

Programmation Web avec *PHP*

bases du langage

cours de M. Clergue
assaisonné par S. Verel

Objectifs de la suite du cours

Maîtriser les techniques à la base de la création de sites web dynamiques côté serveur :

- génération automatique de pages HTML
- récupération des données utilisateur
- création et interrogation des bases de données

I. Sites web dynamiques : principes

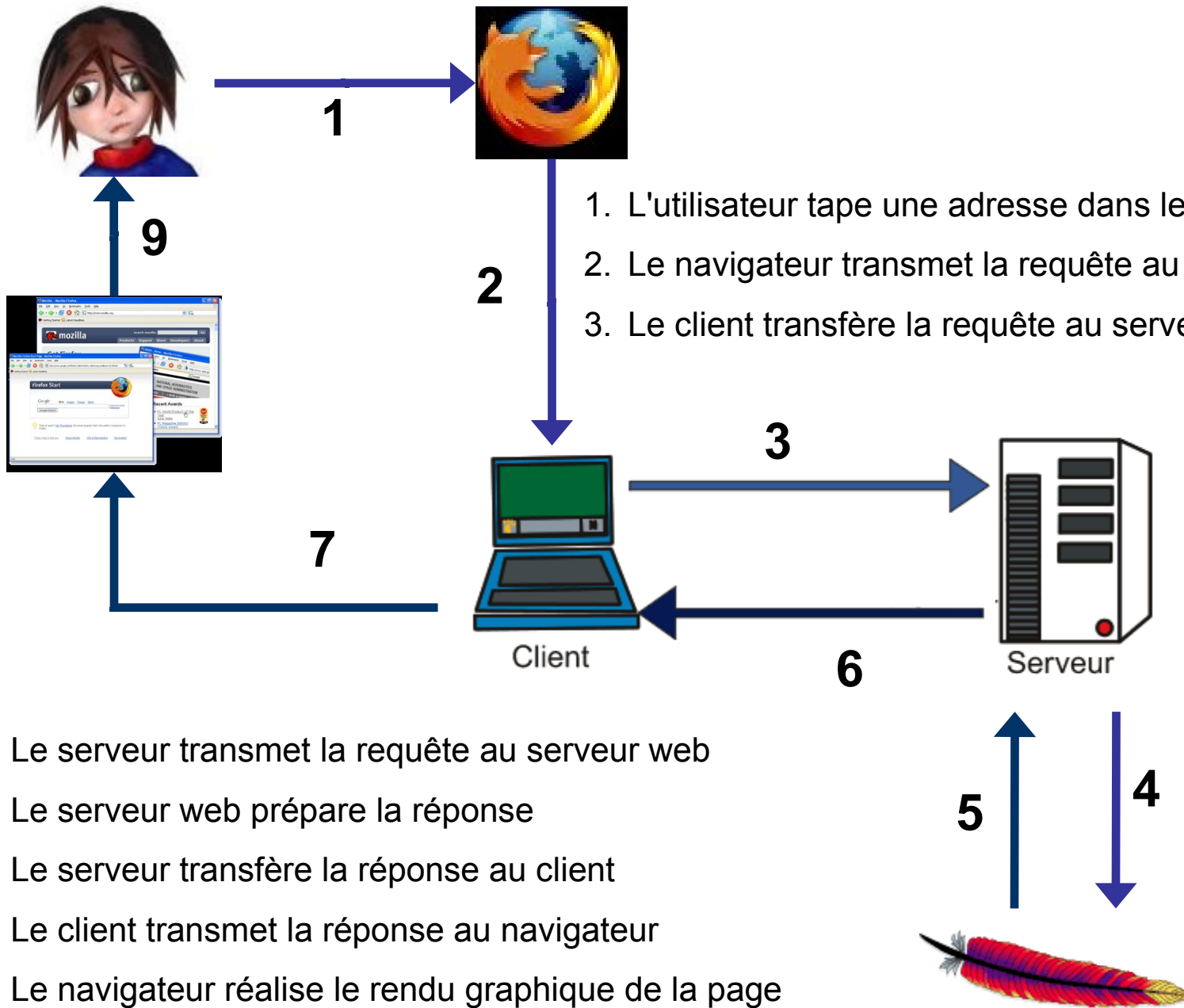
- Rappels sur le web
- Principes de fonctionnement
- Premiers pas en Php

Site Web ?

- **site web** : ensemble de *pages web hyperliées* entre elles, appartenant à un même *serveur web* et mises en ligne à une *adresse Web*.
- **page web** : ressource du World Wide Web conçue pour être consultée par des visiteurs à l'aide d'un navigateur Web.
- **hyperlien** ou **lien hypertexte** : référence dans un système hypertexte permettant de passer automatiquement d'un document consulté à un document lié.
- **adresse Web** ou **URI (URL)** : référence universelle d'une ressource sur le web (et plus encore), de la forme :
protocole : \ \ **serveur** : **port** \ **chemin** # **ancree** ? **params**
- **serveur web** : ensemble machine et logiciel hébergeant des pages web

World Wide Web

- Quels sont les trois principes du WWW ?



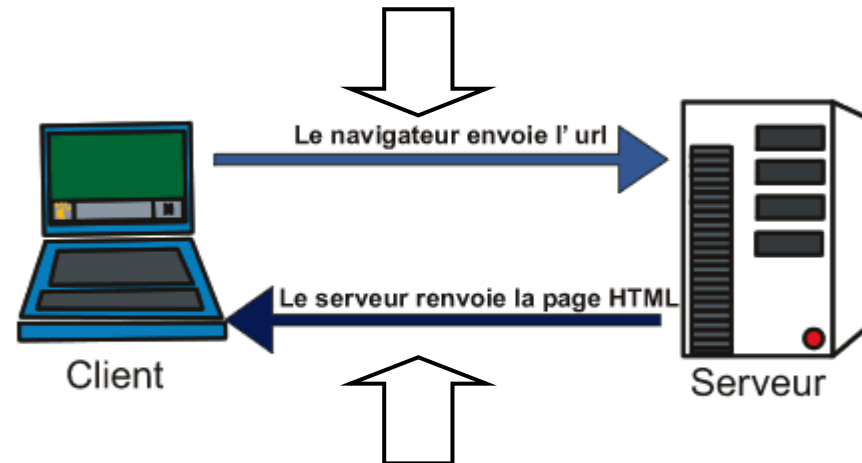
4. Le serveur transmet la requête au serveur web
5. Le serveur web prépare la réponse
6. Le serveur transfère la réponse au client
7. Le client transmet la réponse au navigateur
8. Le navigateur réalise le rendu graphique de la page

```
GET /exemple.html HTTP/1.0
```

```
Accept : text/html
```

```
If-Modified-Since : Saturday, 15-January-2000 14:37:11 GMT
```

```
User-Agent : Mozilla/4.0 (compatible; MSIE 5.0; Windows 95)
```



```
HTTP/1.0 200 OK
```

```
Date : Sat, 15 Jan 2000 14:37:12 GMT
```

```
Server : Apache/2.0
```

```
Content-Type : text/HTML
```

```
Content-Length : 1245
```

```
Last-Modified : Fri, 14 Jan 2000 08:25:13 GMT
```

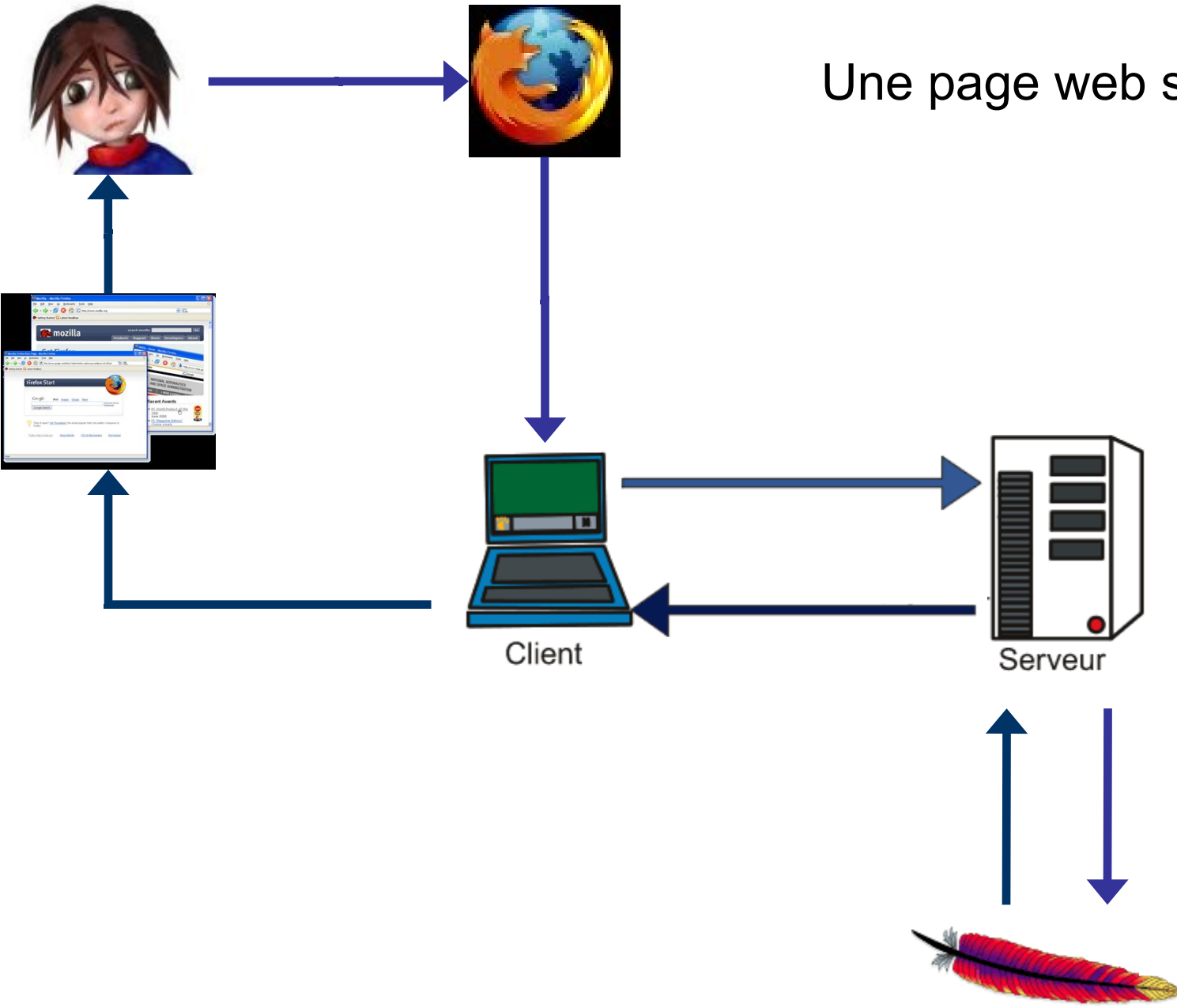
```
<html>
```

```
<head>
```

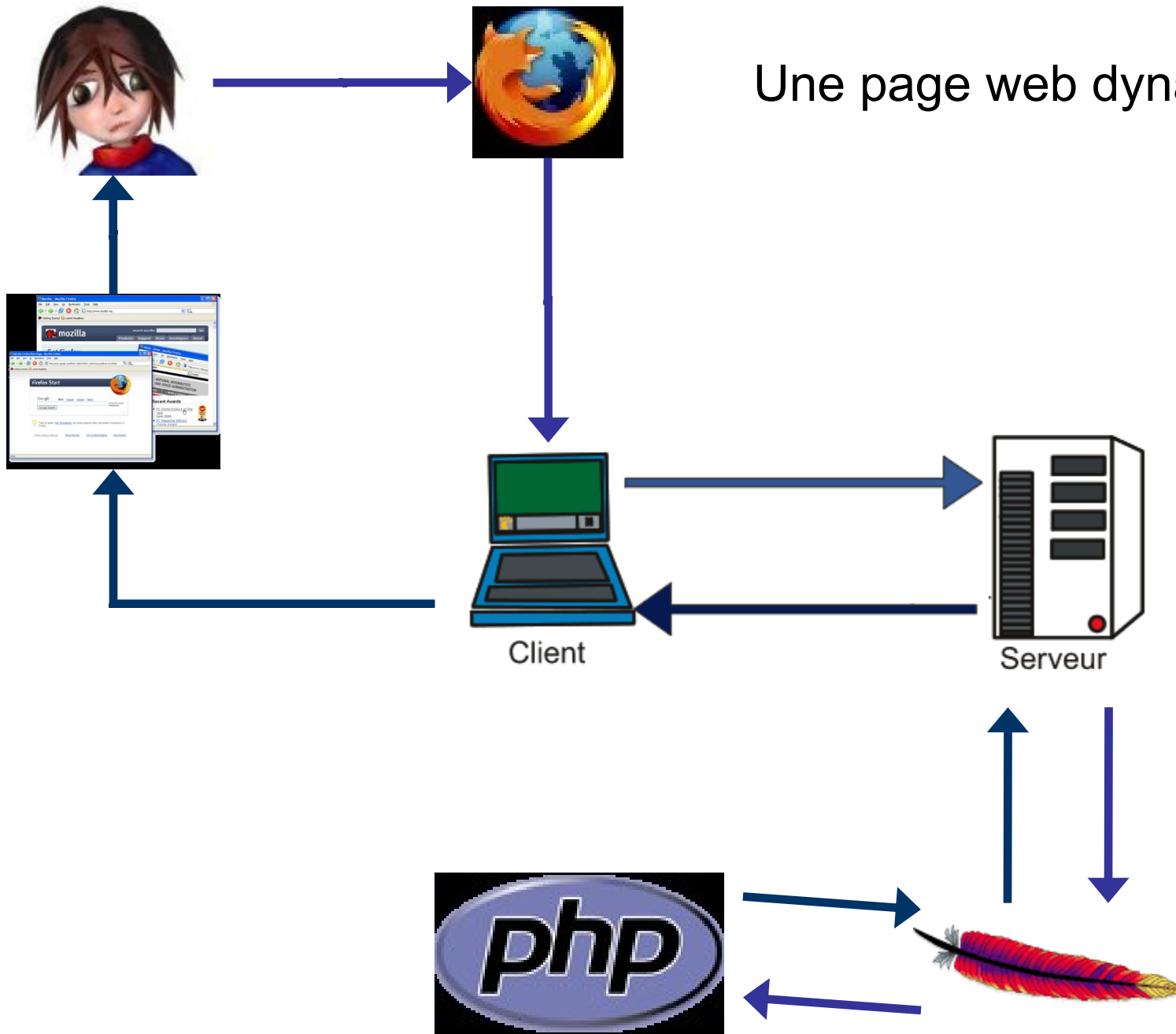
Statique ? Dynamique ?

- Une page web est statique quand son contenu et son apparence sont définis une bonne fois pour toute lors de son écriture.
- Une page web est dite dynamique lorsque son contenu (ou son apparence) dépend d'informations non connues au moment de l'écriture de la page.
- Les sources d'informations peuvent être :
 - Le système lui même
 - Une base de données
 - Le visiteur/utilisateur de la page (du site)

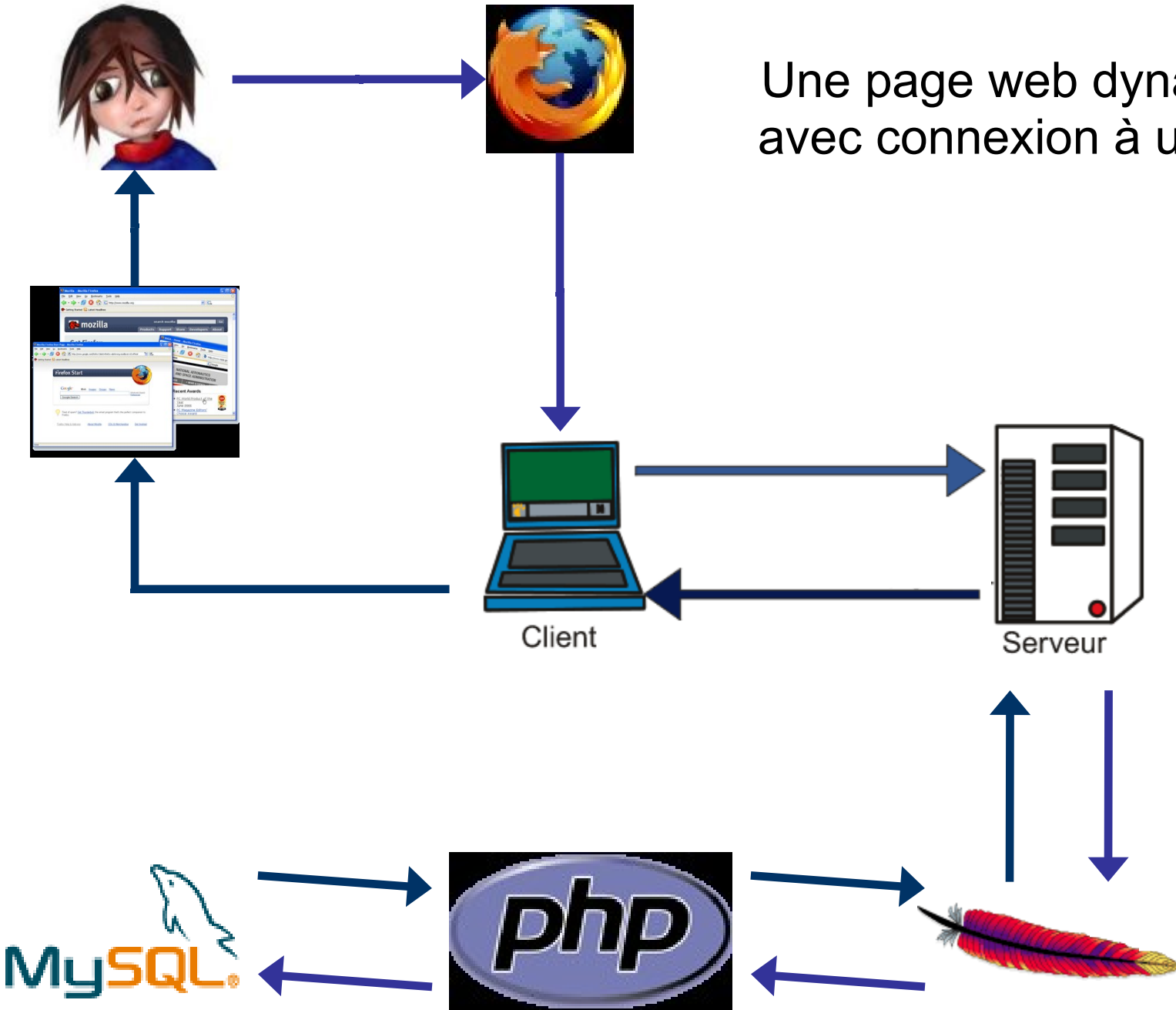
Une page web statique



Une page web dynamique



Une page web dynamique
avec connexion à une BD



Systemes de page web dynamique

- L'origine : Les scripts CGI
 - *Common Gateway Interface*

Langages de script :

- ASP (*Active Server Pages*) & ASP.NET
- PHP
- Servlet Java

Présentation de PHP

PHP : Hypertext Preprocessor (Personal Home Pages)

Historique :

1994 : Rasmus Lerdorf : usage personnel

1995 : 1^o version : "Personal Sommaire Page Tools"

1995 : PHP/FI Version 2

1997 : Suraski & Gutmans : PHP 3

2000 : PHP 4

2005 : PHP 5

Fonctionnalités :

Génération de page web à la volée

Traitement des formulaires clients

Connexion à des bases de donnée

Langage simple, souple, puissant

Principe de fonctionnement

Écriture de scripts PHP :

Inclusion du code PHP à l'intérieur du code HTML

Exécution de scripts PHP :

Exécution du code directement par le serveur

Génération d'une page HTML envoyée au client

Entrées d'un script PHP :

Valeurs de formulaire

Données extraites d'une base de données

Fichiers et données système

Sortie d'un script PHP :

Du code HTML (... mais pas seulement ...)

exemple.php

Du code HTML ...

```
<html>
<head>
  <title>Untitled Document</title>
  <meta http-equiv="Content-Type"
    content="text/html; charset=iso-8859-1">
</head>
<body>
  <?php
    echo "Hello James!";
  ?>
</body>
</html>
```

Du code ?

Une nouvelle balise ...

Et maintenant ?

Attention !

- Il faut que cette page soit servie par le serveur apache
- Il faut donc la mettre dans un répertoire particulier :

`C:\Program Files\EasyPHP\www\`

- Il faut que le fichier ait bien l'extension `.php`
- Il faut indiquer au navigateur qu'il doit s'adresser au serveur web et non au système de fichier :

`http://localhost:port/index.php`

protocole

serveur

chemin de la page

Ca marche pas ...

- Le serveur est arrêté ...
- Le fichier n'est pas au bon endroit ...
- Il n'a pas la bonne extension ...
- Le serveur apache n'est pas activé
- L'adresse entrée dans le navigateur est incorrecte ...
- Y'a une erreur dans le fichier ...

Ca marche !

- Bravo, vous venez d'écrire votre première page en PHP !!
- Mais à quoi sert le PHP ici ? A rien ... On pourrait faire la même chose en HTML pur ...
- Cependant, vous devriez commencer à entrevoir les possibilités ...

Juste pour voir ...

- Affichons le "code source de la page" par le navigateur

Le PHP a disparu, et est remplacé par du HTML !

- Comment est-on passé de :

```
<?php  
    echo "Hello James!";
```

```
<?php
```

à

```
Hello James!
```

I. Le langage PHP

- Chaines de caractères et variables
- Les autres types de données
- Les conditionnelles et les boucles
- Tableau et Date
- Fonctions sur les chaines, les dates et les tableaux
- Les fonctions

Des Chaînes et des Variables

Chaînes de caractères

Ensemble de caractères entourés par :

- ' et ' (chaîne simple)
- " et " (chaîne interprétée)

Chaîne simple :

Les caractères sont utilisés tels quels

Une séquence de caractères spéciale : \'

Chaîne interprétée :

Certains caractères ont des significations spéciales

\n \r \t \\ \\$ \"

Les variables sont remplacées par leur valeur

```
'je suis là !'  
'j'étais là !'  
'je dis : "je suis là !"'  
'je dis : \n "je suis là !"'  
'je dis : <br /> "je suis là !"'
```

```
"je suis là !"  
"j'étais là !"  
"je dis : "je suis là !""  
"je dis : \n "je suis là !""  
"je dis : <br /> "je suis là !""
```

Une chaîne pour quoi faire ?

l'envoyer au navigateur, par exemple ...

```
echo une_chaine_de_caractères ;
```

```
echo 'je suis là !';  
echo 'j'étais là !';  
echo 'je dis : "je suis là !"';  
echo 'je dis : \n "je suis là !"';  
echo 'je dis : <br /> "je suis là !"';
```

```
echo "je suis là !";  
echo "j'étais là !";  
echo "je dis : "je suis là !"";  
echo "je dis : \n "je suis là !"";  
echo "je dis : <br /> "je suis là !"";
```



```
echo 'je suis là !';  
echo 'j'étais là !';  
echo 'je dis : "je suis là !"';  
echo 'je dis : \n "je suis là !"';  
echo 'je dis : <br /> "je suis là !"';
```

```
echo "je suis là !";  
echo "j'étais là !";  
echo "je dis : "je suis là !"";  
echo "je dis : \n "je suis là !"";  
echo "je dis : <br /> "je suis là !"";
```

Mais il y a t-il une différence entre les " et les ' ?

Oui ! L'interprétation des variables ...

Les Variables

Nom d'une variable :

- commence par le caractère \$
- suivi d'une lettre,
- suivi éventuellement de lettres et de chiffres

Les Variables

Chercher l'intrus :

A: \$toto

B: \$james

C: \$007

D: \$Bond

Exemple de variable

Affecter une valeur :

```
$nom = 'Bond, James Bond' ;
```

Envoyer sa valeur vers le navigateur :

```
echo $nom ;
```

ou bien :

```
echo "$nom" ;
```

mais pas :

```
echo '$nom' ;
```

Exemple de variable

Comparez :

```
echo 'My name is ', $nom, ' ! ' ;
```

```
echo "My name is $nom ! " ;
```

```
echo 'My name is '. $nom. ' ! ' ;
```

```
echo 'My name is $nom ! ' ;
```

On peut également récupérer une valeur renvoyée par une *fonction* :

```
$d = date ( "d-m-Y" ) ;
```

Ou bien :

```
$format = "d-m-Y" ;
```

```
$d = date ( $format ) ;
```

nom (paramètres)

A faire en TP

Afficher la date et l'heure avec différents formats

Que des chaînes de caractères ?

Non, bien sûr ...

Types de données

Les données sont manipulées à travers des *constantes* ou des *variables*.

PHP est un langage peu typé :

4 types simples :

- Booléens
- Entiers
- Nombres à virgule flottante
- Chaînes de caractères

2 types composés :

- Tableaux
- Objets

2 types spéciaux :

- La valeur `null`
- Ressources

Booléens

Les booléens sont des données dédiées au calcul logique

Constantes :

`true`

`false`

Conversion :

`0` , `0.0` , `' '` , `"0"` , tableau vide, objet vide, `NULL` : `false`

Tout le reste : `true`

Entiers

Constantes :

12

-10

012

0xff

Conversion :

true : 1

false : 0

Nombre réel : conversion vers 0

Nombres Réels (à virgule flottante)

Constantes :

1.1

1.0

-10.555

1.45e4

Conversion :

?

Attention à la perte de précision :

0.1 est différent de 0.1

Combiner des données

Opérateurs :

Arithmétiques : + - * / % -

Comparaisons : == === != < > <= >=

Logique : **and or xor ! && ||**

Chaîne de caractères : .

Opérateurs arithmétiques

+ (addition)

```
$a=1+1; $b=1+1.0; $c=1+"3"; $d=1+"a3";
```

- (soustraction)

```
$a = 1 - 10;
```

*** (multiplication)**

```
$a =2*2.1; $b=1+1*3;
```

/ (division)

```
$a = 1/4;$b=1.0/4
```

% (modulo)

```
$a = 5%4;
```

- (negation)

```
$a=1; $a = -$a;
```

Opérateurs de comparaison

<code>\$a == \$b</code>	Egal	TRUE si \$a est égal à \$b.
<code>\$a === \$b</code>	Identique	TRUE si \$a est égal à \$b et qu'ils sont de même type.
<code>\$a != \$b</code>	Différent	TRUE si \$a est différent de \$b.
<code>\$a <> \$b</code>	Différent	TRUE si \$a est différent de \$b.
<code>\$a !== \$b</code>	Différent	TRUE si \$a est différent de \$b ou bien qu'ils ne sont pas du même type.
<code>\$a < \$b</code>	Plus petit que	TRUE si \$a est strictement plus petit que \$b.
<code>\$a > \$b</code>	Plus grand	TRUE si \$a est strictement plus grand que \$b.
<code>\$a <= \$b</code>	Inférieur ou égal	TRUE si \$a est plus petit ou égal à \$b.
<code>\$a >= \$b</code>	Supérieur ou égal	TRUE si \$a est plus grand ou égal à \$b.

Attention aux conversions automatiques !
Attention à l'égalité des nombres réels !

```
1 == "1"; 0 !== "0";  
0.2 != 0.1 + 0.1;
```

Opérateurs de comparaison

```
007 == "007"
```

- A : TRUE
- B : FALSE

```
0 != "o"
```

- C : TRUE
- D : FALSE

Opérateurs de comparaison

```
007 !== "007"
```

- A : TRUE
- B : FALSE

```
0.2 == 0.1 + 0.1
```

- A : TRUE
- B : FALSE

Opérateurs logiques

<code>\$a and \$b</code>	ET (And)	TRUE ssi \$a ET \$b valent TRUE.
<code>\$a or \$b</code>	OU (Or)	TRUE ssi \$a OU \$b valent TRUE.
<code>\$a xor \$b</code>	XOR (Xor)	TRUE ssi \$a OU \$b est TRUE, mais pas les deux en même temps.
<code>! \$a</code>	NON (Not)	TRUE ssi \$a est FALSE.
<code>\$a && \$b</code>	ET (And)	TRUE ssi \$a ET \$b sont TRUE.
<code>\$a \$b</code>	OU (Or)	TRUE ssi \$a OU \$b est TRUE.

Opérateur de chaînes

`$a . $b` concaténation

Opérateurs d'assignation

- Opérateur simple :
 `$a = 3;`
 `$a = ($b = 4) + 5;`
- Opérateur combiné :
 `$a += 3;`

Contrôles ...

Structures de Contrôle

Les structures de contrôle modifient le déroulement du programme :

- Conditions (branchements)

- Boucles (répétitions)

Conditions

Les conditions, ou branchements, permettent, en fonction d'un *test logique* d'exécuter ou non une séquence d'instructions.

```
if(expression_logique)
    {suite_d_expression}

if(expression_logique)
    {suite_d_expression1}
else
    {suite_d_expression2}

if(expression_logique)
    {suite_d_expression1}
elseif(expression_logique)
    {suite_d_expression2}
else
    {suite_d_expression3}
```

A faire en TP

```
<?php
$a = $_POST['age'];
if($a < 20) {
    echo "Que vous êtes jeune !";
}
elseif($a < 50) {
    echo "En pleine force de l'âge";
}
else{
    echo "Bientôt la retraite ..."
}
?>
```



```
switch(expression) {  
    case une_valeur1 : sequence1  
    case une_valeur2 : sequence2  
  
    default : sequence_default  
}
```

- Une instruction spéciale :
break ;

A faire en TP

```
<?php
```

```
    $prenom = $_POST['amis']
```

```
switch($prenom) {
```

```
    case 'James' : echo "See you on Friday $prenom !"  
                    break;
```

```
    case '007' :
```

```
    case 'Sébastien' :  
        echo "Hello $prenom";  
        break;
```

```
    default :
```

```
        echo "Dear $prenom";
```

```
}
```

```
?>
```

Boucles

Permettent de répéter une séquence d'instructions selon une condition :

```
while (expression_logique)  
    {suite_d_expression}  
for (init; cond; modif;)  
    {suite_d_expression}
```

A faire en TP

- Une page XHTML formulaire :
demande une boisson et un nombre n
- Une page php, affichant n fois, avec une police croissante "Excellent choix !"
- Ajouter au php une vérification préalable (boissons gazeuses -> « tout de suite monsieur ») mais (Volka Martini -> "je vous reconnais bien là"), (Bière -> "voyons") et autre -> "désolé mais nous n'avons pas"

Une structure de données multi-usage

Array

Tableaux

- Tableau en PHP est un ensemble de valeurs
- Chaque valeur est accessible par une clé
- Création d'un tableau pré-rempli :

```
array( key1 => value1, ..., key2 => value2 )
```

`key_i` peut être de type entier ou chaîne

`value_i` peut être de n'importe quel type

Si `key` est omis, la clé est l'entier immédiatement supérieur à la dernière clé entière

A faire en TP

```
$personne = array('nom'      => 'dupond',  
                  'prenom'  => 'jean',  
                  'age'     => 33);
```

```
$p2 = array('dupond', 'jean', 33);
```

```
$p3 = array(0=>'dupond', 2=>33, 1=>'jean');
```

```
echo $personne['prenom'];
```

```
$personne['nom'] = 'durand';
```

```
$personne['adresse'] = 'nice';
```

```
$p3[] = 'Caen';
```

```
$p4 = array(0=>'dupond', 'age'=>33, 'jean');
```

Comment afficher la valeur 'jean' ?

- A: `echo $p4['prenom'];`
- B: `echo $p4[1];`
- C: `echo $p4[2];`
- D: `echo $p4['age'+1];`

A faire en TP

- Affichage de la date "en français" en utilisant des tableaux

Fonctions sur les tableaux

- **explode** -- Coupe une chaîne en segments :

```
array explode ( string separator, string string [, int  
    limit] )
```
- **implode** -- Rassemble les éléments d'un tableau en une chaîne

```
string implode ( string glue, array pieces )
```
- **shuffle** -- Mélange les éléments d'un tableau

```
bool shuffle ( array &array )
```

Fonctions sur les tableaux

- `count` -- Compte le nombre d'éléments d'un tableau ou le nombre de propriétés d'un objet :

```
int count ( mixed var [, int mode] )
```

- `sort` -- Trie un tableau

```
bool sort ( array &array [, int sort_flags] )
```

- `print_r` -- Affiche des informations lisibles pour une variable

```
bool print_r ( mixed expression [, bool return]
```

Une structure de contrôle dédiée

```
foreach (array_expression as $value) {  
    commandes  
}
```

```
foreach (array_expression as $key => $value) {  
    commandes  
}
```

A faire en TP

- Vérifier la validité d'une date à partir d'une chaîne ('xx/yy/zzzz') récupérée d'un formulaire
- Affichage html d'un répertoire (nom, prénom, courriel) en tableau php

Les fonctions

Code modulaire

Définition des fonctions

```
function foo($arg_1, $arg_2, /* ..., */ $arg_n)
{
    // ...
    return $retval;
}
```

Passage d'argument

- **Par valeur (cas classique)**

Si la valeur est modifiée dans la fonction, les modifications ne sont pas connues à l'extérieur

- **Par référence**

```
function add_1 (&$nombre)
{
    $nombre += 1;
}
```


Portée des variables

- La portée est locale à la fonction :
 - Une variable déclarée à l'extérieur n'est pas connue à l'intérieur
 - Une variable déclarée à l'intérieur n'est pas connue à l'extérieur

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

Portée des variables

- mot clé *global*

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

- Super globale **GLOBALS []**

```
$a = 1; /* portée globale */  
function test() {  
    echo $GLOBALS['a'] ;  
}
```

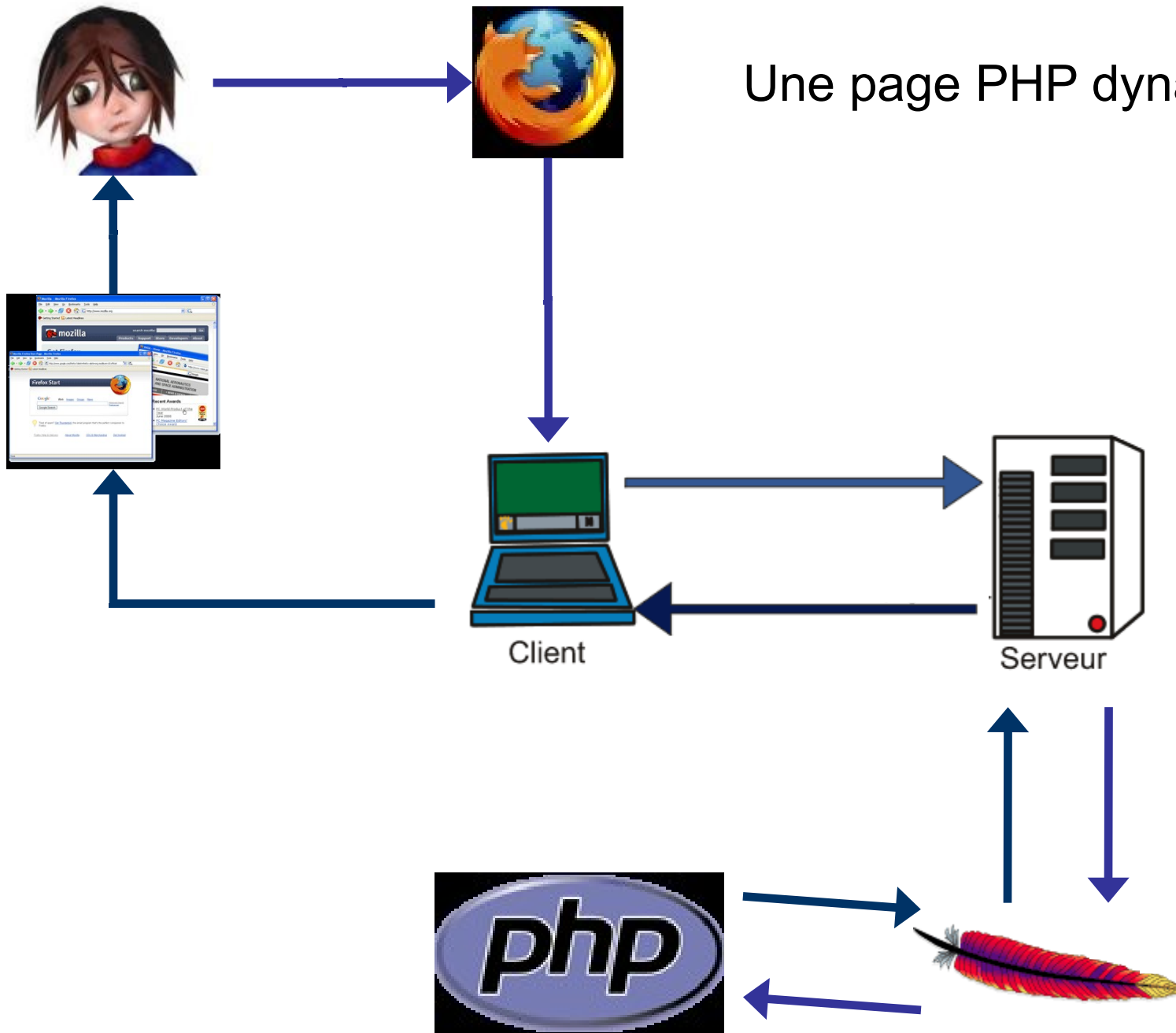
Les variables superglobales

- \$GLOBALS
- \$_SERVER
- \$_GET
- \$_POST
- \$_COOKIE
- \$_FILES
- \$_ENV
- \$_REQUEST

D'où viennent les données ?

Des formulaires, par exemple

Une page PHP dynamique



```
<html><body>
<form method="post" action="affiche.php">
Nom : <input type="text" name="nom"><br />
Prénom : <input type="text" name="prenom">
<input type="submit" value="OK">
</form></body></html>
```

formulaire.html

```
<?php
$prenom = $_POST['prenom'];
$nom = $_POST['nom'];
echo "<center>Bonjour $prenom $nom</center>";
?>
```

affiche.php

Les différents éléments

- text
- textarea
- checkbox
- radio
- liste à choix unique : select
- liste à choix multiple : select
- bouton : submit

```
<html><body>
<form method="post" action="affiche.php">
Nom : <input type="text" name="nom"><br />
Prénom : <input type="text" name="prenom"/>
<input type="submit" value="OK"/>
</form></body></html>
```

formulaire.html

```
<?php
$prenom = $_POST['prenom'];
$nom = $_POST['nom'];
echo "<center>Bonjour $prenom $nom</center>";
?>
```

affiche.php


```
<html><body>
<form method="post" action="affiche.php">
Adresse : <textarea name="adresse"></textarea>
<br />
<input type="submit" value="OK" />
</form></body></html>
```

formulaire.html

```
<?php
$adresse = $_POST['adresse'];
echo "Vous habitez : <br /> $adresse ";
?>
```

affiche.php

```
<html><body>
<form method="post" action="affiche.php">
foot<input type="checkbox" name="sports[]" value="foot"/>
hand<input type="checkbox" name="sports[]" value="hand"/>
rugby<input type="checkbox" name="sports[]"
value="rugby"/>
<input type="submit" value="OK"/>
</form></body></html>
```

formulaire.html

```
<?php
$sports = $_POST['sports']; // tableau
foreach($sports as $value){
    echo "$value <br />";
}
?>
```

affiche.php

```
<html><body>
<form method="post" action="affiche.php">
vert <input type="radio" name="couleur" value="vert"/>
rouge <input type="radio" name="couleur" value="rouge"/>
<input type="submit" value="OK"/>
</form></body></html>
```

formulaire.html

```
<?php
$couleur = $_POST['couleur'];
echo "couleur préférée : $couleur";
?>
```

affiche.php

```
<html><body>
<form method="post" action="affiche.php">
<select name="couleur">
<option value="vert"> vert </option>
<option value="rouge"> rouge </option>
</select>
<input type="submit" value="OK"/>
</form></body></html>
```

formulaire.html

```
<?php
$couleur = $_POST['couleur'];
echo "couleur préférée : $couleur";
?>
```

affiche.php

```
<html><body>
<form method="post" action="affiche.php">
<select name="sports[]" size="3" multiple>
<option value="foot"> foot </option>
<option value="hand"> hand </option>
<option value="rugby"> rugby </option>
</select>
<input type="submit" value="OK"/>
</form></body></html>
```

formulaire.html

```
<?php
$sports = $_POST['sports']; // tableau
foreach($sports as $value){
    echo "$value <br />";
}
?>
```

affiche.php

```
<html><body>  
<form method="post" action="affiche.php">  
<input type="submit" value="OK" name="valide"/>  
</form></body></html>
```

formulaire.html

```
<?php  
$valide = $_POST['valide'];  
echo "$valide";  
?>
```

affiche.php

```
<?php
if (isset($_POST["valide"])) {
// traitement formulaire
}
else { ?>
<html><body>
<form method="post"
      action="formulaire2en1.php">
<!--      le formulaire      -->
<input type="submit" value="OK" name="valide"/>
</form></body></html>
<?php } ?>
```

Fonctions Utiles

addslashes -- Ajoute des anti-slash dans une chaîne

```
string addslashes ( string str )
```

stripslashes -- Supprime les anti-slash d'une chaîne

```
string stripslashes ( string str )
```

mysql_real_escape_string -- Protège les caractères spéciaux d'une commande SQL

```
string mysql_real_escape_string ( string  
unescaped_string [, resource link_identifier] )
```

htmlspecialchars -- Convertit les caractères spéciaux en entités HTML

```
string htmlspecialchars ( string string [, int  
quote_style [, string charset]] )
```

htmlentities -- Convertit tous les caractères éligibles en entités HTML

```
string htmlentities ( string string [, int quote_style  
[, string charset]] )
```

html_entity_decode -- Convertit toutes les entités HTML en caractères normaux

```
string html_entity_decode ( string string [, int quote_style  
[, string charset]] )
```

urldecode -- Décode une chaîne encodée URL

```
string urldecode ( string str )
```

urlencode -- Encode une chaîne en URL

```
string urlencode ( string str )
```

utf8_encode -- Convertit une chaîne ISO-8859-1 en UTF-8

```
string utf8_encode ( string data )
```

utf8_decode -- Convertit une chaîne UTF-8 en ISO-8859-1

```
string utf8_decode ( string data )
```

Passage par cookies

- PHP propose des fonctions pour :

Poser un cookie :

```
<?
    $expire = 365*24*3600; // durée du cookie en sec, 1 an
    setcookie("lenom", "lavaleur", time()+$expire);
?>
```

Récupérer un cookie :

```
<? // On affiche la valeur de nickname
    echo 'le cookie "lenom" a pour valeur : ';
    echo $_COOKIE["lenom"];
?>
```

Mais ...

Aucune sécurité