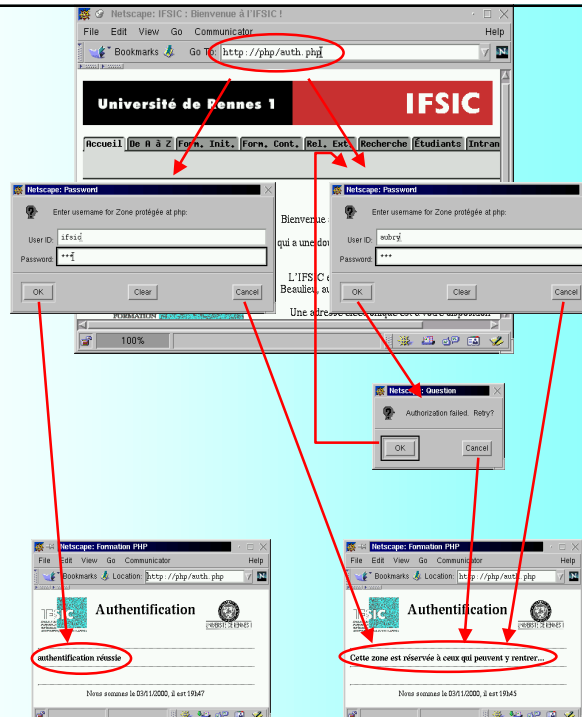
```
if ( ($_SERVER["PHP_AUTH_USER"] != "titi")
    || ($_SERVER["PHP_AUTH_PW"] != "toto") )
{
    header('WWW-Authenticate: Basic realm="zappeur"');
    header("HTTP/1.0 401 Unauthorized");
    echo "Vous n'êtes pas autorisé...\n" ;
    exit ;
}

// à partir d'ici, l'utilisateur est authentifié
```



Formation PHP version 2.1 - Copyright 1999 - 2002 Pascal AUBRY - IFSIC - Université de Rennes 1

## 12. Authentification



Formation PHP version 2.1 - Copyright 1999 - 2002 Pascal AUBRY - IFSIC - Université de Rennes 1

# Protéger le site avec HTTPS

- Par le démon httpd
  - vérification du certificat client suivant configuration
- Par applicatif
  - protocole
    - `SSL_PROTOCOL`, `SSL_CIPHER`, `SSL_CIPHER_ALGKEYSIZE`,
  - vérification du certificat client
    - `SSL_CLIENT_CERT`, `SSL_CLIENT_CERT_CHAINx`,  
`SSL_CLIENT_VERIFY`
  - informations sur les certificats client et serveur
    - `SSL_XXX_I_DN_C`, `SSL_XXX_I_DN_CN`, `SSL_XXX_I_DN_Email`,  
`SSL_XXX_I_DN_L`, `SSL_XXX_I_DN_O`, `SSL_XXX_I_DN_OU`,  
`SSL_XXX_I_DN_ST`, `SSL_XXX_M_SERIAL`, `SSL_XXX_M_VERSION`,  
`SSL_XXX_S_DN`, `SSL_XXX_S_DN_C`, `SSL_XXX_S_DN_CN`,  
`SSL_XXX_S_DN_Email`,



# PHP et les bases de données

- Un support très large
  - Adabas D, dBase, filePro, Hyperwave, Informix, InterBase, mSQL, MS SQL Server, MySQL, Oracle 7, Oracle 8, SyBase, PostgreSQL, Solid, ODBC
- Utiliser la bibliothèque de son SGBD
- En l'absence de bibliothèque, utiliser ODBC
  - ou développer sa propre bibliothèque ;-)
- Il existe des interface génériques
  - phpDB



## Connexions persistantes (aux bases de données)

- La connexion à une base de données est souvent l'opération la plus coûteuse d'une requête
- Identification d'une connexion :  
(*host, database, user, password*)
- Au lieu d'ouvrir et fermer une connexion à une base de données à chaque requête, on peut garder les connexions ouvertes et les mémoriser pour pouvoir les réutiliser par la suite



## Accès à un SGBD (schéma)

- Connexion ([p]=persistante)  

```
$conn = xxx_[p]connect( serveur,  
                        port,  
                        base,  
                        utilisateur,  
                        mot_de_passe) ;
```
- Interrogation/mise à jour  

```
$query = xxx_query(requête) ;  
$res = xxx_fetch($query) ;
```
- Fermeture  

```
xxx_close($conn) ;
```



# Accès à MySQL

- Connexion

```
$conn = mysql_[p]connect(serveur,  
                        utilisateur,  
                        mot_de_passe) ;  
$bdd = mysql_select_db(base,$conn) ;
```

- Interrogation/mise à jour

```
$query = mysql_query(requête) ;  
$res = mysql_fetch_[row|array|assoc]($query) ;
```

- Fermeture

```
mysql_close($conn) ;
```



# Accès à MySQL (exemple)

```
if ( $link = @mysql_connect($host,$user,$pw) )  
{  
    if ( @mysql_select_db($db,$link) )  
    {  
        if ( $query = @mysql_query($req) )  
            . . .  
        else  
            echo "Mauvaise requête ($php_errormsg).\" ;  
    }  
    else  
        echo "BDD introuvable (".mysql_error().").\" ;  
    mysql_close($link) ;  
}  
else  
    echo "Connexion impossible ($php_errormsg).\" ;
```



## Accès générique aux bases de données

```
// appel de la bibliothèque phpDB
Require("DB.php") ;
// connexion à la base de données
$conn = DB::Connect("mysql://user:passwd@host/db" ) ;
// exécution d'une requête SQL
$res = $conn->Query(" . . . " ) ;
// parcours du résultat
while ( $row = $res->FetchRow(DB_FETCHMODE_ASSOC) )
    { . . . }
// fermeture de la connexion
$conn->Close() ;
```



## Accès générique aux bases de données (exemple)

```
Require("DB.php") ;
$conn = DB::Connect("pgsql://user:passwd@host/db") ;
$res = $conn->Query("SELECT nom,prenom FROM personne") ;
while ( $row = $res->FetchRow(DB_FETCHMODE_ASSOC) )
    { echo $row["nom"]." ".$row["prenom"].'<br>' ; }
$conn->Close() ;
```



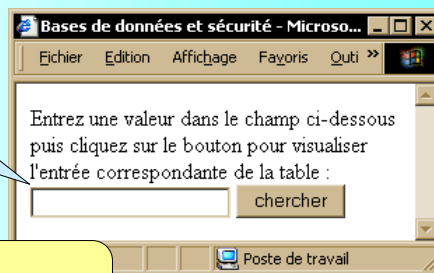
# Récupération des erreurs avec phpDB

```
Require("DB.php") ;
$conn = DB::Connect("pssql://user:passwd@host/db") ;
If ( DB::isError($conn) )
    { echo "Connexion impossible :"  
      .DB::ErrorMessage($conn); exit(); }
$res = $conn->Query("SELECT nom,prenom FROM personne") ;
If ( DB::isError($res) )
    { echo "Connexion impossible :"  
      .DB::ErrorMessage($res); exit(); }
while ( $row = $res->FetchRow(DB_FETCHMODE_ASSOC) )
    { echo $row["nom"]." ".$row["prenom"].'<br>'; }
$conn->Close() ;
```



# Sécurité et bases de données

```
<form ...>
...
<input type="text"
name="cle">
<input type="submit"
name="go"
value="chercher">
</form>
```



```
$req_sql =
"SELECT DesChamps
FROM UneTable
WHERE cle = '$cle' ";
```

J'essaierai bien la clé suivante :

```
4' ;
SELECT * FROM ... ;
DROP DATABASE ... ;
SELECT '1
```



# Sécurité et bases de données

- Tout ce qui provient de l'utilisateur doit être considéré comme suspect :
  - paramètres CGI, fichiers téléchargés, ...

```
$req_sql = "  
SELECT DesChamps FROM UneTable  
WHERE cle = '". addslashes($cle) ."' ;
```



## 13. Exécution de requêtes SQL

La requête a été exécutée correctement.

Serveur: localhost Base: pcd  
Utilisateur: pcd Mot de passe: \*\*\*

Requête:

Nombre de lignes affectées : 2

nombre	texte	reel	date
1	texte d'exemple	3	2000-09-23 15:46:02
-43	deuxieme texte d'exemple	5.23	2000-10-30 16:23:22

Nous sommes le 06/11/2000, il est 17h48

Mauvaise requête (Unknown column 'inexistent' in 'field list').

Serveur: localhost Base: pcd  
Utilisateur: pcd Mot de passe: \*\*\*

Requête:

Nous sommes le 06/11/2000, il est 17h49

## Remarques

- Le couple utilisateur/mot de passe peut être passé par HTTP
- Les informations de connexion peuvent être passées en permanence à l'aide de cookies
  - serveur, port, base, utilisateur, mot de passe
- Penser à échapper les variables passées dans les requêtes
  - `addslashes()`
- Pour MySQL
  - pas de problème de transactions ;-)
  - en revanche problèmes d'accès concurrent



## Aller plus loin...

Gestion de sessions  
Téléchargement de fichiers  
Gestion des connexions HTTP  
Exécution de programmes externes  
Courrier électronique  
Création d'images





# Transmettre des variables d'état

- Objectif : propager des variables entre les requêtes
  - identité du visiteur
  - informations de connexion
  - ...
- Le problème
  - HTTP est un protocole sans état
- Les solutions
  - paramètres CGI cachés
  - cookies
  - sessions



## Solution 1 : paramètres CGI cachés

- Dans chaque formulaire et chaque lien hypertexte, on ajoute un (des) paramètre(s) caché(s) :

```
<form ...>
  <input type="hidden" name="ident" value="durand">
  <input type="hidden" name="pass" value="xx98yy">
  ...
</form>
```

```
<a href="action.php?ident=durand&pass=xx98yy">texte</a>
```

- Problèmes :
  - Les liens hypertextes et les grosses variables d'état
  - on transmet toutes les variables d'état à chaque requête
  - ça marche, mais ça va bien un moment... ;-)



## Solution 2 : cookies

- À chaque changement d'une variable d'état, on envoie un cookie au client qui pourra nous le renvoyer lors de la requête suivante
- Intérêts :
  - on ne transmet les variables que lorsqu'elles changent
  - on ne fait pas de ré-écriture des formulaires et des liens
- Problèmes :
  - nombre et longueur des cookies
  - il faut encore le faire « à la main »
  - certains ne supportent pas les cookies...



## Solution 3 : les sessions

- Affectation d'un ID unique
  - pour chaque visiteur non connu (sans ID)
  - de forme aléatoire
- Liaison (identifiant - données sur le serveur)
  - stockage mémoire, disque, base de données, ...
  - format propriétaire, XML (WDDX), ...
- Transmission de l'identifiant
  - par cookie, ré-écriture (automatique)



## Transmission des IDs de session

- Ré-écriture automatique :

```
- <form ...>
    <input type="hidden"
          name="SESS_ID"
          value="5kj81112yhs3">
    ...
</form>
- <a href="...?...&SESS_ID=5kj81112yhs3">
- 
- <frame src="...?...&SESS_ID=5kj81112yhs3">
- ...
```



## Sessions dans PHP (1)

- Un support transparent
  - compilation avec `--enable-trans-sid`
  - transmission via les paramètres CGI (≠ cookies)
  - démarrage automatique ou manuel
    - `session_register()`, `session_start()`
- Les variables de session se trouvent dans `$_SESSION`.
- L'identifiant de session est accessible par `session_name()`



## Sessions dans PHP (2)

- Génération aléatoire des identifiants
  - spécification possible de sources d'entropie
- Stockage des informations de session
  - sur fichier (par défaut) ou base de données
- Format de stockage
  - format interne PHP (par défaut) ou WDDX
- Garbage collector automatique

## Tracer la connexion des utilisateurs

```
// on active le mécanisme de session
session_start() ;
// on déclare les variables de session
session_register("nb_req") ;
session_register("debut") ;

$_SESSION["nb_req"]++ ;
if ( empty($_SESSION["debut"]) )
    $_SESSION["debut"] = time() ;
// on calcule le temps de connexion
$_SESSION["temps"] = time()-$_SESSION["debut"] ;
```

Formation PHP version 2.1 - Copyright© 1999 - 2003 Pascal AUBRY - F3SC - Université de Rennes 1

php

14. Traçage d'un utilisateur à l'aide de sessions

## Télécharger des fichiers sur le serveur (uploading)

- Il est parfois intéressant de transmettre les données sous forme de fichiers
  - données de taille importante
  - données déjà stockées sous forme de fichier sur le client
- Un encodage particulier du formulaire
- Une prise en charge par le serveur

## Formulaire de téléchargement

```
<form enctype="multipart/form-data"
      action="upload.php"
      method="post">
  <input type="hidden"
        name="MAX_FILE_SIZE"
        value="50000">
  Envoyer ce fichier :
  <input name="fichier" type="file">
  <br>
  <input type="submit"
        value="Envoyer">
</form>
```



## Accepter un téléchargement

- Nom du fichier (temporaire) sur le serveur
  - `$_FILES["fichier"]["tmp_name"]`
  - "none" si aucun fichier n'a été fourni
- Nom du fichier sur le client
  - `$_FILES["fichier"]["name"]`
- Taille du fichier
  - `$_FILES["fichier"]["size"]`
- Type du fichier
  - `$_FILES["fichier"]["type"]`
  - pas toujours disponible



# Accepter un téléchargement (exemple)

```
// test pour vérifier que l'utilisateur
// a bien transmis un fichier
if ( $_FILES["fichier"]["tmp_name"] == "none" )
    return ;

// sauvegarde sous le nom original
// (nom du fichier sur le client)
if ( !move_uploaded_file(
    $_FILES["fichier"]["tmp_name"],
    "repert/" . $_FILES["fichier"]["name"]
) )

    return ;

// taille du fichier : $_FILES["fichier"]["size"]
...

```



## 15. Téléchargement de fichiers

total 16  
drwxrwxrvt 2 root root 4096 Nov 3 21:11  
drwxrwxrvt 7 root root 4096 Nov 3 21:11  
-rw-rw-r-- 1 apache apache 5721 Nov 3 21:11 logo\_ifsic.gif

Nous sommes le 03/11/2000, il est 21h11



## Une connexion dure...

- ... au moins un certain temps !
- Largement le temps pour qu'elle soit interrompue
  - par le client (arrêt du chargement en cours)
  - par le serveur (défaillance quelconque)
  - par le média (panne sur le réseau)
- Il faut s'y préparer
  - transactions avec des bases de données
  - ...



## Connexions HTTP (gestion)

- La durée d'exécution d'un programme PHP est limitée
- Une connexion est dans un des états suivants :
  - 0 - ACTIVE (tout va bien)
  - 1 - ABORTED (l'utilisateur a stoppé le chargement)
  - 2 - TIMEOUT (durée d'exécution dépassée)
- Pour connaître l'état d'une connexion :
  - `connection_aborted()`, `connection_timeout()`





## Connexions HTTP (contrôle)

- Continuer l'exécution même lorsque la connexion est interrompue :
  - `ignore_user_abort()`
- Changer le temps maximal d'exécution :
  - `set_time_limit()`



## Exécution de programmes externes

- Réserve aux actions ne pouvant être effectuées à l'aide des bibliothèques
  - moins rapide
  - moins portable
- **Attention : trous de sécurité !**
  - `escapeshellarg()`, `escapeshellcmd()`
- Penser à rediriger les flux (`2>&1`)

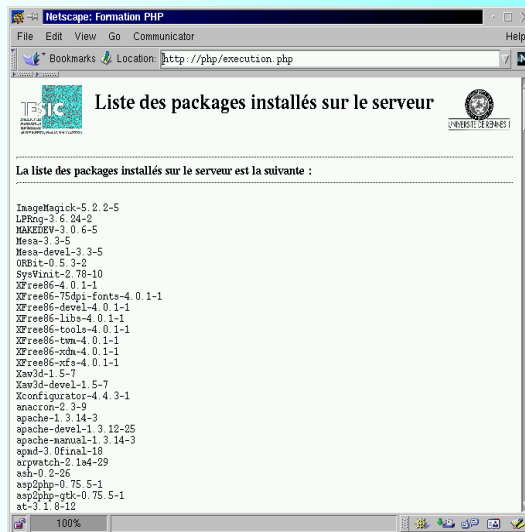


# Exécution de programmes externes (fonctions)

- **system** (\$commande)
  - sortie standard
  - code de retour : celui de la dernière commande
- **passthru** (\$commande)
  - sortie standard affichée directement (*raw output*)
  - code de retour : celui de la dernière commande
- **exec** (\$commande [, \$sortie [, \$resultat]])
  - sortie : aucune (formatage possible de \$sortie)
  - code de retour : \$resultat



## 16. Exécution de programmes externes



# Envoi d'un courrier électronique

- Grâce à la fonction `mail()`

- Exemple :

```
mail("sysadmin",  
    "*** alerte ***",  
    "<font color=ff0000>Aïe aïe aïe !</font>",  
    "Content-Type: text/html") ;
```

- Attention : c'est l'utilisateur qui exécute le démon `httpd` qui envoie le mail !



## 17. Envoi de courriers électroniques

The screenshot shows a Netscape browser window titled "Formation PHP". The address bar shows "http://php/mail.php". The page content includes a header with logos for "F3SC" and "UNIVERSITÉ DE RENNES 1". Below the header, there is a form with the following elements:

- A heading: "Envoi d'un courrier électronique"
- A sub-heading: "Remplissez les champs puis cliquez sur le bouton envoyer."
- A text input field labeled "Destinataire:".
- A text input field labeled "Sujet:" with the placeholder text "[pas de sujet]".
- A text area labeled "Message:" with the placeholder text "Entrez votre message ici."
- A button labeled "Envoyer".
- A footer line: "Nous sommes le 03/11/2000, il est 15h34".



## Création d'image

- À l'aide de la librairie GD
  - <http://www.boutell.com/gd>
- Grâce à Unisys :
  - GD 1.6 ou avant : format GIF
  - aujourd'hui (1.8.3) : format PNG



## Création d'images (exemple)

```
// ouverture de l'image
if (!$im = ImageCreateFromPNG("logos/img")) )
    return ;
// allocation de la couleur d'écriture
$fg = ImageColorAllocate ($im, 255, 255, 255);
// affichage d'un texte
ImageTTFText ($im, 16, 0, 10, 20, $fg, "comicbd.ttf", "texte") ;
// envoi sur la sortie standard
header("Content-Type: image/jpeg") ;
ImageJpeg ($im) ;
ImageDestroy ($im) ;
```



## 18. Création/modification d'images

The screenshot shows a Netscape browser window titled "Netscape: Formation PHP". The address bar shows the URL "http://php/param\_image.php". The page content includes the logos for "TESIC" and "UNIVERSITE DE KENNES". Below the logos, the text reads "Entrez les paramètres puis cliquez sur créer." followed by several input fields: "Taille:" with "H:" (40) and "L:" (125); "Couleur de fond:" with "R:" (255), "G:" (128), and "B:" (128); "Couleur du texte:" with "R:" (0), "G:" (0), and "B:" (0); "Texte:" with a text box containing "Formation PHP"; and "Position du texte:" with "X:" (10) and "Y:" (10). A "Créer l'image" button is located to the right of the position fields. Below the form, a red rectangular image with the text "Formation PHP" is displayed. At the bottom of the page, it says "Nous sommes le 05/11/2000, il est 18h36". The browser's status bar shows "100%".

Formation PHP version 2.1 - Copyright© 1999 - 2002 Pascal AUBRY - FISC - Université de Rennes 1

## 19. Mise en œuvre d'un compteur graphique avec MySQL

The screenshot shows a Netscape browser window titled "Netscape: Formation PHP". The address bar shows the URL "http://php/test\_compteur.php". The page content includes the logos for "TESIC" and "UNIVERSITE DE KENNES". Below the logos, there are two horizontal dotted lines. The text "Nombre de visites pour cette page : 00306" is displayed in red. Below this, there are two more horizontal dotted lines. At the bottom of the page, it says "Nous sommes le 05/11/2000, il est 18h16". The browser's status bar shows "100%".

Formation PHP version 2.1 - Copyright© 1999 - 2002 Pascal AUBRY - FISC - Université de Rennes 1

# Le chiffrement

SSL + HTTP = HTTPS



Formation à PHP version 2.1 - Copyright© 1999 - 2003 Pascal AUBRY - IFSIC - Université de Rennes 1

# Le chiffrement

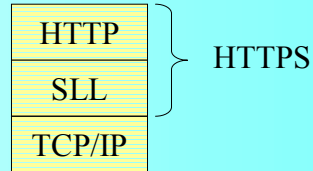
- Être sûr que personne ne puisse lire des données envoyées sur un réseau
- chiffrement (codage) à l'aide d'un algorithme



Formation à PHP version 2.1 - Copyright© 1999 - 2003 Pascal AUBRY - IFSIC - Université de Rennes 1

# Le cryptage de HTTP (HTTPS)

- SSL : Secured Socket Layout
- HTTPS : HTTP sur SSL
- Objectifs de HTTPS :
  - crypter les communications
  - identifier le serveur
    - le serveur auquel je suis connecté est-il le bon ?
  - Identifier le client
    - la personne qui se connecte au serveur est-elle la bonne ?
- Certificats (X509)






# Chiffrement symétrique

- Une seule clé pour chiffrer/déchiffrer
  - Chiffrement :  
 $f(\text{message}, \text{clé}) = \text{message\_chiffré}$
  - Déchiffrement :  
 $f(\text{message\_chiffré}, \text{clé}) = \text{message}$
  - Symétrie :  
 $\text{message} = f(f(\text{message}, \text{clé}), \text{clé})$
- Avantage : rapidité
- Inconvénient : comment se passer la clé ?

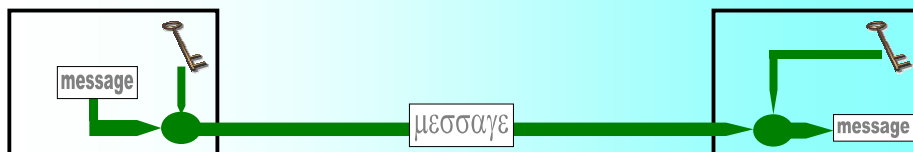


# Chiffrement symétrique

- Chiffrement :  
 $f(\text{message}, \text{clé}) = \text{message\_chiffré}$   

  - Déchiffrement :  
 $f(\text{message\_chiffré}, \text{clé}) = \text{message}$   

  - Avantages :
    - Une seule clé pour chiffrer/déchiffrer
-  Algorithme de chiffrement léger

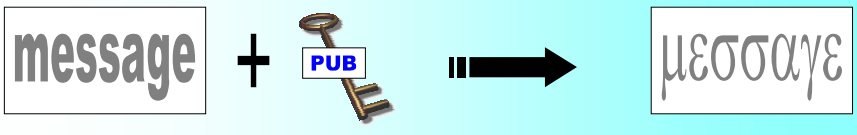
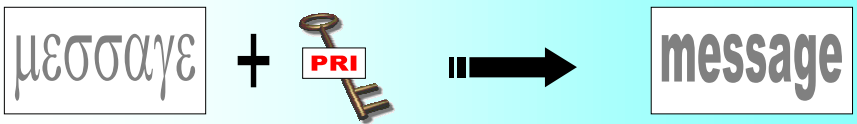
# Communication chiffrée symétrique

- Inconvénient :
  - Comment se passer la clé ?





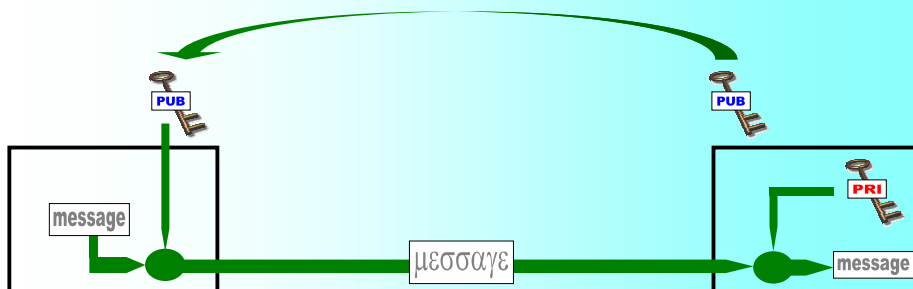
# Chiffrement asymétrique

- Chiffrement :  
 $f(\text{message}, \text{clé\_pub}) = \text{message\_chiffré}$   

- Déchiffrement :  
 $f(\text{message\_chiffré}, \text{clé\_pri}) = \text{message}$   


Formation à PHP version 2.1 - Copyright © 1999 - 2002 Pascal AUBRY - IJSC - Université de Rennes 1



# Communication chiffrée asymétrique



- Inconvénient :
  - L'algorithme de chiffrement/déchiffrement est lourd

Formation à PHP version 2.1 - Copyright © 1999 - 2002 Pascal AUBRY - IJSC - Université de Rennes 1



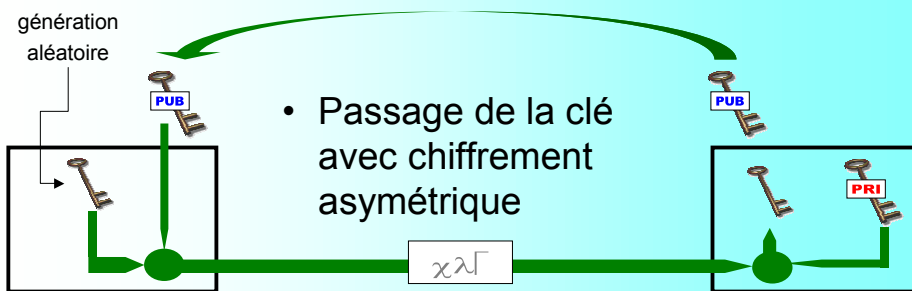
# La solution

- On utilise un chiffrement asymétrique pour se communiquer une clé symétrique
- On communique ensuite avec un chiffrement symétrique

Formation à PHP version 2.1 - Copyright © 1999 - 2002 Pascal AUBRY - FSC - Université de Rennes 1



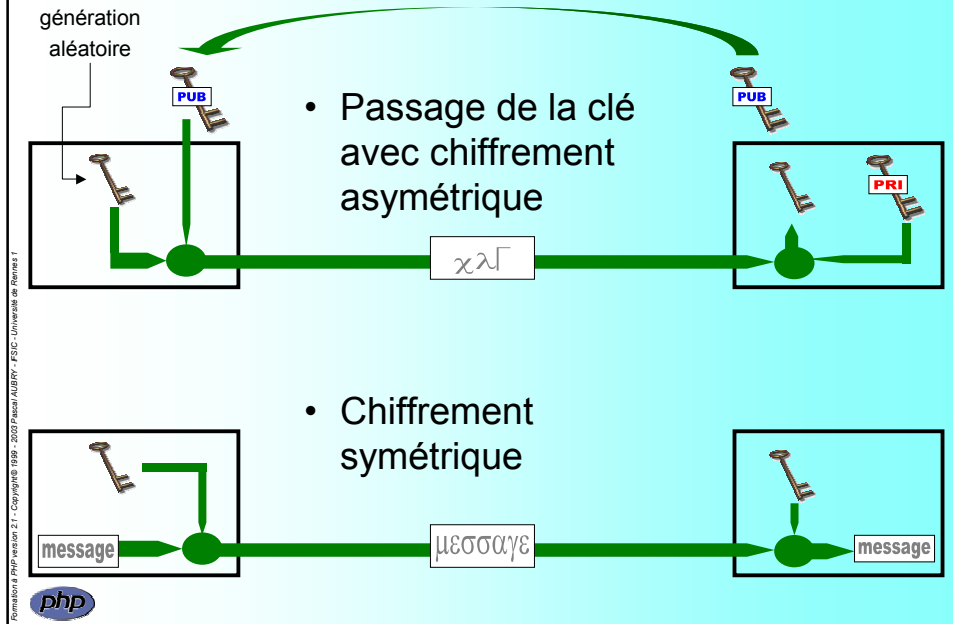
# La solution



Formation à PHP version 2.1 - Copyright © 1999 - 2002 Pascal AUBRY - FSC - Université de Rennes 1



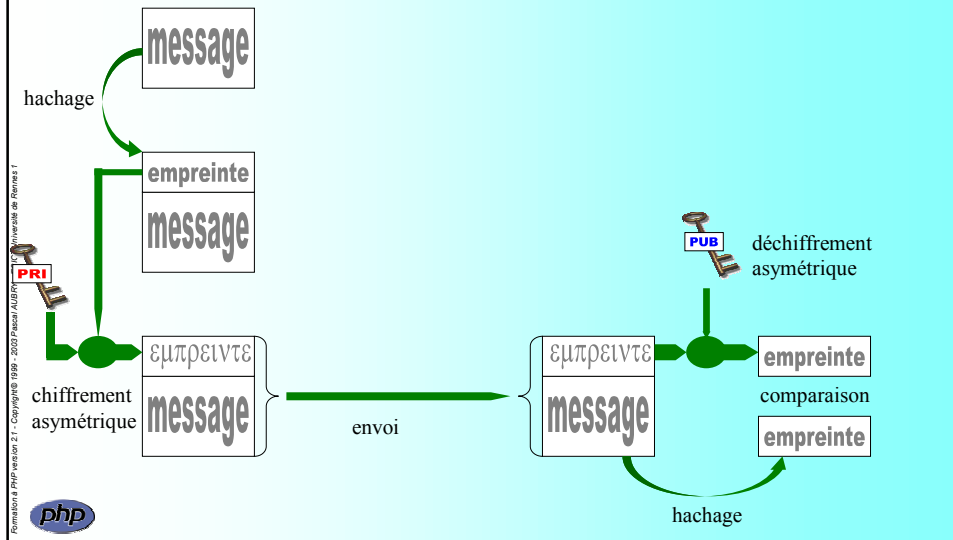
## La solution



## La signature

- Être sûr que la personne/machine avec laquelle on communique est bien la bonne
  - Être sûr de l'intégrité des données reçues
- Formation PHP version 2.1 - Copyright 1999 - 2003 Pascal AUBRY - FSC - Université de Rennes 1  
php

# La signature (fonctionnement)



Chiffrement + signature  
=  
confidentialité + intégrité

# Notes



# Formation à PHP (exercices)



Juin 2003  
Formation du Grand-Ouest  
Version 2.1

Pascal AUBRY  
IFSIC - Université de Rennes 1  
Pascal.Aubry@univ-rennes1.fr  
<http://perso.ifsic.univ-rennes1.fr/aubry>

L'utilisation de ce document dans un but de formation de groupe, y compris dans le cadre universitaire, est soumise à une autorisation explicite et préalable de son auteur.





# Table des matières

1. Configuration du serveur
2. Utilisation de fonctions
3. Écriture de fonctions
4. Inclusion de fichiers
5. Paramètres CGI
6. Création d'une classe
7. Héritage
8. Accès au système de fichiers
9. Rafraîchissement automatique
10. Sorties non HTML
11. Cookies
12. Authentification
13. Exécution de requêtes SQL
14. Traçage d'un utilisateur à l'aide de sessions
15. Téléchargement de fichiers
16. Exécution de programmes externes
17. Envoi de courriers électroniques
18. Création/modification d'images
19. Mise en œuvre d'un compteur graphique avec MySQL

Configuration de PHP (/etc/php.ini)



# 1. Configuration du serveur

**But :** Afficher la configuration du serveur.

**Intérêt :** Connaître les variables à disposition du programmeur.

**Référence :** `phpinfo()`.

**Fichier :** `info.php`.

**Indication :** Écrire le programme `info.php` qui doit afficher la configuration du serveur.

**Aller plus loin :** Modifier la couleur du fond de la fenêtre (`<body bgcolor=...>`).



## 2. Utilisation de fonctions

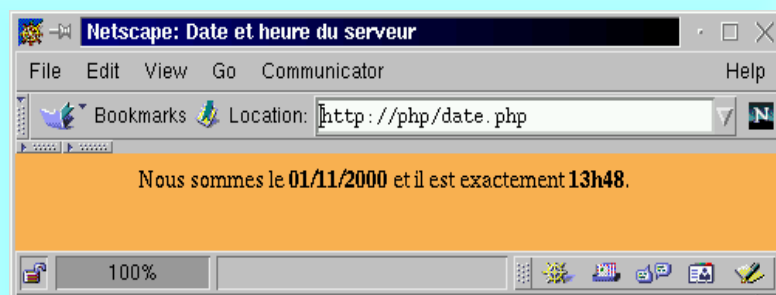
**But :** Afficher la date et l'heure du serveur.

**Intérêt :** Utiliser la valeur retournée par une fonction.

**Pointeurs :** `echo, date ()`.

**Fichier :** `date.php`.

**Indication :** Écrire le fichier `date.php`.



## 3. Écriture de fonctions

**But :** Afficher la date et l'heure du serveur.

**Intérêt :** Écrire et appeler une fonction.

**Pointeurs :** `function, date ()`.

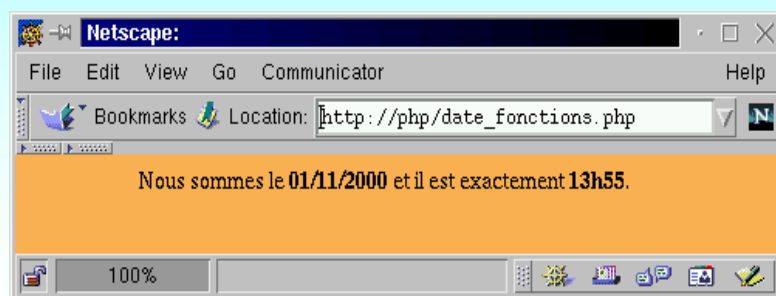
**Fichier :** `date_fonctions.php`.

**Indications :**

Créer deux fonctions `debut_html ()` et `fin_html ()` qui écrivent sur la sortie standard du programme le début et la fin de la sortie HTML.

La fonction `debut_html` doit avoir un paramètre (le titre de la page).

Le paramètre doit être facultatif.



## 4. Inclusion de fichiers

**But :** Afficher la date et l'heure du serveur.

**Intérêt :** Utiliser l'inclusion de fichier pour modulariser le code PHP.

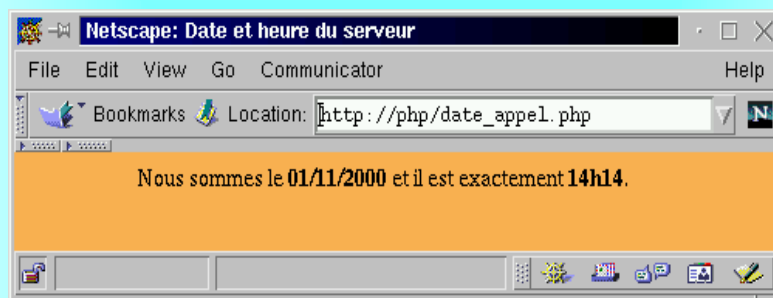
**Fichiers :**

sortie.inc, le fichier définissant les fonctions ;  
date\_appel.php, le fichier qui inclut sortie.inc.

**Pointeurs :** include(), require(), include\_once(), require\_once().

**Indications :**

Déplacer les deux fonctions debut\_html() et fin\_html() dans un fichier sortie.inc.  
Inclure sortie.inc dans date\_appel.php.



## 5. Paramètres CGI

**But :**

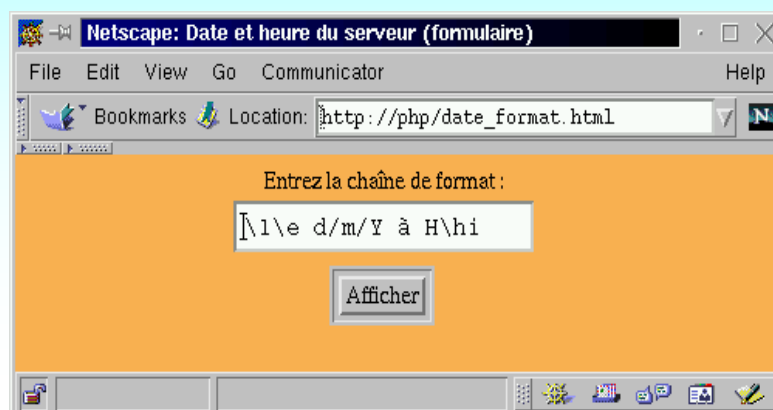
Demander une chaîne de format à l'utilisateur (date\_format.html) et afficher la date et l'heure du serveur selon ce format (date\_format.php).

**Intérêt :**

Écrire un formulaire simple (une boîte de dialogue et un bouton) ;  
Utiliser un paramètre CGI.

**Fichiers :**

date\_format.html, le fichier HTML affichant le formulaire.  
date\_format.php, le programme PHP qui prend les données du formulaire.





## 6. Création d'une classe

**But :** Transformer des fonctions en méthodes d'une classe.

**Intérêt :**

- Créer une classe ;
- Créer et utiliser une instance de la classe.

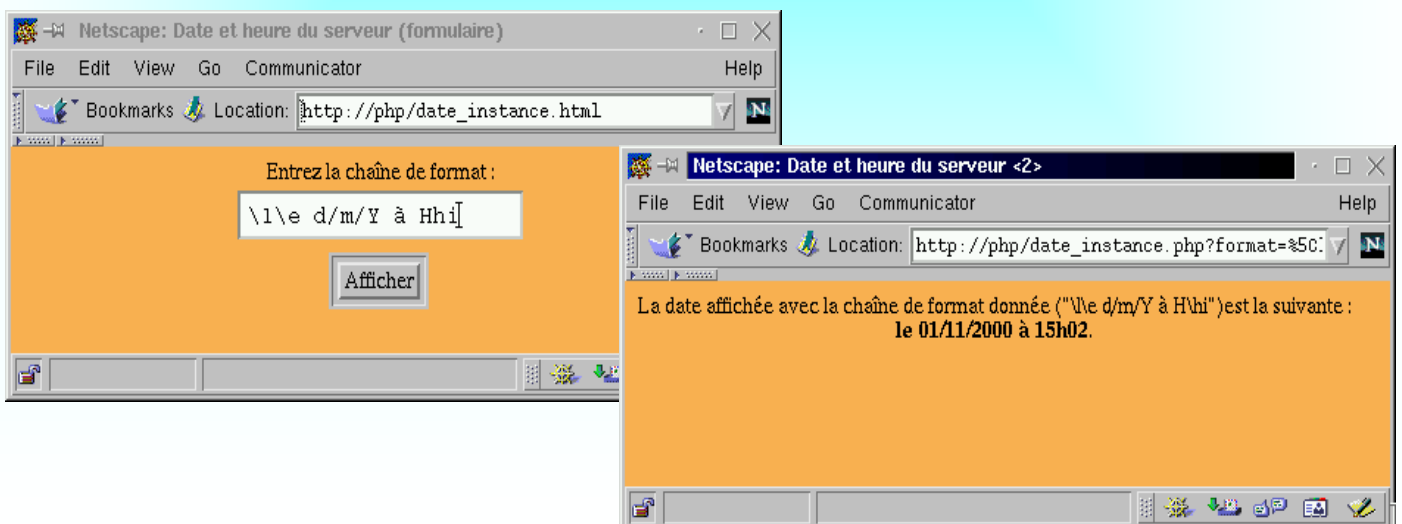
**Fichiers :**

- sortie\_classe.inc, qui définit la classe ;
- date\_instance.php, qui inclut sortie\_classe.inc et instancie la classe.

**Pointeurs :** class, var, new

**Indications :**

- Sauvegarder date\_format.html dans date\_instance.html et appeler date\_instance.php (au lieu de date\_format.php) ;
- Sauvegarder sortie.inc dans sortie\_classe.inc ;
- Créer dans sortie\_classe.inc une classe sortie qui doit avoir :
  - deux attributs (le titre et la couleur de la page) ;
  - un constructeur initialisant les attributs à l'aide de deux paramètres ;
  - une méthode debut\_html (sans paramètre, qui utilise les attributs) ;
  - une méthode fin\_html (également sans paramètre).
- Sauvegarder date\_format.php dans date\_instance.php ;
- Créer dans date\_instance.php une instance de la classe sortie, et utiliser ses méthodes.



## 7. Héritage

**But :** Faire hériter une classe d'une classe existante

**Intérêt :**

- Utiliser les propriétés d'une classe ;
- Étendre ces propriétés ;
- Utiliser une alternative.

**Fichiers :**

- sortie\_classe.inc (inchangé) ;
- date\_classe.php.

**Pointeurs :**

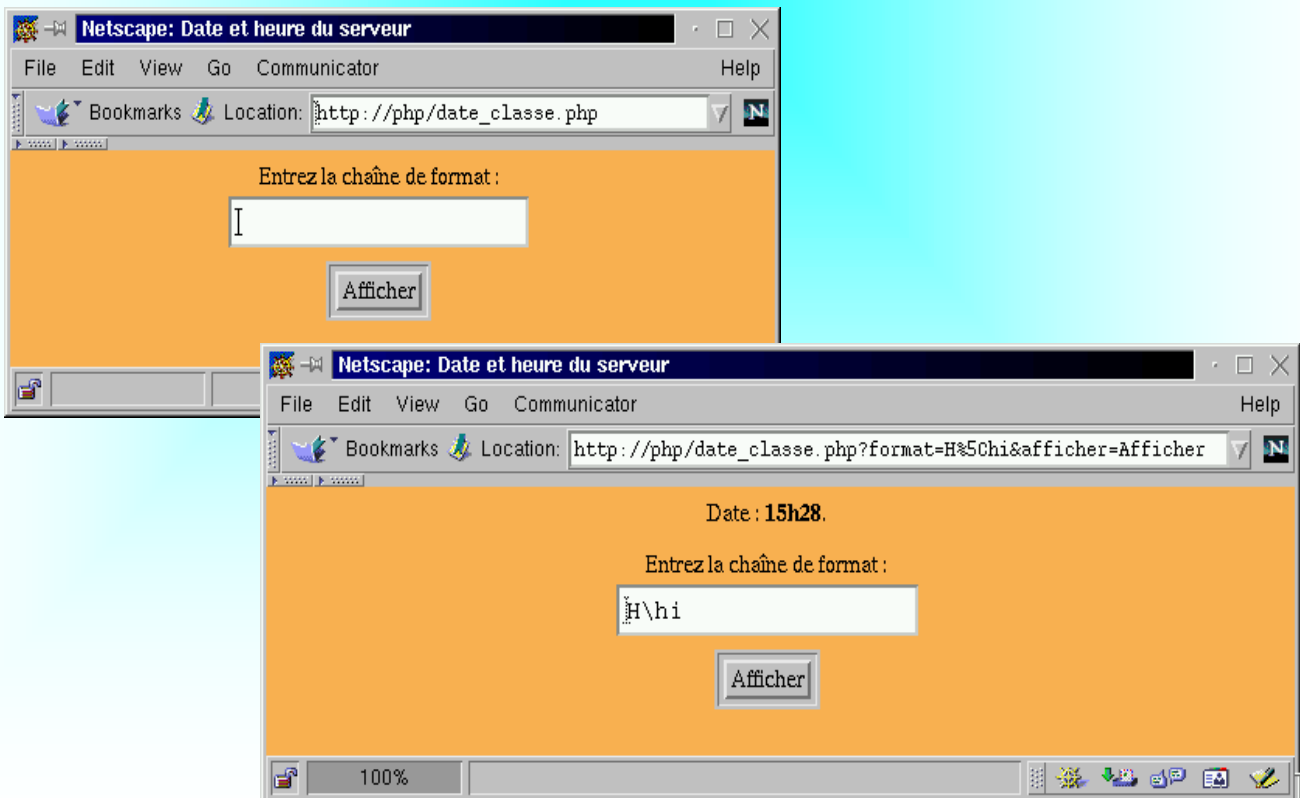
- extends, isset().

**Indications :**

- Sauvegarder date\_instance.php dans date\_classe.php.
- Créer une classe sortie\_date, héritant de la classe sortie, et dont une instance réalise l'affichage de l'heure du serveur.
- Créer une instance de la classe sortie\_date.

**Aller plus loin...**

Faire en sorte que la classe ait deux comportements différents :  
si le script qui la contient est appelé via un formulaire tel que celui présenté avant dans date\_instance.html, alors elle affiche la date, puis le formulaire ;  
sinon, elle affiche simplement le formulaire.



## 8. Accès au système de fichier

### But :

Lire un modèle pour la sortie HTML et effectuer les remplacements nécessaires pour formater la sortie finale et ainsi uniformiser l'aspect d'une application.

### Intérêt :

Lire un fichier du serveur ;  
Effectuer des remplacements dans des chaînes.

### Fichiers :

modele.html (à sauvegarder dans le répertoire courant) ;  
sortie\_modele.inc ;  
modele.php.

### Références :

foreach ;  
file(), pour lire le contenu du modèle ;  
strstr(), pour trouver l'occurrence d'une chaîne dans une autre chaîne ;  
str\_replace(), pour effectuer des remplacement dans une chaîne.

### Indications :

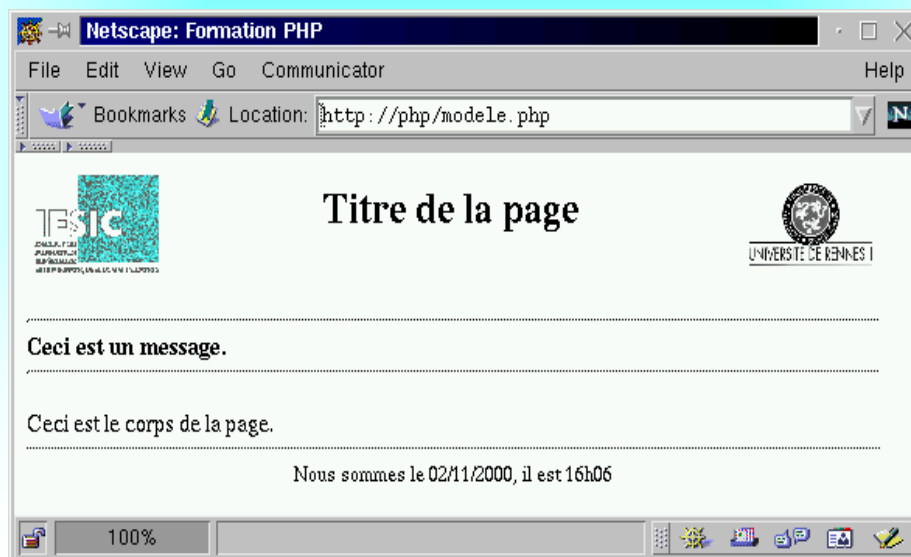
Rapatrier le modèle modele.html dans votre répertoire de travail, et le modifier si vous le désirez ;  
Sauvegarder sortie\_classe.inc dans sortie\_modele.inc et modifier les méthodes debut\_html() et fin\_html() de la classe sortie ;  
Sauvegarder modele.php, qui utilise la nouvelle classe sortie, dans votre répertoire de travail et tester.

Le fichier modele.html est constitué d'instructions HTML, et des champs spéciaux qui doivent être interprétés par les méthodes de la classe sortie.

Les champs spéciaux sont \_\_JOUR\_\_, \_\_HEURE\_\_, \_\_META\_\_, \_\_TITRE\_\_, \_\_COULEUR\_\_, \_\_MESSAGE\_\_, qui doivent être remplacés respectivement par le jour, l'heure, d'éventuels tags HTML <meta ...> supplémentaires, le titre de la page, la couleur de fond de la page, et un message.

La méthode debut\_html() peut parcourir le fichier de modèle jusqu'à rencontrer le champ spécial \_\_CORPS\_\_, en remplaçant les champs spéciaux par les valeurs correspondantes et envoyant le résultat sur la sortie standard du programme.

La méthode fin\_html() peut parcourir le fichier de modèle jusqu'à rencontrer le champ spécial \_\_CORPS\_\_ (sans rien faire), puis remplacer les champs spéciaux de la fin du fichier de modèle.





## 9. Rafraîchissement automatique

**But :** Faire une page dont le contenu se rafraîchit périodiquement.

**Intérêt :**

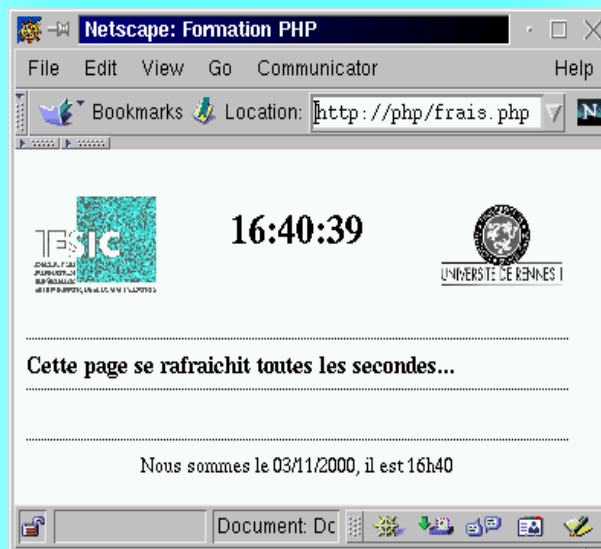
On a parfois besoin de rafraîchir une page, par exemple lorsque celle-ci dépend d'un état qui peut se modifier à n'importe quel moment, ou lorsque l'on souhaite revenir à une page donnée du serveur (la page d'accueil par exemple pour les bornes interactives) au bout d'un certain temps d'inactivité.

**Fichier :** `frais.php`.

**Indications :**

Sauvegarder `modele.php` sous `frais.php` ;

Faire en sorte que `frais.php` affiche l'heure à la seconde près et se rafraîchisse toutes les secondes.



# 10. Sorties non HTML

## But :

Renvoyer via HTTP des données sous un format autre que HTML.

## Intérêt :

Un script PHP peut ainsi renvoyer toute sorte de données (PostScript, PDF, programmes exécutables, archives, données compressées, images, ...).

## Fichier :

autre.php.

## Pointeurs :

header().

## Indications :

Le script autre.php, appelé sans paramètre, présente une liste de boutons correspondant à des fichiers du serveur ; L'appui d'un des boutons doit envoyer le fichier avec le bon type MIME ;

Les fichiers à renvoyer sont les suivants :

/etc/mime.types : (text/plain)

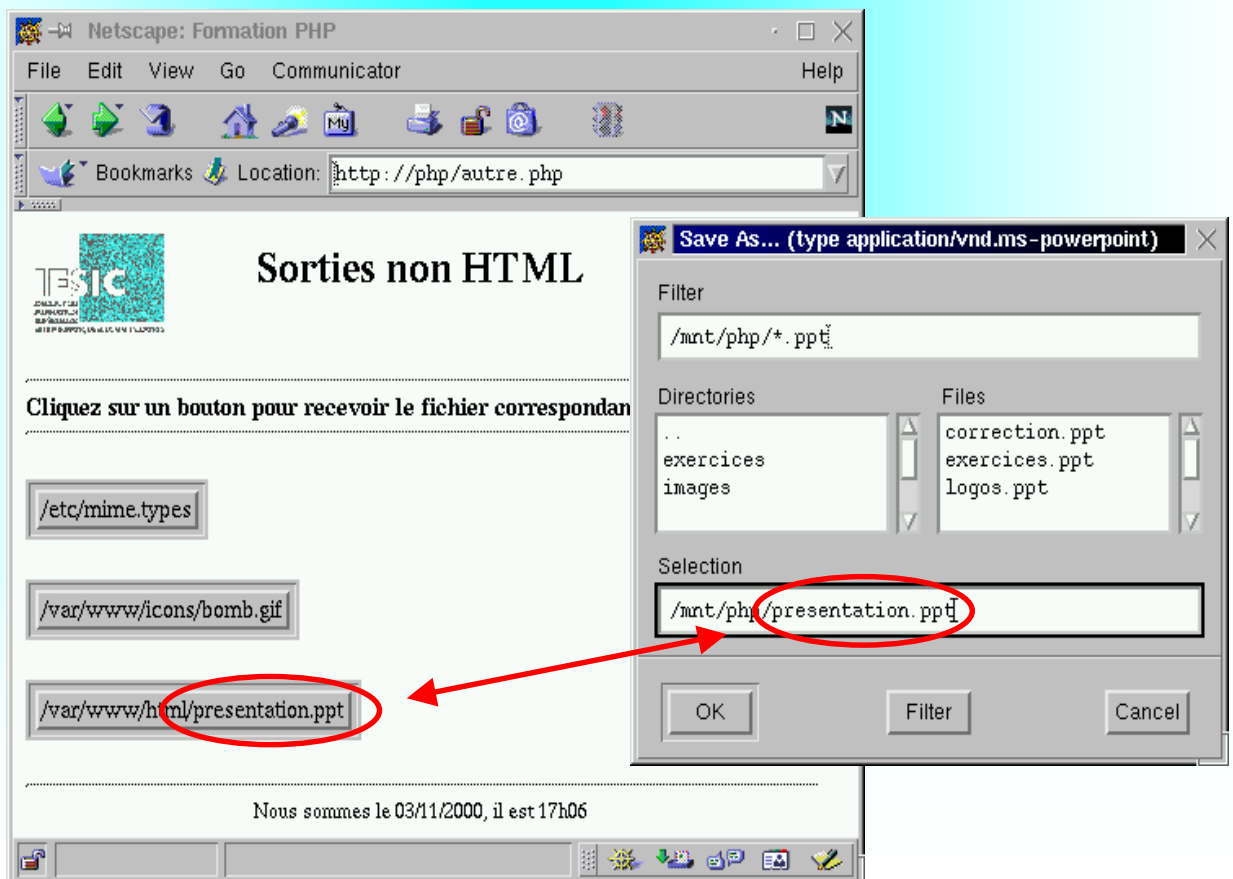
/var/www/icons/bomb.gif : (image/gif)

/var/www/html/presentation.ppt : (application/vnd.ms-powerpoint)

## Aller plus loin :

Passer le nom du fichier dans la réponse pour faire en sorte que le fichier soit sauvegardé sous le bon nom (proposé dans la boîte de dialogue des clients lors de la sauvegarde sur le disque).

Utiliser un tableau associatif pour pouvoir ajouter des fichiers à volonté.





# 11. Cookies

**But :** Connaître le nombre de pages visitées par un utilisateur.

**Intérêt :** Utiliser des cookies.

**Fichier :** cookie.php.

**Pointeurs :** SetCookie(), header(), readfile().

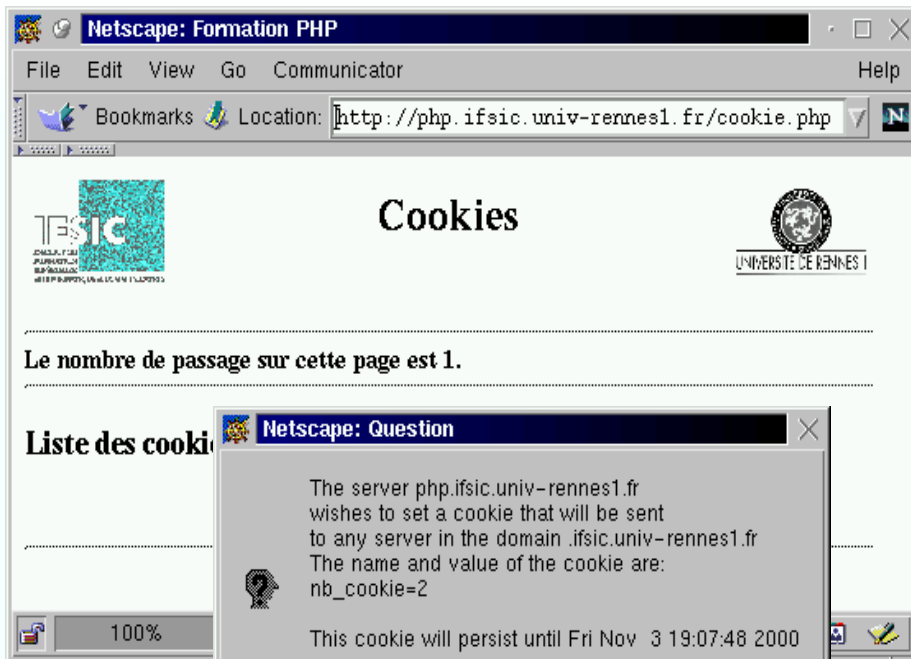
## Indications :

Le but de l'exercice est de positionner un cookie (nb\_cookie) entier, incrémenté à chaque requête, qui donne ainsi le nombre de passages de l'utilisateur ;  
Lors de cet exercice, il est conseillé de paramétrer le navigateur pour qu'il vous informe de la transmission des cookies ;  
Pour vous aider à mettre au point, le script affichera les différents cookies présents à chaque requête.

## Aller plus loin :

Faire en sorte que le script soit portable sur n'importe quelle machine.





## 12. Authentification

**But :** Forcer l'authentification HTTP et effectuer un contrôle sur le couple user/password donné.

**Intérêt :** Cela permet de ne développer au sein d'une application que la partie authentification elle-même, en s'affranchissant de l'interface.

**Fichier :** `auth.php`.

**Pointeurs :** `header()`.

### **Indications :**

Si aucune authentification n'est donnée (`$PHP_AUTH_USER` absent), la demander ;

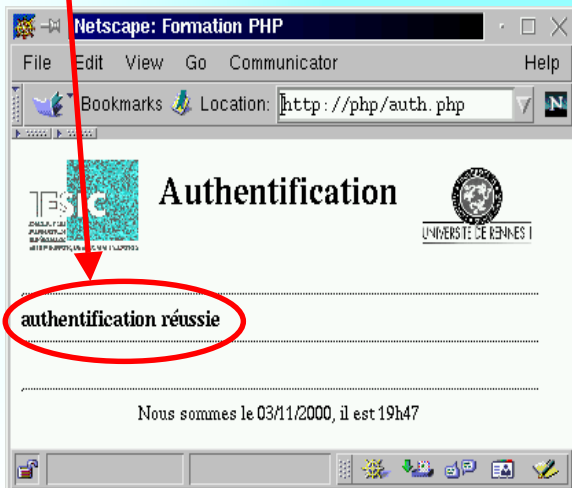
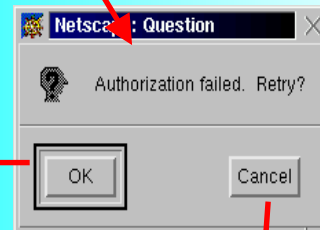
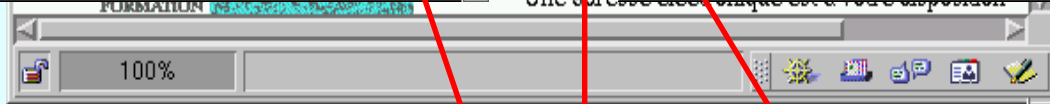
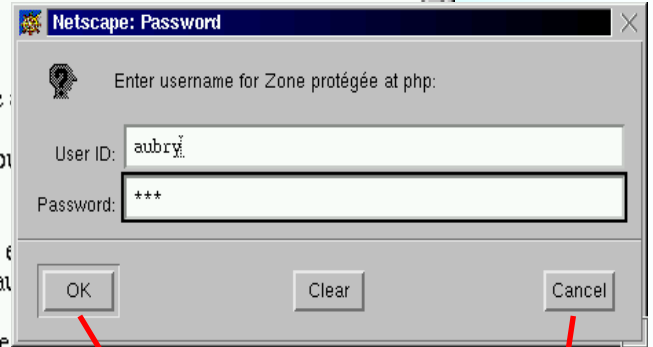
Si une authentification est donnée, la comparer avec un couple donné (par ex : `ifsic/mdp`) :

si le couple correspond, afficher le succès de l'authentification ;

sinon redemander l'authentification (erreur 401).

[debut.doc](#)

(schéma page suivante)



## 14. Exécution de requêtes SQL

**But :** Réaliser une interface pour exécuter des requêtes SQL dans une base de données MySQL.

**Intérêt :** Tester un SGBD MySQL avec différents paramètres.

**Fichier :** interface.php.

**Pointeurs :** `mysql_connect()`, `mysql_pconnect()`, `mysql_select_db()`, `mysql_query()`, `mysql_db_query()`, `mysql_num_rows()`, `mysql_affected_rows()`, `mysql_fetch_assoc()`, `array_keys()`.

### Indications :

Sauvegarder `tables.php` sous `interface.php` ;

Le script doit en plus des paramètres de `interface.php` demander une requête

La requête doit être exécutée sur la base et le serveur choisi ;

Dans tous les cas, le script doit afficher le nombre de colonnes affectées par la requêtes ;

Dans le cas des requêtes SELECT, il doit également afficher les colonnes affectées dans un tableau.

La requête a été exécutée correctement.

Serveur :  Base :

Utilisateur :  Mot de passe :

Requête :

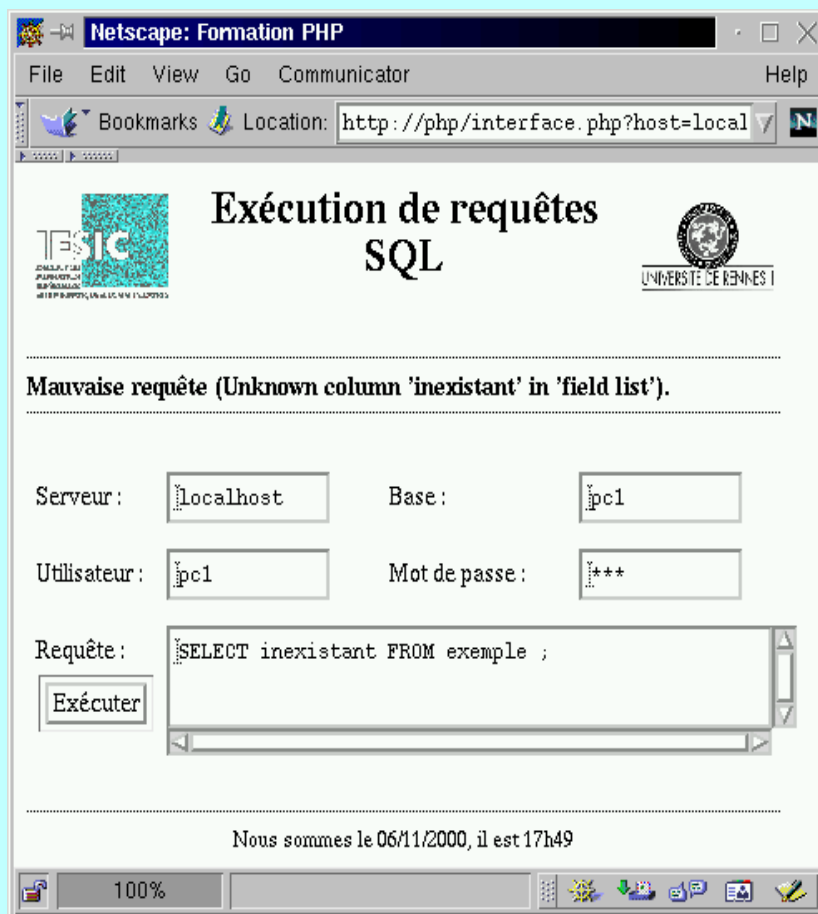
Nombre de lignes affectées : 2

nombre	texte	reel	date
1	texte d'exemple	3	2000-09-23 15:46:02
-43	deuxième texte d'exemple	5.23	2000-10-30 16:23:22

Nous sommes le 06/11/2000, il est 17h48

(voir autre capture d'écran page suivante)





## 15. Traçage d'un utilisateur à l'aide de sessions

**But :** Tracer le parcours d'un utilisateur.

**Intérêt :** Se familiariser avec les sessions.

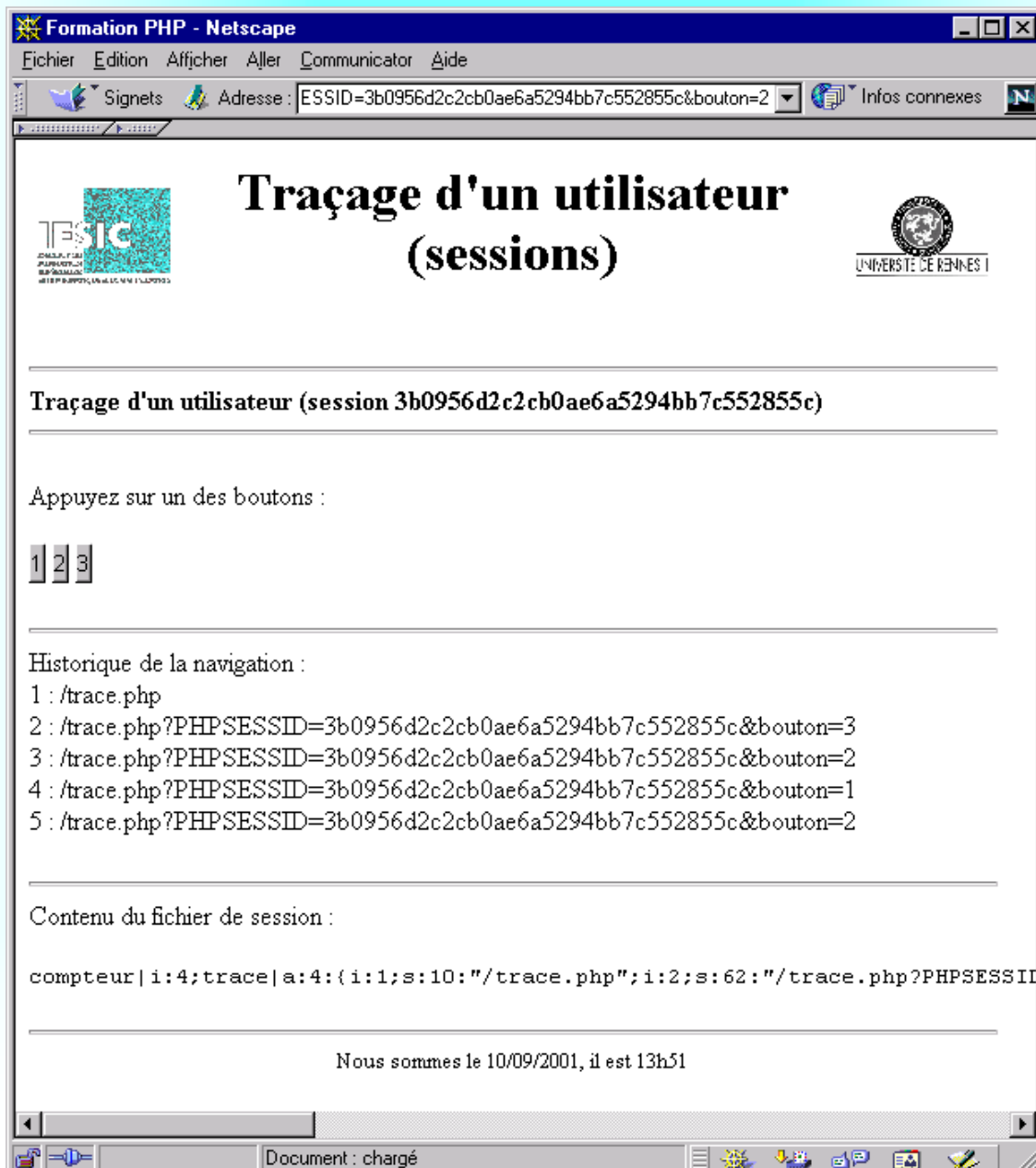
**Fichier :** trace.php.

**Pointeurs :** session\_start(), session\_register().

### Indications :

Le script doit afficher :

- l'identificateur de session courant ;
- un historique de la navigation de l'utilisateur pendant sa session.
- Le contenu du fichier de session sur le serveur (situé dans /tmp).



## 16. Téléchargement de fichiers

**But :** Télécharger un fichier sur le serveur.

**Intérêt :** Transmettre de gros volumes de données en une seule fois.

**Fichier :** telecharge.php.

**Pointeurs :** move\_uploaded\_file()

### Indications :

Le script telecharge.php doit proposer une interface simple pour choisir un fichier sur le système local du client ; Lorsqu'un fichier est effectivement téléchargé, il doit être stocké dans le répertoire /tmp/telechargement du serveur ; Dans tous les cas (pour contrôler), le script affiche la liste des fichiers présents dans le répertoire de téléchargement (echo "

```
".`ls -al /tmp/telechargement`."</pre>").
```

The screenshot shows a Netscape browser window with the address bar set to `http://php/telecharge.php`. The page content includes a form for uploading a file, a terminal window showing the output of a `ls` command, and a confirmation message. A 'Browse...' dialog box is open, showing the local file system with the file `logo_ifsic.gif` selected. A red arrow points from the selected file in the dialog to the file name in the terminal output.

**Terminal Output (Left):**

```
total 8
drwxrwxrwt  2 root  root    4096 Nov  3 21:03 .
drwxrwxrwt  7 root  root    4096 Nov  3 21:02 ..
```

**Terminal Output (Right):**

```
total 16
drwxrwxrwt  2 root  root    4096 Nov  3 21:11 .
drwxrwxrwt  7 root  root    4096 Nov  3 21:11 ..
-rw-----  1 apache apache  5721 Nov  3 21:11 logo_ifsic.gif
```

## 17. Exécution de programmes externes

### But :

Exécuter un programme du serveur.

### Intérêt :

Malgré l'étendue des bibliothèques, certaines actions ne peuvent être effectuées qu'à l'aide d'un appel système ; Attention : l'exécution d'une commande sur le serveur constitue potentiellement un énorme trou de sécurité.

### Fichier :

execution.php.

### Références :

exec(), passthru(), system().

### Indications :

Le script execution.php doit afficher la liste des packages RPM installés sur le serveur. Cette liste peut être obtenue à l'aide de la commande rpm -qa.

### Aller plus loin :

La liste peut être présentée de manière ordonnée.



## 18. Envoi de courriers électroniques

**But :** Envoyer un courrier électronique depuis un programme PHP.

**Intérêt :** Cela peut permettre de prévenir rapidement d'un événement exceptionnel au sein d'une application.

**Fichier :** mail.php.

**Pointeurs :** mail().

**Indications :** Envoyer par mail (à vous-même, pour pouvoir tester) la date, l'origine de la connexion et le nom du script en cours d'exécution.

### Aller plus loin...

Envoyer les informations au format HTML.

The screenshot shows a Netscape browser window titled "Netscape: Formation PHP". The address bar contains "http://php/mail.php". The main content area displays a form titled "Envoi d'un courrier électronique" with the IFSIC logo on the left and the University of Rennes I logo on the right. Below the title, there is a instruction: "Remplissez les champs puis cliquez sur le bouton envoyer." The form consists of three input fields: "Destinataire:" (empty), "Sujet:" (containing "(pas de sujet)"), and "Message:" (containing "Entrez votre message ici."). Below these fields is an "Envoyer" button. At the bottom of the page, it says "Nous sommes le 03/11/2000, il est 15h34". The browser's status bar at the bottom shows "100%" zoom and various system icons.

## 19. Création/modification d'images

**But :** Créer une image et la renvoyer via HTTP.

**Intérêt :** Il est parfois nécessaire de véhiculer l'information sous la forme d'une image. PHP permet de créer ces images de manière dynamique (à chaque requête si nécessaire).

### Fichiers :

param\_image.php (à sauvegarder dans votre répertoire de travail) ;  
image.php (à écrire).

**Pointeurs :** ImageCreate(), ImageColorAllocate(), ImageString(), header(), ImagePng().

### Indications :

Sauvegarder param\_image.php dans votre répertoire de travail. Ce script offre un formulaire demandant les paramètres :

- l et h (la hauteur et la largeur de l'image) ;
- bg\_r, bg\_g, bg\_b (les indices RGB de la couleur de fond), fg\_r, fg\_g, fg\_b (les indices RGB de texte) ;
- text (le texte à afficher) ;
- x et y (la position du texte dans l'image).

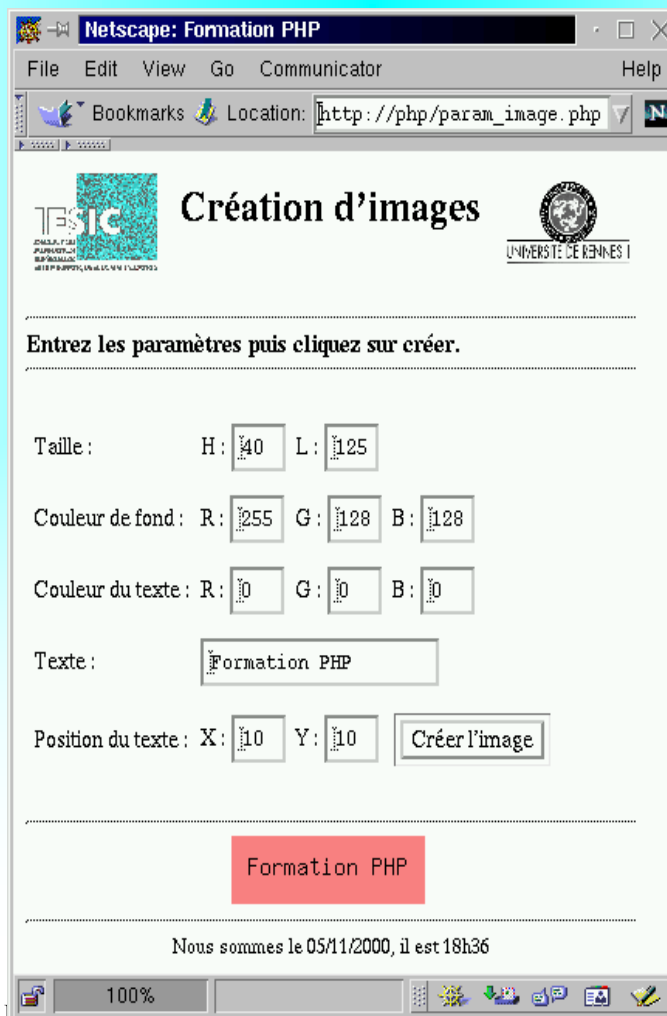
Il fait indirectement appel au script image.php, en comportant le tag suivant :

```
"<img src=\"image.php?l=$l&h=$h&bg=$bg_r:$bg_g:$bg_b&fg=$fg_r:$fg_g:$fg_b&text=
.urlencode($text) . \"&x=$x&y=$y\">"
```

On trouvera par exemple :

```
.
```

Le but de l'exercice est d'écrire le script image.php.



## 20. Mise en œuvre d'un compteur graphique avec MySQL

**But :** Écrire un programme qui affiche de manière graphique le nombre d'accès à une page d'un site.

### Intérêt :

- Consulter et mettre à jour une base de données ;
- Utiliser les bibliothèques graphiques ;
- Soigner l'entête HTTP.

### Fichier :

test\_compteur.php (à sauvegarder dans votre répertoire de travail) ;  
compteur.php (à écrire).

### Indications :

Créer dans votre base de données une table « compteur », comportant les champs « page » (texte, qui indique la page dont on compte le nombre d'accès, et « nombre » (entier, qui indique le nombre d'accès à la page)

```
CREATE TABLE compteur ( page text, nombre integer) ;
```

Pour trouver le nombre d'accès à une page, utiliser la requête suivante :

```
SELECT nombre FROM compteur WHERE page = 'http://...' ;
```

Lorsqu'aucune entrée n'existe dans la base pour une page, il faut la créer :

```
INSERT INTO compteur (page,text) VALUES ('http://...',1) ;
```

Penser ensuite à mettre à jour le nombre de visites pour la page :

```
UPDATE compteur SET nombre = nombre + 1 WHERE page = 'http://...' ;
```

Lorsque l'accès à la base de données ou la requête se déroule mal, afficher la chaîne « erreur ».







# **Solutions aux exercices**



# 1. Configuration du serveur

```
<?
// info.php
// affichage des informations de configuration du serveur

// affichage de l'entête HTML (fond de la page en orange)
echo "
<html>
  <head>
    <title>Configuration du serveur</title>
  </head>
  <body bgcolor=\"ffb050\">
" ;

// affichage des informations de configuration du serveur
phpinfo() ;

// affichage de la fin de la sortie HTML
echo "
  </body>
</html>
" ;

?>
```

# 2. Utilisation de fonctions

```
<?
// date.php
// affichage de la date et de l'heure du serveur

// affichage de l'entête HTML (fond de la page en orange)
echo "
<html>
  <head>
    <title>Configuration du serveur</title>
  </head>
  <body bgcolor="ffb050">
" ;

// affichage de la date et de l'heure du serveur
$jour = date("d/m/Y") ;
$heure = date("H\hi") ;
echo "
<p>Nous sommes le <b>$jour</b> et il est exactement <b>$heure</b>.</p>
" ;

// affichage de la fin de la sortie HTML
echo "
  </body>
</html>
" ;

?>
```

### 3. Écriture de fonctions

```
<?
// date_fonctions.php
// affichage de la date et de l'heure du serveur
// (fonctions)

// affichage de l'entête HTML
function debut_html($titre="(pas de titre)")
{
    echo "<html><head><title></title></head><body bgcolor=\"fffb050\">" ;
}

// affichage de la fin de la sortie HTML
function fin_html()
{
    echo "</body></html>" ;
}

debut_html("Date et heure du serveur") ;
// affichage de l'heure du serveur
$jour = date("d/m/Y") ;
$heure = date("H\hi") ;
echo "
<p align=\"center\">Nous sommes le <b>$jour</b> et il est exactement <b>$heure</b>.</p>
" ;
fin_html() ;

?>
```

### 4. Inclusion de fichiers

```
<?
// sortie.inc
// fonctions d'affichage de la sortie HTML

// affichage de l'entête HTML
function debut_html($titre="(pas de titre)")
{
    echo "<html><head><title>$titre</title></head><body bgcolor=\"fffb050\">" ;
}

// affichage de la fin de la sortie HTML
function fin_html()
{
    echo "</body></html>" ;
}
?>

<?
// date_appel.php
// affichage de la date et de l'heure du serveur
// (inclusion de sortie.inc)

require("sortie.inc") ;
```

```

debut_html("Date et heure du serveur") ;
// affichage de l'heure du serveur
$jour = date("d/m/Y") ;
$heure = date("H\hi") ;
echo "
<p align=\"center\">Nous sommes le <b>$jour</b> et il est exactement <b>$heure</b>.</p>
" ;
fin_html() ;
?>

```

## 5. Paramètres CGI

```

<! date_format.html >
<html>
<head>
<title>Date et heure du serveur (formulaire)</title>
</head>
<body bgcolor="ffb050">
<center>
<form action="date_format.php" method="get">
Entrez la chaîne de format :<br>
<input type="text" name="format"><br>
<input type="submit" name="afficher" value="Afficher">
</form>
</center>
</body>
</html>

```

```

<?
// date_format.php
// affichage de la date et de l'heure du serveur
// (inclusion de sortie.inc)

require("sortie.inc") ;

debut_html("Date et heure du serveur") ;
// affichage de l'heure du serveur
$date_heure = date($_REQUEST["format"]) ;
echo "<p align=\"center\">
La date affichée avec la chaîne de format donnée
(\"".$_REQUEST["format"]."\")
est la suivante :<br>
<b>$date_heure</b>.</p>" ;
fin_html() ;
?>

```

## 6. Création d'une classe

```
<?
// sortie_classe.inc
// classe d'affichage de la sortie HTML

class sortie {

//
// déclaration des attributs de la classe
//
var
    $couleur,      // couleur de fond de la page
    $titre ;       // titre de la page

//
// constructeur de la classe (fonction de même nom que la classe)
//
function sortie($couleur="ffb050",$titre="(pas de titre)")
{

    $this->couleur = $couleur ;
    $this->titre = $titre ;
}

//
// affichage de l'entête HTML
//
function debut_html()
{
    echo "<html><head><title>$this->titre</title></head><body bgcolor=\"\$this->couleur\">" ;
}

//
// affichage de la fin de la sortie HTML
//
function fin_html()
{
    echo "</body></html>" ;
}

} // fin de la classe sortie
?>

<?
// date_instance.php
// affichage de la date et de l'heure du serveur
// (inclusion de sortie.inc, utilisation de la classe sortie)

require("sortie_classe.inc") ;

//
// création de l'instance de la classe sortie
//
$s = new sortie("ffb050","Date et heure du serveur") ;

//
// affichage du début de la sortie HTML
//
$s->debut_html() ;
```

```

//
// affichage de l'heure du serveur
//
$date_heure = date($_REQUEST["format"]) ;
echo "
<p align=\"center\">
La date affichée avec la chaîne de format donnée (\".$_REQUEST["format"].\")
est la suivante :<br>
<b>$date_heure</b>.</p>
" ;

//
// affichage de la fin de la sortie HTML
//
$s->fin_html() ;

?>

```

## 7. Héritage

```

<?
// date_classe.php
// affichage de la date et de l'heure du serveur (inclusion de sortie.inc, héritage de la
classe sortie)

require("sortie_classe.inc") ;

class date extends sortie {          // la classe date hérite de la classe sortie

function date($couleur,$titre)
{ // appel du constructeur de la classe encêtre (non automatique)
  $this->sortie($couleur,$titre) ;
}

function affiche_date($format)      // affichage de la date selon le format voulu
{
  $date_heure = date($format) ;
  echo "<p align=\"center\">Date : <b>$date_heure</b>.</p>" ;
}

function affiche_formulaire($format) // affichage du formulaire
{
  echo "
<center><form action=\"$_SERVER[PHP_SELF]\" method=\"get\">
Entrez la chaîne de format :<br>
<input type=\"text\" name=\"format\" value=\"$format\"><br>
<input type=\"submit\" name=\"afficher\"
value=\"Afficher\"><br>
</form></center>
" ;
}

} // fin de la classe date

$d = new date("ffb050","Date et heure du serveur") ;
$d->debut_html() ;                      // début de la sortie HTML
if ( isset($_REQUEST["afficher"]) )
  $d->affiche_date($_REQUEST["format"]) ; // affichage de la date
$d->affiche_formulaire($_REQUEST["format"]) ; // affichage du formulaire
$d->fin_html() ;                        // fin de la sortie HTML

?>

```

## 8. Accès au système de fichier

```
<?
// sortie_modele.inc
// classe d'affichage de la sortie HTML

define("MODELE", "modele.html") ;

class sortie {

// déclaration des attributs de la classe
var
    $couleur, // couleur de fond de la page
    $titre ; // titre de la page

// constructeur de la classe (fonction de même nom que la classe)
function sortie($couleur="ffffff", $titre="(pas de titre)")
{
    $this->couleur = $couleur ;
    $this->titre = $titre ;
}

// affichage de l'entête HTML
function debut_html($message="", $meta="")
{
    $data = file(MODELE) ;
    $jour = date("d/m/Y") ;
    $heure = date("H\hi") ;
    foreach ( $data as $ligne )
    {
        // on sort de la méthode quand on rencontre "__CORPS__"
        if ( strstr($ligne, "__CORPS__") )
            return ;
        $ligne = str_replace("__META__", $meta, $ligne) ;
        $ligne = str_replace("__COULEUR__", $this->couleur, $ligne) ;
        $ligne = str_replace("__TITRE__", $this->titre, $ligne) ;
        $ligne = str_replace("__MESSAGE__", $message, $ligne) ;
        $ligne = str_replace("__JOUR__", $jour, $ligne) ;
        $ligne = str_replace("__HEURE__", $heure, $ligne) ;
        echo $ligne ;
    }
}
```



```

// affichage de la fin de la sortie HTML
function fin_html()
{
    $data = file(MODELE) ;
    $jour = date("d/m/Y") ; $heure = date("H\hi") ;
    $found = FALSE ;
    foreach ( $data as $ligne )
    {
        // on ne fait quelque chose que lorsque l'on a rencontré "__CORPS__"
        if ( ! $found )
            $found = strstr($ligne,"__CORPS__") ;
        else
        {
            $ligne = str_replace("__META__",$meta,$ligne) ;
            $ligne = str_replace("__COULEUR__",$this->couleur,$ligne) ;
            $ligne = str_replace("__TITRE__",$this->titre,$ligne) ;
            $ligne = str_replace("__MESSAGE__",$message,$ligne) ;
            $ligne = str_replace("__JOUR__",$jour,$ligne) ;
            $ligne = str_replace("__HEURE__",$heure,$ligne) ;
            echo $ligne ;
        }
    }
}

} // fin de la classe sortie
?>

<?
// modele.php
// (inclusion de sortie_modele.inc, utilisation de la classe sortie)

require("sortie_modele.inc") ;

$s = new sortie("ffffff","Titre de la page") ;
$s->debut_html("Ceci est un message.") ;
echo "Ceci est le corps de la page.\n" ;
$s->fin_html() ;
?>

```

## 9. Rafraîchissement automatique

```

<?
// frais.php
// rafraîchissement automatique

require("sortie_modele.inc") ;

$s = new sortie("ffffff","<font size=\"+3\">.date("H:i:s")."</font>") ;

$s->debut_html("Cette page se rafraichit toutes les secondes...",
              "<meta http-equiv=\"refresh\" content=\"1\"") ;

$s->fin_html() ;

?>

```

## 10. Sorties non HTML

```
<?
// autre.php

// sorties non HTML

if ( isset($_REQUEST["fichier"]) )
{
    // on positionne le type MIME de la réponse
    header("Content-type: ".$_REQUEST["type_mime"]) ;
    // on indique le nom du fichier transmis
    header("Content-Disposition: attachment;
filename=\"".$_REQUEST["fichier"].\"");
    // on envoie le fichier
    readfile($_REQUEST["fichier"]) ;
}
else
{
    $stab = array (
        "/etc/mime.types"           => "text/plain",
        "/var/www/icons/bomb.gif"   => "image/gif",
        "/var/www/html/presentation.ppt" => "application/vnd.ms-powerpoint"
    ) ;
    require("sortie_modele.inc") ;
    $s = new sortie("ffffff", "Sorties non HTML") ;
    $s->debut_html("Cliquez sur un bouton pour recevoir le fichier correspondant...") ;
    foreach ( $stab as $fichier => $type_mime )
        echo "
<form>
<input type=\"hidden\" name=\"type_mime\" value=\"$type_mime\">
<input type=\"submit\" name=\"fichier\" value=\"$fichier\">
</form>
" ;
    $s->fin_html() ;
}
?>
```

## 11. Cookies

```
<?
// cookie.php
// comptage du nombre de passages

$nb_cookie = $_COOKIE["nb_cookie"] ;

if ( empty($nb_cookie) )
{
    // on initialise le cookie...
    SetCookie("nb_cookie", "1", time()+3600, "/",
        ereg_replace("^[\.\.]*\.", ".", $_SERVER["SERVER_NAME"])) ;
    // ... et la variable locale
    $nb_cookie = 1 ;
}
else
{
    // on incrémente la variable locale
    $nb_cookie ++ ;
    // et on la redonne en cookie pour la prochaine visite
    SetCookie("nb_cookie", $nb_cookie, time()+3600, "/",
        ereg_replace("^[\.\.]*\.", ".", $_SERVER["SERVER_NAME"])) ;
}
}
```

```

require("sortie_modele.inc") ;
$s = new sortie("ffffff","Cookies") ;
$s->debut_html("Le nombre de passage sur cette page est $nb_cookie.") ;

// titre et début de la partie préformatée
echo "<h3>Liste des cookies :</h3><pre>" ;

foreach ($_COOKIE as $nom => $valeur)
    echo "$nom =&gt; $valeur\n" ;

// fin de la partie préformatée
echo "</pre>" ;

$s->fin_html() ;

?>

```

## 12. Authentification

```

<?
// auth.php
// forçage de l'authentification via HTTP

include ("sortie_modele.inc") ;

// renvoi d'une erreur 401 pour l'authentification
function demande_auth()
{
    header("WWW-Authenticate: Basic realm=\"Zone protégée\"");
    header("HTTP/1.0 401 Unauthorized");
    $s = new sortie("ffffff","Authentification") ;
    $s->debut_html("Cette zone est réservée à ceux qui peuvent yrentrer...") ;
    $s->fin_html() ;
}

if (!isset($_SERVER["PHP_AUTH_USER"]))
    demande_auth() ; // pas d'authentification
else
{
    if ( ($_SERVER["PHP_AUTH_USER"] == "ifsic")
        && ($_SERVER["PHP_AUTH_PW"] == "mdp") )
    { // authentification réussie
        $s = new sortie("ffffff","Authentification") ;
        $s->debut_html("authentification réussie") ;
        $s->fin_html() ;
    }
    else // mauvaise authentification
        demande_auth() ;
}
?>

```

## 13. Exécution de requêtes SQL

```

<?
// interface.php
// interface pour exécuter des requêtes sur une base de données MySQL

require("sortie_modele.inc") ;

```

```

class interface extends sortie {

function interface()
{
    $this->sortie("ffffff","Exécution de requêtes SQL") ;
    // lecture des paramètres
    $this->host = $_REQUEST["host"] ;
    if ( empty($this->host) ) $this->host = "localhost" ;
    $this->db    = $_REQUEST["db"] ;
    $this->user  = $_REQUEST["user"] ;
    $this->pw    = $_REQUEST["pw"] ;
    $this->req   = chop($_REQUEST["req"]) ;

    if ( isset($_REQUEST["executer"]) )
        $this->execution() ;
    else
        {
            $this->debut_html("Remplissez les champs et cliquer le bouton \"Exécuter\".") ;
            $this->affiche_dialogue() ;
            $this->fin_html() ;
        }
}

function affiche_dialogue()
{
    echo "
<form action=\"".$_SERVER["PHP_SELF"]."\" method=\"get\">
<table>
<tr><td nowrap>Serveur :</td>
<td><input type=\"text\" name=\"host\" value=\"$this->host\" size=\"12\"></td>
<td nowrap>Base :</td>
<td><input type=\"text\" name=\"db\"    value=\"$this->db\" size=\"12\"></td></tr>
<tr><td nowrap>Utilisateur :</td>
<td><input type=\"text\" name=\"user\" value=\"$this->user\" size=\"12\"></td>
<td nowrap>Mot de passe :</td>
<td><input type=\"password\" name=\"pw\"    value=\"$this->pw\" size=\"12\"></td></tr>
<tr><td nowrap>Requête :<br>
<input type=\"submit\" name=\"executer\" value=\"Exécuter\"></td>
<td colspan=\"3\"><textarea name=\"req\" value=\"$this->req\" cols=\"50\"
rows=\"3\">$this->req</textarea></td></tr>
</table>
</form>
" ;
}

```

```

function affiche_resultat($query)
{
    echo "<p><hr>Nombre de lignes affectées : ".mysql_affected_rows()."</p>" ;
    if ( @mysql_num_rows($query) )
        { // il y a au moins un résultat
            $arr = mysql_fetch_assoc($query) ;
            echo "<hr><center><table border=\"1\"><tr>" ; // début de la table, début entête
            foreach ( array_keys($arr) as $cle ) // affichage de l'entête (clés)
                echo "<th>$cle</th>" ;
            echo "</tr>" ; // fin entête
            while ( $arr )
                {
                    echo "<tr>" ; // début ligne résultat
                    foreach ( $arr as $valeur ) // champ résultat
                        echo "<td>$valeur</td>" ; // champ résultat
                    echo "</tr>" ; // fin ligne résultat
                    $arr = mysql_fetch_assoc($query) ;
                }
            echo "</table></center>" ; // fin table
        }
}

function execution()
{
    if ( $link = @mysql_connect($this->host,$this->user,$this->pw) )
        {
            if ( mysql_select_db($this->db,$link) )
                {
                    if ( !($query = mysql_query($this->req)) )
                        {
                            $this->debut_html("Mauvaise requête (".mysql_error().").") ;
                            $this->affiche_dialogue() ;
                        }
                    else
                        {
                            $this->debut_html("La requête a été exécutée correctement.") ;
                            $this->affiche_dialogue() ;
                            $this->affiche_resultat($query) ;
                        }
                    $this->fin_html() ;
                }
            else
                {
                    $this->debut_html("Sélection de la base de données impossible
(".mysql_error().").") ;
                    $this->affiche_dialogue() ;
                    $this->fin_html() ;
                }
            mysql_close($link) ;
        }
    else
        {
            $this->debut_html("Connexion impossible ($php_errormsg).") ;
            $this->affiche_dialogue() ;
            $this->fin_html() ;
        }
}

} // fin de la classe interface

$i = new interface() ;

?>

```

## 14. Traçage d'un utilisateur à l'aide de sessions

```
<?
// trace.php
// utilisation de sessions

require("sortie_modele.inc") ;

$s = new sortie("ffffff","Traçage d'un utilisateur (sessions)") ;

// enregistrement de l'entier compteur et du tableau trace en variables de session

session_start() ;

session_register("compteur") ;
session_register("trace") ;

// enregistrement dans le tableau de la requête courante
$compteur ++ ;
$trace[$compteur] = $_REQUEST_URI ;

// affichage de l'entête et de différents boutons pour simuler un parcours

$s->debut_html("Traçage d'un utilisateur (session $_SESSION)") ;

echo "<p>Appuyez sur un des boutons :</p>
<form action=trace.php method=get>
<input type=submit name=bouton value=1>
<input type=submit name=bouton value=2>
<input type=submit name=bouton value=3>
</form>
" ;

// affichage de l'historique de la navigation
echo "<p><hr>Historique de la navigation :<br>" ;
foreach ( $trace as $cle => $valeur)
{
    echo "$cle : $valeur<br>" ;
}
echo "</p>" ;

// affichage du contenu du fichier de session (pas très recommandé ;- )
if ( file_exists("/tmp/sess_$_SESSION") )
{
    echo "<p><hr>Contenu du fichier de session :<pre>" ;
    readfile("/tmp/sess_$_SESSION") ;
    echo "</pre></p>" ;
}
else
    echo "<p><hr>Fichier de session (/tmp/sess_$_SESSION) introuvable.</p>" ;

$s->fin_html() ;

?>
```

## 15. Téléchargement de fichiers

```
<?
// telecharge.php
// telechargement de fichier (upload)
define("UPLOAD_DIR","/tmp/telechargement") ;

require("sortie_modele.inc") ;

$s = new sortie("ffffff","Téléchargement de fichiers") ;

if ( isset($_REQUEST["envoyer"]) )
{
    if ( $_HTTP_POST_FILES["fichier"]["tmp_name"] == "none" )
        $s->debut_html("Veuillez préciser le fichier que vous voulez envoyer...") ;
    else
    {
        if (move_uploaded_file($_HTTP_POST_FILES["fichier"]["tmp_name"],
            UPLOAD_DIR."/".$_HTTP_POST_FILES["fichier"]["name"]))
            $s->debut_html("Le fichier a été téléchargé.") ;
        else
            $s->debut_html("Le fichier n'a pas été téléchargé.") ;
    }
}
else
    $s->debut_html("Vous pouvez soumettre un fichier...") ;

echo "
<form enctype=\"multipart/form-data\" action=\"telecharge.php\" method=\"post\">
<input type=\"hidden\" name=\"MAX_FILE_SIZE\" value=\"50000\">
Envoyer ce fichier : <input name=\"fichier\" type=\"file\"><br>
<input type=\"submit\" name=\"envoyer\" value=\"Envoyer\">
</form>
<pre>" ;
passthru("ls -al ".UPLOAD_DIR) ;
echo "</pre>" ;

$s->fin_html() ;

?>
```

## 16. Exécution de programmes externes

```
<?
// execution.php
// exécution d'une commande externe

require("sortie_modele.inc") ;

$s = new sortie("ffffff", "Liste des packages installés sur le serveur") ;

// début de la sortie HTML
$s->debut_html("La liste des packages installés sur le serveur est la suivante :") ;

// début d'affichage préformaté
echo "<pre>" ;

// exécution de la commande et envoi de sa sortie standard
passthru("rpm -qa | sort") ;

// fin de l'affichage préformaté
echo "</pre>" ;

// fin de la sortie HTML
$s->fin_html() ;

?>
```

## 17. Envoi de courriers électroniques

```
<?
// mail.php
// envoi d'un message électronique

require("sortie_modele.inc") ;

$s = new sortie("ffffff") ;

// envoi du message et entête HTML
if ( isset($_REQUEST["send"]) )
{
    $corps = "
<p>date : ".date("\l\e d/m/Y à H\hi")."</p>
<p>origine : "._SERVER["REMOTE_ADDR"]."</p>
<p>programme : "._SERVER["PHP_SELF"]."</p>
<hr>
".$_REQUEST["body"] ;
    mail($_REQUEST["to"],
        $_REQUEST["subject"],
        $corps,
        "Content-Type: text/html\nFrom: Application PHP <root@php>") ;
    $s->debut_html("Le mail a été envoyé.") ;
}
else
    $s->debut_html("Remplissez les champs puis cliquez sur le bouton envoyer.") ;
```



```

// affichage du dialogue
echo "
<form action=\"".$_SERVER["PHP_SELF"]."\" method=\"post\">
<table>
<tr>
  <td nowrap>Destinataire :</td>
  <td><input type=\"text\" size=\"30\" name=\"to\"></td>
</tr>
<tr>
<td nowrap>Sujet :</td>
  <td><input type=\"text\" size=\"50\" name=\"subject\" value=\"(pas de sujet)\"></td>
</tr>
<tr><td nowrap>Message :</td>
  <td><textarea name=\"body\" cols=\"50\" rows=\"5\">
    Entrez votre message ici.
  </textarea></td>
</tr>
<tr>
  <td nowrap>&nbsp;</td>
  <td><input type=\"submit\" name=\"send\" value=\"Envoyer\"></td>
</tr>
</table>
</form>
" ;

$s->fin_html() ;

?>

```

## 18. Création/modification d'images

```
<?
// image.php
// création d'une image

// analyse des paramètres
$bg_r = strtok($_REQUEST["bg"],":") ;
$bg_g = strtok(":") ;
$bg_b = strtok(":") ;
$fg_r = strtok($_REQUEST["fg"],":") ;
$fg_g = strtok(":") ;
$fg_b = strtok(":") ;

// création de l'image
$image = ImageCreate($_REQUEST["l"],$_REQUEST["h"]) ;

// allocation des couleurs
$bg = ImageColorAllocate($image,$bg_r,$bg_g,$bg_b) ;
$fg = ImageColorAllocate($image,$fg_r,$fg_g,$fg_b) ;

// écriture du texte
ImageString($image,
            4 /*font*/,
            intval($_REQUEST["x"]),
            intval($_REQUEST["y"]),
            $_REQUEST["text"],
            $fg) ;

// modification de l'entête HTTP
header("Content-Type: image/png") ;

// envoi de l'image sur la sortie standard
ImagePng($image) ;

?>
```

## 19. Mise en œuvre d'un compteur graphique avec MySQL

```
<?
// compteur.php
// compteur graphique d'accès aux pages

define("NB_CHIF",5) ; // nombre de chiffres du compteur
define("FONTE",5) ; // fonte utilisée

// envoi d'une image contenant le texte de la chaîne $str (et fin du programme)
function envoi_image($str)
{
    $str = str_pad(substr($str,0,NB_CHIF),NB_CHIF,"0",STR_PAD_LEFT) ;
    $image = ImageCreate(NB_CHIF * ImageFontWidth(FONTE),ImageFontHeight(FONTE)) ;
    $bg = ImageColorAllocate($image,255,0,0) ;
    ImageString($image,FONTE,0,0,$str,ImageColorAllocate($image,255,255,255)) ;
    header ("Expires: 1 1 1970 12:00:00 GMT"); // pour forcer le rechargement
    header("Content-Type: image/png") ; // type MIME renvoyé
    ImagePng($image) ;
    exit ;
}

// on logue dans la base la page appelante
$ref_page = strtok($_SERVER["HTTP_REFERER"],"?") ;
if ( empty($ref_page) )
    envoi_image("référence ?") ; // page appelante non définie
// connexion à MySQL
if ( ! ($link=@mysql_pconnect("localhost","compteur","mdp")) )
    envoi_image("connexion ?") ; // connexion impossible
// sélection de la base de données
if ( ! ($query = @mysql_select_db("comptage")) )
    envoi_image("base ?") ; // base introuvable
// recherche du nombre d'accès pour cette page dans la base comptage
if ( ! ($query = @mysql_query(
    "SELECT nombre FROM log WHERE page = '".addslashes($ref_page)."'")) )
    envoi_image("requête ?") ; // mauvaise requête
if ( @mysql_num_rows($query) == 0 )
    { // 1ère requête sur la page $ref_page
        @mysql_query("INSERT INTO log (nombre,page) VALUES (1,'".addslashes($ref_page)."'") ;
        envoi_image(1) ;
    }
    $arr = @mysql_fetch_array($query) ;
    $n = $arr["nombre"] ;
    @mysql_query("UPDATE log SET nombre = nombre + 1 WHERE page =
    '".addslashes($ref_page)."'") ;
    envoi_image($n) ;

?>
```



# **Configuration de PHP (/etc/php.ini)**



[PHP]

```
;;;;;;;;;;;;;;
; About this file ;
;;;;;;;;;;;;;;
; This file controls many aspects of PHP's behavior.  In order for PHP to
; read it, it must be named 'php.ini'.  PHP looks for it in the current
; working directory, in the path designated by the environment variable
; PHPRC, and in the path that was defined in compile time (in that order).
; Under Windows, the compile-time path is the Windows directory.  The
; path in which the php.ini file is looked for can be overridden using
; the -c argument in command line mode.
;
; The syntax of the file is extremely simple.  Whitespace and Lines
; beginning with a semicolon are silently ignored (as you probably guessed).
; Section headers (e.g. [Foo]) are also silently ignored, even though
; they might mean something in the future.
;
; Directives are specified using the following syntax:
; directive = value
; Directive names are *case sensitive* - foo=bar is different from FOO=bar.
;
; The value can be a string, a number, a PHP constant (e.g. E_ALL or M_PI), one
; of the INI constants (On, Off, True, False, Yes, No and None) or an expression
; (e.g. E_ALL & ~E_NOTICE), or a quoted string ("foo").
;
; Expressions in the INI file are limited to bitwise operators and parentheses:
; | bitwise OR
; & bitwise AND
; ~ bitwise NOT
; ! boolean NOT
;
; Boolean flags can be turned on using the values 1, On, True or Yes.
; They can be turned off using the values 0, Off, False or No.
;
; An empty string can be denoted by simply not writing anything after the equal
; sign, or by using the None keyword:
;
; foo =          ; sets foo to an empty string
; foo = none     ; sets foo to an empty string
; foo = "none"   ; sets foo to the string 'none'
;
; If you use constants in your value, and these constants belong to a dynamically
; loaded extension (either a PHP extension or a Zend extension), you may only
; use these constants *after* the line that loads the extension.
;
; All the values in the php.ini-dist file correspond to the builtin
; defaults (that is, if no php.ini is used, or if you delete these lines,
; the builtin defaults will be identical).
```

```

;;;;;;;;;;;;;
; Language Options ;
;;;;;;;;;;;;;

engine = On          ; Enable the PHP scripting language engine under Apache
short_open_tag = On ; allow the <? tag.  otherwise, only <?php and <script> tags are recognized.
asp_tags = Off      ; allow ASP-style <% %> tags
precision = 14     ; number of significant digits displayed in floating point numbers
y2k_compliance = Off ; whether to be year 2000 compliant (will cause problems with non y2k compliant browsers)
output_buffering = Off ; Output buffering allows you to send header lines (including cookies)
                    ; even after you send body content, in the price of slowing PHP's
                    ; output layer a bit.
                    ; You can enable output buffering by in runtime by calling the output
                    ; buffering functions, or enable output buffering for all files
                    ; by setting this directive to On.
output_handler =    ; You can redirect all of the output of your scripts to a function,
                    ; that can be responsible to process or log it.  For example,
                    ; if you set the output_handler to "ob_gzhandler", than output
                    ; will be transparently compressed for browsers that support gzip or
                    ; deflate encoding.  Setting an output handler automatically turns on
                    ; output buffering.
implicit_flush = Off ; Implicit flush tells PHP to tell the output layer to flush itself
                    ; automatically after every output block.  This is equivalent to
                    ; calling the PHP function flush() after each and every call to print()
                    ; or echo() and each and every HTML block.
                    ; Turning this option on has serious performance implications, and
                    ; is generally recommended for debugging purposes only.
allow_call_time_pass_reference = On ; whether to enable the ability to force arguments to be
                                    ; passed by reference at function-call time.  This method
                                    ; is deprecated, and is likely to be unsupported in future
                                    ; versions of PHP/Zend.  The encouraged method of specifying
                                    ; which arguments should be passed by reference is in the
                                    ; function declaration.  You're encouraged to try and
                                    ; turn this option Off, and make sure your scripts work
                                    ; properly with it, to ensure they will work with future
                                    ; versions of the language (you will receive a warning
                                    ; each time you use this feature, and the argument will
                                    ; be passed by value instead of by reference).

; Safe Mode
safe_mode = Off
safe_mode_exec_dir =
safe_mode_allowed_env_vars = PHP_ ; Setting certain environment variables
                                   ; may be a potential security breach.
                                   ; This directive contains a comma-delimited
                                   ; list of prefixes.  In Safe Mode, the
                                   ; user may only alter environment
                                   ; variables whose names begin with the
                                   ; prefixes supplied here.
                                   ; By default, users will only be able
                                   ; to set environment variables that begin
                                   ; with PHP_ (e.g. PHP_FOO=BAR).
                                   ; Note: If this directive is empty, PHP
                                   ; will let the user modify ANY environment
                                   ; variable!
safe_mode_protected_env_vars = LD_LIBRARY_PATH ; This directive contains a comma-
                                                ; delimited list of environment variables,
                                                ; that the end user won't be able to
                                                ; change using putenv().
                                                ; These variables will be protected
                                                ; even if safe_mode_allowed_env_vars is
                                                ; set to allow to change them.

disable_functions = ; This directive allows you to disable certain
                    ; functions for security reasons.  It receives
                    ; a comma separated list of function names.
                    ; This directive is *NOT* affected by whether
                    ; Safe Mode is turned on or off.

```



```

; Colors for Syntax Highlighting mode. Anything that's acceptable in <font color=???> would work.
highlight.string      = #DD0000
highlight.comment    = #FF8000
highlight.keyword    = #007700
highlight.bg         = #FFFFFF
highlight.default    = #0000BB
highlight.html       = #000000

; Misc
expose_php = On      ; Decides whether PHP may expose the fact that it is installed on the
                    ; server (e.g., by adding its signature to the Web server header).
                    ; It is no security threat in any way, but it makes it possible
                    ; to determine whether you use PHP on your server or not.

;;;;;;;;;;;;;;;;;;;;;
; Resource Limits ;
;;;;;;;;;;;;;;;;;;;;;

max_execution_time = 30      ; Maximum execution time of each script, in seconds
memory_limit = 8M           ; Maximum amount of memory a script may consume (8MB)

;;;;;;;;;;;;;;;;;;;;;
; Error handling and logging ;
;;;;;;;;;;;;;;;;;;;;;
; error_reporting is a bit-field. Or each number up to get desired error reporting level
; E_ALL- All errors and warnings
; E_ERROR- fatal run-time errors
; E_WARNING- run-time warnings (non fatal errors)
; E_PARSE- compile-time parse errors
; E_NOTICE- run-time notices (these are warnings which often result from a bug in
; your code, but it's possible that it was intentional (e.g., using an
; uninitialized variable and relying on the fact it's automatically
; initialized to an empty string)
; E_CORE_ERROR- fatal errors that occur during PHP's initial startup
; E_CORE_WARNING- warnings (non fatal errors) that occur during PHP's initial startup
; E_COMPILE_ERROR- fatal compile-time errors
; E_COMPILE_WARNING- compile-time warnings (non fatal errors)
; E_USER_ERROR- user-generated error message
; E_USER_WARNING- user-generated warning message
; E_USER_NOTICE- user-generated notice message
; Examples:
; error_reporting = E_ALL & ~E_NOTICE; show all errors, except for notices
; error_reporting = E_COMPILE_ERROR|E_ERROR|E_CORE_ERROR; show only errors
error_reporting = E_ALL & ~E_NOTICE      ; Show all errors except for notices
display_errors = On                    ; Print out errors (as a part of the output)
                                        ; For production web sites, you're strongly encouraged
                                        ; to turn this feature off, and use error logging instead (see below).
                                        ; Keeping display_errors enabled on a production web site may reveal
                                        ; security information to end users, such as file paths on your Web server,
                                        ; your database schema or other information.
display_startup_errors = Off           ; Even when display_errors is on, errors that occur during
                                        ; PHP's startup sequence are not displayed. It's strongly
                                        ; recommended to keep display_startup_errors off, except for
                                        ; when debugging.

log_errors = On ; Log errors into a log file (server-specific log, stderr, or error_log (below))
                ; As stated above, you're strongly advised to use error logging in place of
                ; error displaying on production web sites.
;track_errors = Off ; Store the last error/warning message in $php_errormsg (boolean)
track_errors = On ; *** pour faciliter le débogage
;error_prepend_string = "<font color=ff0000>" ; string to output before an error message
;error_append_string = "</font>" ; string to output after an error message
;error_log = filename ; log errors to specified file
error_log = /tmp/php-errors.log ; *** enregistrent des erreurs sur le serveur
                                ; pour faciliter le travail du formateur ;- )
;error_log = syslog ; log errors to syslog (Event Log on NT, not valid in Windows 95)
warn_plus_overloading = Off ; warn if the + operator is used with strings

```

```

;;;;;;;;;;;;
; Data Handling ;
;;;;;;;;;;;;
; Note - track_vars is ALWAYS enabled as of PHP 4.0.3
variables_order = "EGPCS"      ; This directive describes the order in which PHP registers
                                ; GET, POST, Cookie, Environment and Built-in variables (G, P,
                                ; C, E & S respectively, often referred to as EGPCS or GPC).
                                ; Registration is done from left to right, newer values override
                                ; older values.
register_globals = On          ; Whether or not to register the EGPCS variables as global
                                ; variables. You may want to turn this off if you don't want
                                ; to clutter your scripts' global scope with user data. This makes
                                ; most sense when coupled with track_vars - in which case you can
                                ; access all of the GPC variables through the $HTTP_*_VARS[],
                                ; variables.
                                ; You should do your best to write your scripts so that they do
                                ; not require register_globals to be on; Using form variables
                                ; as globals can easily lead to possible security problems, if
                                ; the code is not very well thought of.
register_argc_argv = On        ; This directive tells PHP whether to declare the argv&argc
                                ; variables (that would contain the GET information). If you
                                ; don't use these variables, you should turn it off for
                                ; increased performance
post_max_size = 8M            ; Maximum size of POST data that PHP will accept.
gpc_order = "GPC"             ; This directive is deprecated. Use variables_order instead.

; Magic quotes
magic_quotes_gpc = On         ; magic quotes for incoming GET/POST/Cookie data
magic_quotes_gpc = Off        ; on n'échappe pas les données entrantes du programme
                                ; en conséquence, il faut les échapper à la main
                                ; lors de certains appels (xxx_exec sur les BDD par ex.)
magic_quotes_runtime = Off    ; magic quotes for runtime-generated data, e.g. data
                                ; from SQL, from exec(), etc.
magic_quotes_sybase = Off     ; Use Sybase-style magic quotes (escape ' with '' instead of \')

; automatically add files before or after any PHP document
auto_prepend_file =
auto_append_file =

; As of 4.0b4, PHP always outputs a character encoding by default in
; the Content-type: header. To disable sending of the charset, simply
; set it to be empty.
; PHP's built-in default is text/html
default_mimetype = "text/html"
default_charset = "iso-8859-1"

;;;;;;;;;;;;
; Paths and Directories ;
;;;;;;;;;;;;
include_path= "./usr/share/php" ; UNIX: "/path1:/path2" Windows: "\path1;\path2"
doc_root =                      ; the root of the php pages, used only if nonempty
user_dir =                       ; the directory under which php opens the script using ~/username,
                                ; used only if nonempty
extension_dir = /usr/lib/php4    ; directory in which the loadable extensions (modules) reside
enable_dl = On                   ; Whether or not to enable the dl() function.
                                ; The dl() function does NOT properly work in multithreaded
                                ; servers, such as IIS or Zeus, and is automatically disabled
                                ; on them.

;;;;;;;;;;;;
; File Uploads ;
;;;;;;;;;;;;
file_uploads = On               ; Whether to allow HTTP file uploads
upload_tmp_dir =                ; temporary directory for HTTP uploaded files
                                ; (will use system default if not specified)
upload_tmp_dir =                ;
upload_tmp_dir = /usr/local/tmp ; ne pas oublier de donner les droits à apache sur ce répertoire
upload_max_filesize = 2M        ; Maximum allowed size for uploaded files

;;;;;;;;;;;;
; Fopen wrappers ;
;;;;;;;;;;;;
allow_url_fopen = On            ; Whether to allow trating URLs like http:... or ftp:... like files

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;
; Dynamic Extensions ;
;;;;;;;;;;;;;;;;;;;;;;;;;
; if you wish to have an extension loaded automaticly, use the
; following syntax: extension=modulename.extension
; Note that it should be the name of the module only, no directory information
; needs to go here. Specify the location of the extension with the extension_dir directive above.

;extension=cpdf.so
;extension=cybercash.so
;extension=db.so
;extension=dbase.so
;extension=domxml.so
;extension=dotnet.so
;extension=exif.so
;extension=fdi.so
;extension=gd.so
;extension=gettext.so
;extension=ifx.so
;extension=imap.so
;extension=interbase.so
;extension=java.so
;extension=ldap.so
;extension=mhash.so
;extension=mssql65.so
;extension=mssql70.so
;extension=mysql.so
;extension=oas.so
;extension=oci8.so
;extension=oracle.so
;extension=pdf.so
;extension=pgsql.so
;extension=sablot.so
;extension=swf.so
;extension=sybase_ct.so
;extension=zlib.so

;;;;;;;;;;;;;;;;;;;;;;;;;
; Module Settings ;
;;;;;;;;;;;;;;;;;;;;;;;;;

[Syslog]
define_syslog_variables= Off; Whether or not to define the various syslog variables,
; e.g. $LOG_PID, $LOG_CRON, etc. Turning it off is a
; good idea performance-wise. In runtime, you can define
; these variables by calling define_syslog_variables()

[mail function]
SMTP = localhost          ; for win32 only
sendmail_from = me@localhost.com ; for win32 only
;sendmail_path =         ; for unix only, may supply arguments as well
;                       ; (default is 'sendmail -t -i')

[Debugger]
debugger.host = localhost
debugger.port = 7869
debugger.enabled = False

[Logging]
; These configuration directives are used by the example logging mechanism.
; See examples/README.logging for more explanation.
;logging.method = db
;logging.directory = /path/to/log/directory

[Java]
;java.class.path = .\php_java.jar
;java.home = c:\jdk
;java.library = c:\jdk\jre\bin\hotspot\jvm.dll
;java.library.path = .\

[SQL]
sql.safe_mode = Off

```

```

[ODBC]
;odbc.default_db = Not yet implemented
;odbc.default_user = Not yet implemented
;odbc.default_pw = Not yet implemented
odbc.allow_persistent = On ; allow or prevent persistent links
odbc.check_persistent = On ; check that a connection is still valid before reuse
odbc.max_persistent = -1 ; maximum number of persistent links. -1 means no limit
odbc.max_links = -1 ; maximum number of links (persistent+non persistent). -1 means no limit
odbc.defaultlrl = 4096 ; Handling of LONG fields. Returns number of bytes to variables, 0 means passthru
odbc.defaultbinmode= 1 ; Handling of binary data. 0 means passthru, 1 return as is, 2 convert to char
; See the documentation on odbc_binmode and odbc_longreadlen for an explanation of
; uodbc.defaultlrl and uodbc.defaultbinmode

[MySQL]
mysql.allow_persistent = On ; allow or prevent persistent link
mysql.max_persistent = -1 ; maximum number of persistent links. -1 means no limit
mysql.max_links = -1 ; maximum number of links (persistent+non persistent).
; -1 means no limit
mysql.default_port = ; default port number for mysql_connect(). If unset,
; mysql_connect() will use the $MYSQL_TCP_PORT, or the mysql-tcp
; entry in /etc/services, or the compile-time defined MYSQL_PORT
; (in that order). Win32 will only look at MYSQL_PORT.
mysql.default_socket = ; default socket name for local MySQL connects. If empty, uses the built-in
; MySQL defaults
mysql.default_host = ; default host for mysql_connect() (doesn't apply in safe mode)
mysql.default_user = ; default user for mysql_connect() (doesn't apply in safe mode)
mysql.default_password = ; default password for mysql_connect() (doesn't apply in safe mode)
; Note that this is generally a *bad* idea to store passwords
; in this file. *Any* user with PHP access can run
; 'echo cfg_get_var("mysql.default_password")' and reveal that
; password! And of course, any users with read access to this
; file will be able to reveal the password as well.

[mSQL]
msql.allow_persistent = On ; allow or prevent persistent link
msql.max_persistent = -1 ; maximum number of persistent links. -1 means no limit
msql.max_links = -1 ; maximum number of links (persistent+non persistent). -1 means no limit

[PostgreSQL]
pgsql.allow_persistent = On ; allow or prevent persistent link
pgsql.max_persistent = -1 ; maximum number of persistent links. -1 means no limit
pgsql.max_links = -1 ; maximum number of links (persistent+non persistent). -1 means no limit

[Sybase]
sybase.allow_persistent = On ; allow or prevent persistent link
sybase.max_persistent = -1 ; maximum number of persistent links. -1 means no limit
sybase.max_links = -1 ; maximum number of links (persistent+non persistent). -1 means no limit
;sybase.interface_file = "/usr/sybase/interfaces"
sybase.min_error_severity = 10 ; minimum error severity to display
sybase.min_message_severity = 10 ; minimum message severity to display
sybase.compatibility_mode = Off ; compatibility mode with old versions of PHP 3.0.
; If on, this will cause PHP to automatically assign types to results
; according to their Sybase type, instead of treating them all as
; strings. This compatibility mode will probably not stay around
; forever, so try applying whatever necessary changes to your code,
; and turn it off.

[Sybase-CT]
sybct.allow_persistent = On ; allow or prevent persistent link
sybct.max_persistent = -1 ; maximum number of persistent links. -1 means no limit
sybct.max_links = -1 ; maximum number of links (persistent+non persistent). -1 means no limit
sybct.min_server_severity = 10 ; minimum server message severity to display
sybct.min_client_severity = 10 ; minimum client message severity to display

[bcmath]
bcmath.scale = 0 ; number of decimal digits for all bcmath functions

[browscap]
;browscap = extra/browscap.ini

```

```

[Informix]
ifx.default_host =           ; default host for ifx_connect() (doesn't apply in safe mode)
ifx.default_user =          ; default user for ifx_connect() (doesn't apply in safe mode)
ifx.default_password =      ; default password for ifx_connect() (doesn't apply in safe mode)
ifx.allow_persistent = On   ; allow or prevent persistent link
ifx.max_persistent = -1     ; maximum number of persistent links. -1 means no limit
ifx.max_links=-1           ; maximum number of links (persistent+non persistent). -1 means no limit
ifx.textasvarchar = 0       ; if set on, select statements return the contents of a text blob instead of it's id
ifx.byteasvarchar = 0      ; if set on, select statements return the contents of a byte blob instead of it's id
ifx.charasvarchar = 0      ; trailing blanks are stripped from fixed-length char columns. May help the life
                           ; of Informix SE users.
ifx.blobinfile = 0         ; if set on, the contents of text&byte blobs are dumped to a file instead of
                           ; keeping them in memory
ifx.nullformat = 0         ; NULL's are returned as empty strings, unless this is set to 1. In that case,
                           ; NULL's are returned as string 'NULL'.

[Session]
session.save_handler        = files    ; handler used to store/retrieve data
session.save_path           = /tmp     ; argument passed to save_handler
                           ; in the case of files, this is the
                           ; path where data files are stored
;session.use_cookies        = 1       ; whether to use cookies
session.use_cookies        = 0       ; *** on évite les cookies ;- )
session.name                = PHPSESSID
                           ; name of the session
                           ; is used as cookie name
session.auto_start         = 0       ; initialize session on request startup
session.cookie_lifetime    = 0       ; lifetime in seconds of cookie
                           ; or if 0, until browser is restarted
session.cookie_path        = /       ; the path the cookie is valid for
session.cookie_domain      =         ; the domain the cookie is valid for
session.serialize_handler  = php     ; handler used to serialize data
                           ; php is the standard serializer of PHP
session.gc_probability     = 1       ; percentual probability that the
                           ; 'garbage collection' process is started
                           ; on every session initialization
session.gc_maxlifetime     = 1440    ; after this number of seconds, stored
                           ; data will be seen as 'garbage' and
                           ; cleaned up by the gc process
session.referer_check      =         ; check HTTP Referer to invalidate
                           ; externally stored URLs containing ids
session.entropy_length     = 0       ; how many bytes to read from the file
session.entropy_file       =         ; specified here to create the session id
;session.entropy_length     = 16
;session.entropy_file       = /dev/urandom
session.cache_limiter      = nocache ; set to {nocache,private,public} to
                           ; determine HTTP caching aspects
session.cache_expire       = 180    ; document expires after n minutes
session.use_trans_sid      = 1       ; use transient sid support if enabled
                           ; by compiling with --enable-trans-sid
url_rewriter.tags         = "a=href,area=href,frame=src,input=src,form=fakeentry"

[MSSQL]
mssql.allow_persistent = On           ; allow or prevent persistent link
mssql.max_persistent = -1            ; maximum number of persistent links. -1 means no limit
mssql.max_links = -1                ; maximum number of links (persistent+non persistent). -1 means no limit
mssql.min_error_severity = 10        ; minimum error severity to display
mssql.min_message_severity = 10      ; minimum message severity to display
mssql.compatibility_mode = Off        ; compatibility mode with old versions of PHP 3.0.
;mssql.textlimit = 4096              ; valid range 0 - 2147483647 default = 4096
;mssql.textsize = 4096               ; valid range 0 - 2147483647 default = 4096
;mssql.batchsize = 0                 ; limits the number of records in each batch.
                                   ; 0 = all records in one batch.

[Assertion]
;assert.active = On                 ; assert(expr); active by default
;assert.warning = On                ; issue a PHP warning for each failed assertion.
;assert.bail = Off                  ; don't bail out by default.
;assert.callback = 0                ; user-function to be called if an assertion fails.
;assert.quiet_eval = 0              ; eval the expression with current error_reporting().
                                   ; set to true if you want error_reporting(0) around the eval().

```

```

[Ingres II]
ingres.allow_persistent = On      ; allow or prevent persistent link
ingres.max_persistent = -1       ; maximum number of persistent links. (-1 means no limit)
ingres.max_links = -1           ; maximum number of links, including persistents (-1 means no limit)
ingres.default_database =        ; default database (format : [node_id:]dbname[/srv_class]
ingres.default_user =            ; default user
ingres.default_password =        ; default password

[Verisign Payflow Pro]
pfpro.defaulthost = "test.signio.com" ; default Signio server
pfpro.defaultport = 443              ; default port to connect to
pfpro.defaulttimeout = 30            ; default timeout in seconds

; pfpro.proxyaddress =            ; default proxy IP address (if required)
; pfpro.proxyport =              ; default proxy port
; pfpro.proxylogon =             ; default proxy logon
; pfpro.proxypassword =          ; default proxy password

[SocketS]
sockets.use_system_read = On        ; Use the system read() function instead of
; the php_read() wrapper.

; Local Variables:
; tab-width: 4
; End:

```

# Notes

