

Web dynamique

Techniques, outils, applications

(Partie B)

SOMMAIRE

1. Introduction (5mn)
2. la chaine de développement Web (5mn)
3. Scripts Serveur (30mn)
4. PHP précis et concis (40mn)

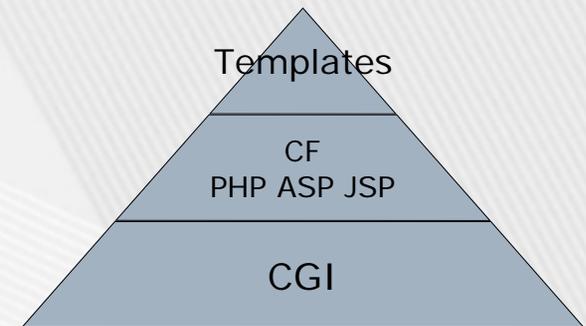
la chaine de développement Web

Particularités et difficultés

Problématiques et réponses

- Rappel
 - Le Web ne s'est vraiment développé qu'à partir du moment où il a proposé des contenus dynamiques
 - Un contenu dynamique est nécessairement connecté à une base de données (dans son sens large)
- les problématiques associées au web dynamique
 - Le maintien de sessions transactionnelles
 - la performance
 - la sécurité
 - la réutilisabilité du code
 - Séparation logique / Cosmétique
- A retenir
 - CGI est le dénominateur commun à tous les serveurs
 - Les API serveurs ont permis d'améliorer les performances de CGI
 - Les scripts serveurs s'appuyant sur CGI proposent des facilités (sessions, mixage codes tiers serveur/client)
 - Les Templates permettent la séparation du contenu et de la présentation

	CGI	API Serveurs	Scripts serveurs	Templates
sessions			X	
performance		X		
Sécurité			X	
réutilisabilité			X	X
Séparation logique / Cosmétique				X
Disponibilité serveur/OS	X		X	X

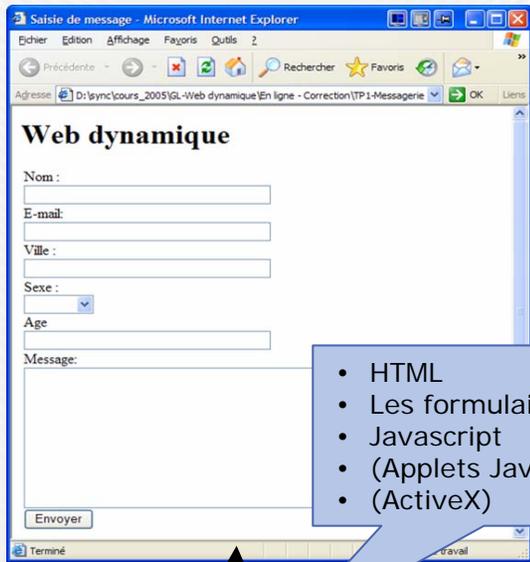


la chaine de développement Web

- Une application Web typique sera divisée
 - Une partie côté navigateur
 - Une partie côté serveur
- Particularité:
 - La totalité des fichiers de l'application se trouvent sur le serveur
 - Mais les contenus de ces fichiers ne seront pas exécutés au même moment (HTML, PHP, Javascript)
 - Ni par le même tiers
- Au cours du dev, une réflexion sur la position (le tiers) d'une fonctionnalité doit être menée
 - Exemples:
 - la validation d'un formulaire
 - Une calculatrice euro
 - un script qui affiche l'heure ?

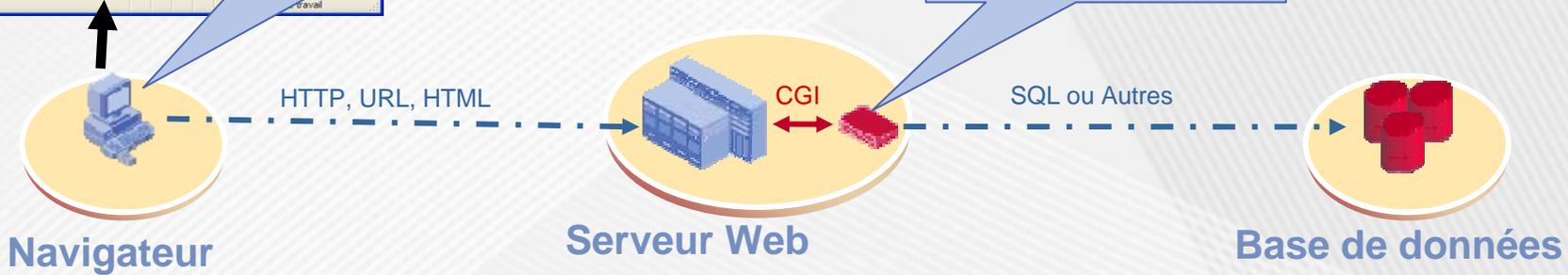
la chaine de développement Web

Seuls les scripts serveurs offrent une vue complète (code tier client/tiers serveur) dans un même fichier, c'est ce qui fait leurs succès



- HTML
- Les formulaires HTML
- Javascript
- (Applets Java)
- (ActiveX)

- Programmes CGI
 - Perl et C
- Les scripts serveur
 - ASP
 - PHP
 - ColdFusion
 - (JSP)
 - (Servlets java)



la chaine de développement Web

- Exemple de la validation d'un formulaire
 - par exemple vérifier qu'un numéro de tel ou un email a un format correct)
- Réflexion sur la position (le tier) à mener
 - Vérifier coté client?
 - utiliser JavaScript
 - Avantage : rapidité, pas d'accès réseau
 - Inconvénient : javascript peut être désactivé sur le navigateur
 - Vérifier coté serveur?
 - utiliser CGI/PHP/ASP/JSP/CF sur le serveur
 - Avantage : impossible de désactiver la vérification par le client
 - Inconvénient : accès serveur (aller-retour), pas de vérification pas à pas sur le client
 - Vérifier des deux cotés?
 - Avantage : rapidité, vérification pas à pas sur le client, robustesse (vérification au moins sur le serveur)
 - Inconvénient : double codage

Langages de scripts Serveur

Tour d'horizon: PHP,ASP,CF,JSP
Points communs et différences

Cold Fusion

- ❑ Bi-plateforme
- ❑ Utilise CGI ou ISAPI selon la config
- ❑ Définis un langage sur le modèle de HTML (à base de balises) nommé CFML
- ❑ Fichier avec extension « .cfm »
- ❑ 4 pseudo-balises d'accès SQL :
 - **CFquery** : requête SQL
 - **CFinsert** : insérer dans la base de données
 - **CFupdate** : mettre à jour dans la base
 - **Cfoutput** : pour afficher les résultats des requêtes

Cold Fusion : DBML

```
<CFQUERY NAME= "ListeContact" DATASOURCE="SOURCEODBC" >
SELECT NUM,Nom,Prenom,Telephone FROM Contact
</CFQUERY>
```

```
<HTML>
```

```
<BODY>
```

```
<CFOUTPUT>
```

```
<H2>#Form.HelloText#</H2>
```

```
</CFOUTPUT>
```

```
<CFOUTPUT QUERY="ListeContact">
```

```
(#NUM#) #Nom#<BR> #Prenom# #Telephone#<BR>
```

```
</CFOUTPUT>
```

```
</BODY></HTML>
```

- ❑ « #champs# » sera remplacé par la valeur issue de la requête
- ❑ Gestion de formulaire: « #Form.champs# » avec champs étant le nom de la balise input

Cold Fusion

- Gestion des champs obligatoires
 - `<INPUT type="text" NAME="SSN" VALUE="" >`
 - `<INPUT type="hidden" NAME="SSN_required" VALUE="Vous devez entrer un numéro de sécurité sociale" >`

- Support du branchement conditionnel
 - CFif et Cfelse

- CFML contient d'autres balises
 - par exemple envoi de Email via balise CFmail

Active Server Page (ASP)

- Vision prémachée du développement
 - via les objets Application, Session, Request, Response
 - Fichiers d'extension « asp »
 - Code serveur délimité par pseudo-balises <%...%>
- Objet « Request »
 - La récupération des données des formulaires (« Request.Form ».)
 - informations sur l'utilisateur
 - Lire des Cookies
- Objet « Response »
 - Page renvoyée au client Web
 - Création des Cookies

Active Server Page (ASP)

```
<%
Set Param= Request.Form
Parametrevide = 1
For each p in Param
    If strcmp(param(p), "ON",1)=0 then
        Parametrevide = 0
        Exit for
    End if
Next
If Parametrevide = 1 then
    Response.Redirect « home.htm »
End if
%>
<html><head><title>Essai</title></head></html>
<body>
du texte avant.<br>
<% if request.form("checkbox1")="ON" then %>
    <i>ceci ne sera affiché que si on à coché checkbox 1</i>
<% end if %>
du texte après.<br>
</body>
```

Active Server Page (ASP)

□ Avantages:

- Bonne formalisation du dialogue client/serveur et des transactions via les objets Application, Session, Request, Response
- générations automatiques avec Visual Interdev
- Programmation facile (Vbscript ou javascript)

□ Inconvénients:

- Windows uniquement
- bases ODBC uniquement
- serveurs IIS / NT uniquement
- Nombreux problèmes de sécurités liés à IIS/ASP
- Fonctionnalités avancées payantes

Java Server Pages (JSP)

- Réponse proposée par SUN à la complexité des servlets et à la popularité des asp
- Propose les mêmes principes que les asp
 - Intégration code client et serveur dans le même fichier
 - Mêmes balises `<%...%>`
 - Mêmes objets (session, application, request, response)
- avantage tout de même:
 - Les JSP deviennent au moment de l'accès des servlets: compilés VRAIMENT donc rapides
 - On dispose de toute la richesse du JDK en matière de fonctionnalités (images, math,...) gratuitement

Java Server Pages (JSP)

```
<HTML><HEAD><TITLE>Exemple de JavaServerPage</TITLE></HEAD>
<BODY>
<CENTER>Exemple de JavaServerPage</CENTER>
<P>Contenu dynamique créé en utilisant différents mecanismes JSP : <BR>

<B>Expression: </B>Votre hostname: <%= request.getRemoteHost() %><BR>
<B>Scriptlet : </B><%= out.println(« data HTTP: " + request.getQueryString()); %><BR>
<B>Declaration (+ expression) : </B>
<%! private int accessCount = 0; %>Nombre d'accès à la page : <%= ++accessCount %>
<B>Directive (+ expression) : </B>
<%@ page import = "java.util.*" %>Date courante: <%= new Date() %><BR>

<P>JSP c'est super <%= if (Math.random() < 0.5) { %>
<B>simple</B> !
<%= } else { %>
<B>compliqué</B> !
<%= } %>
</BODY>
</HTML>
```

PHP

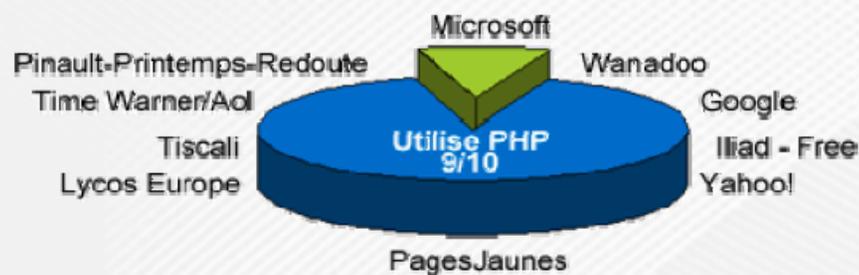
- ❑ PHP est un langage de scripts exécutable sur le serveur
- ❑ Le principe est le même que ASP ou JSP : imbriquer le code PHP dans la page HTML (cad mixer le code client et serveur)

- ❑ On le voit dans le détail tout de suite!
- ❑ Pourquoi?
 - PHP est déployé sur 53% des serveurs dans le monde soit 1 site sur 3
 - 46% des sites web français

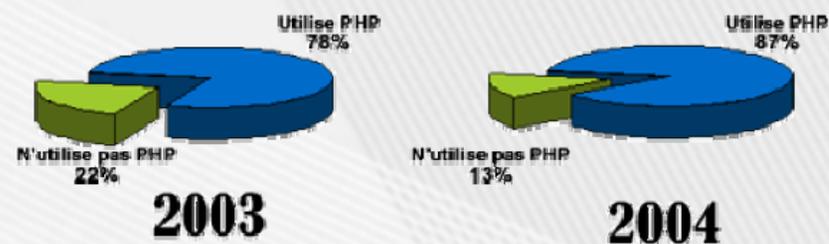
Pourquoi PHP

- **90% des sites français les plus fréquentés**
- Parmi les compagnies totalisant le plus de visites (Microsoft, Wanadoo, Google, Iliad – Free, Yahoo!, etc.) 90% d'entre elles utilisent PHP pour des fonctions spécifiques (chat, petites annonces, etc.) ou la totalité de leurs services. Source : Nielsen/NetRatings & AFUP
- **87% des entreprises du CAC40 et 95 % des 20 premières entreprises françaises**
- AFP, Alcatel, Alstom, Altran, Atos Origin, Aventis, Axa, Bayer, BHV, BNP Paribas, Bouygues, Carrefour, Casino, CCF, Cegetel, Cetelem, Club Med, Coca Cola, Crédit Agricole, Crédit Lyonnais, Danone, Dassault, EADS, EDF-GDF, Europe 2, France 2, France Telecom, Gemplus, Givenchy, Le Figaro, Le Monde, Libération, M6, Michelin, Nokia, Novartis, Pechiney, Présidence de la République, PSA, RATP, Sagem, Sanofi, SNCF, Snecma, Société Générale, Suez, TF1, Ubi Soft, Wanadoo, etc. Source : Place de la plate-forme PHP dans l'économie française – Globalis Media Systems

Les plus gros trafics et PHP



Les entreprises du CAC 40 et PHP



Bases de données utilisées

□ ASP

- MS Jet Engine (accès aux fichiers MS Access)
- MS SQL Server
- Oracle (ODBC)

□ ColdFusion

- Oracle
- Informix
- Sybase

□ PHP

- MySQL
- Postgres
- Oracle
- mSQL

PHP

Précis et concis



PHP

Mini-plan:

- Structure d'un source php
- Variables et types
- Structures de contrôle
- fonctions
- interaction avec HTML

PHP

- ❑ Les pages contenant du code PHP ont l'extension « .php »
- ❑ PHP est disponible :
 - comme module sur Apache et Microsoft IIS (isapi)
 - également en exécutable CGI
 - sur tous les serveurs Web et sur tous les OS
- ❑ Avantages
 - Simple : variables sans type
 - Grand nombre de module externes (Images, Zip, pdf, graphiques...)
 - Gratuit (GNU OpenSource)
- ❑ Concurrent de Microsoft ASP
- ❑ Plus de 150.000 sites Web avec PHP en Juin 1998

- ❑ Historique:
 - 1995 : première version (sous le nom de Personal Home Pages) par Rasmus Lerdorf
 - 1997 : PHP3.
Renommé "PHP: Hypertext Preprocessor" (un acronyme récursif)
 - 1999 : PHP4
 - 2005: PHP5 (attention, très différent de la version 4)

PHP

- ❑ PHP est un langage de scripts exécutable sur le serveur
- ❑ Le principe est le même que ASP ou JSP : imbriquer le code PHP dans la page HTML (cad mixer le code client et serveur)
- ❑ Exemple:

```
<HTML>
<HEAD>
<TITLE>Test HelloWorl en PHP</TITLE>
</HEAD>
<BODY>
<? echo "Hello world !"; ?>
</BODY>
</HTML>
```

- ❑ La syntaxe du langage est proche de C et Perl
 - instructions terminées par un point-virgule ;
 - Bloc d'instructions délimité par des accolades {}
 - Commentaires comme C et C++

PHP : script serveur

- Ce que le serveur voit et exécute:
 - `<HTML>`
 - `<HEAD>`
 - `<TITLE>Test HelloWorl en PHP</TITLE>`
 - `</HEAD>`
 - `<BODY>`
 - `<? echo "Hello world !"; ?>`
 - `</BODY>`
 - `</HTML>`

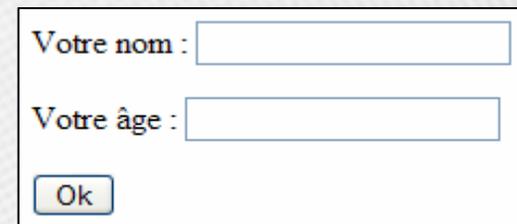
- Ce que le navigateur reçoit:
 - `<HTML>`
 - `<HEAD>`
 - `<TITLE>Test HelloWorl en PHP</TITLE>`
 - `</HEAD>`
 - `<BODY> Hello world !`
 - `</BODY>`
 - `</HTML>`

- On ne peut donc pas voir ou copier le source PHP depuis le navigateur

PHP et Formulaires HTML

- Champs des formulaires directement dans des variables :

```
<FORM ACTION="essai.php" METHOD="POST">  
  Votre nom : <INPUT TYPE=TEXT NAME=nom>  
  Votre âge : <INPUT TYPE=TEXT NAME=age>  
  <INPUT TYPE=SUBMIT VALUE="Ok">  
</FORM>
```



- Dans `essai.php`, on dispose de variables pré-crées:

- `$nom`
- `$age`

- On peut écrire en réponse dans `essai.php`:

```
...  
Bonjour <? echo $nom; ?>  
...
```

PHP : variables

- ❑ Variables marquées par un \$
- ❑ Le nom est sensible à la casse (ie : \$x != \$X)
- ❑ Les variables n'ont pas à être déclarées

- ❑ concaténation de chaînes avec '.' (un point) : le reste est comme C/C++ :
- ❑ Exemple:

```
<?php
    $salut = "Hello ";
    $num = 3 + 2;
    $num++;
    echo "$salut vous $num !";
?>
```

- ❑ Affiche :
Hello vous 6!

PHP : types

- Les variables reçoivent leur type par des expressions affectés
 - Opérateur arithmétiques, de comparaison, et booléens identiques à C
- Type connus de PHP:
 - booléen TRUE FALSE (i.e. 0, "", "0")
 - entier
 - flottant
 - chaîne de caractères (entre " ou ')
 - Tableau

PHP : types

- une variable peut changer de type à tout moment.
 - Les conversions de type se font automatiquement.
- Exemple:
 - `$chaine = "1+1=" . (1+1)`
→ `$chaine` vaut : `"1+1=2 "`
- Exemple:

```
$a = 2;  
$b = 6;  
$b = $a . $b;  
$b /= 2;  
→ print "$b\n" affichera 13 !
```

Les chaînes de caractères

- Comme avec les scripts Unix ou Perl, les 2 types de guillemets autorise l'interprétation du contenu
 - Simple quote : chaine brute, aucune substitution
 - Double quote : interprete les caracteres \$, \, et '
- Exemple:
 - echo "bonjour \$NOM"
 - → \$NOM est remplacé par sa valeur
 - \$A = "mot";
 - \$B = "un autre \$A";
 - \$C = 'un autre \$A';
 - → \$A est remplacé par sa valeur dans \$B
- Comme en C
 - '\n' est le retour chariot
 - le backslash \ est le caractère d'échappement
- Concaténation de chaînes de caractères.
 - " Hello ". " World!";
 - \$A = \$B . \$C . " mickey";
 - Plus simple: \$A = "\$B\$C mickey";

Fonctions de chaînes

- ❑ `strchr()` ou `strstr()` ou `strpos()`: Trouve la première occurrence d'un caractère dans une chaîne
- ❑ `strlen()`: retourne la taille de la chaîne.
- ❑ `Chop()`: supprimer les retours chariots finaux
- ❑ `ltrim()` `rtrim()` `trim()` enlève les blancs de début ou/et de fin
- ❑ `strip_tags()`: Supprime les balises HTML et PHP d'une chaîne
- ❑ `htmlspecialchars()`: convertit les caractères spéciaux en leur équivalent HTML (accents par exemple) + les caractères qui ont des significations spéciales en HTML sont remplacés par des codes HTML
 - `'&'` devient `'&'`
 - `"` devient `'"'`
 - `'` devient `'''`
 - `'<'` devient `'<'`
 - `'>'` devient `'>'`

Fonctions de chaines: substr

- Substr() Retourne un segment de chaîne
 - string **substr** (string string, int start [, int length])

- Exemples:
 - \$rest = substr("abcdef", 1); // retourne "bcdef"
 - \$rest = substr("abcdef", 1, 3); // retourne "bcd"
 - \$rest = substr("abcdef", 0, 4); // retourne "abcd"
 - \$rest = substr("abcdef", 0, 8); // retourne "abcdef"

- Accès aux caractères via les accolades
 - \$mickey = 'abcdef';
 - echo \$mickey{0}; // retourne a
 - echo \$mickey{3}; // retourne d

- *start* peut etre négatif :
 - \$rest = substr("abcdef", -1); // retourne "f"
 - \$rest = substr("abcdef", -2); // retourne "ef"
 - \$rest = substr("abcdef", -3, 1); // retourne "d"

PHP : types tables

□ Tableaux et listes associatives :

- `$animal[0] = 'chien';`
`$animal[1] = 'chat';`

Et

- `$animal['poisson'] = 'eau';`
`$animal['chat'] = 'terre';`

□ Une liste associative est un tableau à indexation littérale

□ Création « compacte »:

- `$tab = array(1,2,3);`
- `$tab = array("chien", "chat", "oiseau", "poisson");`
- `$liste = array('couleur' => 'orange',`
■ `'forme' => 'rond',`
■ `'nom' => abricot');`

PHP : types tables

- Exemple:
 - \$fruits = array (
 - "fruits" => array ("a"=>"orange", "b"=>"banane", "c"=>"pomme"),
 - "nombres" => array (1, 2, 3, 4, 5, 6),
 - "trous" => array ("premier", "second", "troisieme"));

- Exemple:
 - \$array = array(1, 1, 1, 1, 1, 8=>1, 4=>33, 19, 3=>13);

- Print_r(): Affiche des informations de tables
 - print_r(\$array); affichera :
 - Array (
 - [0] => 1
 - [1] => 1
 - [2] => 1
 - [3] => 13
 - [4] => 33
 - [8] => 1
 - [9] => 19)

- Count() ou sizeof(): nombre d'éléments d'un tableau

PHP: Structures de contrôle

- if..then..**else**, do..while, for, while, switch/case : comme C/C++

```
for($i = 0; $i < 10; $i++) {  
    echo $i;  
}
```

- Foreach:

```
<?php  
$tab=array("a","b","c");  
foreach($tab as $val)  
{  
    echo "valeur:$val\n";  
}
```

Equivaut à:

```
for($i=0;$i<sizeof($tab);$i++)  
{  
    echo("valeur:$tab[$i]\n");  
}  
?>
```

PHP : Fonctions (1)

□ Fonctions standards

- `function func($arg1, $arg2,...) {`
- `...instructions...`
- `return $valeur_retour; //optionnel`
- `}`

□ Exemple:

- `function double($i) {`
- `return $i * 2;`
- `}`

PHP : Fonctions (2)

- passage de paramètres par référence:

```
// Passage de paramètres par référence
```

```
function essai(&$arg) {  
    $arg = $arg. " supplement ";  
}
```

```
$str = "debut";
```

```
essai($str);
```

```
Echo $str; → « debut supplement »
```

- → Essai contient la concaténation des 2 chaînes

PHP : Fonctions (3)

- ❑ Possibilité de définir des valeurs par défaut pour les arguments
- ❑ Exemple:
 - `function double($i=1) {`
 - `return $i * 2;`
 - `}`
- ❑ Exemple:
 - `function manger($s="caviar") { }`
 - ❑ Appel: `manger();`
 - ❑ Appel: `manger("abricot");`

PHP : Fonctions (4)

- ❑ Alternative au passage d'argument par référence: global
- ❑ Modifie la portée d'une variable locale à la fonction

- ❑ Piège classique en débutant PHP:

```
$a = 1; /* portée globale */  
function test() {  
    echo $a; /* portée locale */  
}  
test();
```

- ❑ Exemple:

```
$a = 1;  
$b = 2;  
function somme() {  
    global $a, $b;  
    $b = $a + $b;  
}  
somme();  
echo $b;
```

PHP : Les fichiers

- ❑ Construction classique mais autorise aussi des URLs:

```
$fp = fopen("C:\\php\\messages.txt", "r");
```

```
$fp = fopen("http://www.php.net/", "r");
```

```
$fp = fopen("ftp://user:password@example.com/toto.txt", "w");
```

...puis `fclose($fp)` pour fermer le fichier

- ❑ modes d'ouverture :

- 'r' - Ouvre en lecture seule

- 'r+' - Ouvre en lecture et écriture

- 'w' - Ouvre en écriture seule

- 'w+' - Ouvre en lecture et écriture

- 'a' - Ouvre en écriture seule en ajout

- 'a+' - Ouvre en lecture et écriture en ajout

- ❑ Vérifier si un fichier existe.

- `if (!file_exists($fichier))...`

PHP : Les fichiers

- Lire/écrire dans un fichier texte :
 - Ouvrir: `$file = fopen($fichier, "w+");`
 - Lire: `$nom=fgets($file); //jusqu'au retour chariot`
 - Ecrire: `fputs($file,"$nom\n"); // ecrit $nom dans le fichier`
 - `feof($file) →` renvoie un boolean indiquant si la fin du fichier est atteinte (à utiliser avec la lecture)
 - Fermer le fichier: `fclose($file)`

- exemple:

```
$in=fopen("data.txt","r");
$out=fopen("out.txt","w");
$i=1;
while (!feof($in)) {
    $ligne = fgets($in);
    fputs($out,$i++ . "\t".$ligne);
}
fclose ($in);
fclose ($out);
```

PHP: Inclusion de fichiers

- ordres include, require :
 - **include('fonctions.php');**
 - L'extension est libre!

- Commandes include et require :
 - require('entete.inc');
 - include('detail.inc');
 - Ré-inclusion du fichier à chaque appel de la fonction
 - require provoque une erreur fatale si le fichier n'est pas trouvé
 - include provoque un warning

- Commandes d'inclusions unique:
 - require_once('entete.inc');
 - include_once('detail.inc');
 - Contenu du fichier remplacé au premier appel de la fonction et plus par la suite

PHP: fonctions API (1)

- ❑ API très riche. Propose un nombre incalculable de fonctions (2700 fonctions aux dernières nouvelles)
- ❑ Il est tout de même nécessaire d'activer ces modules PHP correspondants sur le serveur
- ❑ Groupes de fonctions de l'API de PHP:
 - manipulation de chaînes de caractères : addslashes, chop, crypt, explode, htmlspecialchars, htmlentities, QuoteMeta, rawurldecode, print, printf, sprintf
→ 50 fonctions
 - manipulation de variables : gettype, is_array, is_double, empty, doubleval, etc...
 - Manipulation de tableaux et listes : array, array_walk, key, list, key, ksort, asort...
 - manipulation de dates, heures, calendriers : date, checkdate, gmdate, getdate, time, etc...
 - Fonctions mathématiques : sin, cos, sqrt, number_format

PHP: API (2)

- Principaux groupes de fonctions de l'API de PHP:
 - Fonctions de manipulation de fichiers et répertoires : `dir`, `chdir`, `opendir`, etc... `copy`, `rename`, `chmod`, `chgrp`, `fopen`, `fclose`, `fgets`..
 - Fonctions relatives aux Bases de Données. Classées par type de BD supportée : `dBase`, `dbm`, `MiniSQL`, `MySQL`, `Oracle`... plus `ODBC`
 - Fonctions HTTP : `setcookie`, `header`
 - Fonctions de génération d'images (`gd`) à la volée pouvant contenir du texte: `ImageCreate`, `ImageColorTransparent`...
 - Fonctions mail : `mail` pour envoyer un mail plus gestion complète du protocole `IMAP`, API de 50 fonctions...
 - Fonctions de support format PDF : génération de fichiers PDF à la volée: `PDF_open`, `PDF_close`, etc..
 - Fonctions de compression/décompression : support pour `gzip` : `gzopen`, `gzread`, etc...

Quelques liens

- ❑ Apache: <http://www.apache.org/>
- ❑ ASP: <http://msdn.microsoft.com> puis rechercher "Active Server Pages"
- ❑ ColFusion:
<http://www.macromedia.com/software/coldfusion/>
- ❑ MySQL: <http://www.mysql.com/>
- ❑ PHP: <http://www.php.net/>
 - En français
 - ❑ <http://www.phpindex.com/>
 - ❑ <http://www.phpdebutant.com> (forum)
 - ❑ <http://www.afup.org/>