PHP

© Philippe Roose
IUT de Bayonne/UPPA

PRÉAMBULE - INTERNET

Internet

- Réseau d'interconnexion entre deux ou plusieurs réseaux.

L'internet

- L'ensemble des réseaux internet utilisant le protocole TCP / IP en respectant les standards en vigueurs

PRÉAMBULE - INTERNET

Structure arborescente

- Réseau local (laboratoire, département,…)
- Réseau local (campus, entreprise,…)
- Réseau régional
- Réseau national
- Réseau mondial

HISTORIQUE D'INTERNET

1959-1968 : Programme ARPA

- Ministère américain de la défense : lancer un réseau capable de supporter les conséquences d'un conflit nucléaire.

1969 : ARPANET (l'ancêtre d'Internet)

- Les universités américaines s'équipent de gros ordinateurs et se connectent au réseau ARPANET.

1970-1982 : ouverture sur le monde

- Premières connexions avec la Norvège et Londres

1983 : Naissance d'Internet

- Protocole TCP/IP tous les réseaux s'interconnectent, les militaires quittent le navire

HISTORIQUE D'INTERNET

1986

- La National Science Fondation (NSF) décide de déployer des super-ordinateurs pour augmenter le débit d'Internet

1987-1992

- Les fournisseurs d'accès internet apparaissent et les entreprises se connectent au réseau

1993-2006

- Ouverture au grand public
- Avènement du courrier électronique et du Web
- Marché considérable
- Naissance des services Internet (cf. dispo suivante)

SERVICES D'INTERNET

Un service

- Vision « réseau informatique»
  - Application qui utilise un protocole et un numéro de port
  - Fonctionnement en mode client/serveur au dessus de TCP/IP

- Vision « utilisateur »
  - Application qui « *fait ou permet de faire quelque chose* »

MODÈLE INTERNET

Une version du modèle OSI (*très simplifié*)

Service

- **ftp, www, mail, …**

Transport

- TCP, UDP (entre deux processus aux extrémités)

Réseau

- IP (routage)

Transmission

- Entre 2 sites : pas de protocoles spécifiques

SERVICES DE BASE DE L'INTERNET

FTP (**F**ile **T**ransfert **P**rotocol)

- Transfert de fichier

E-mail

- SMTP (Simple Mail Transfert Protocol)
- POP (Post Office Protocole)
- IMAP (Internet Message Access Protocol)

News

- NNTP (News Network Transport Protocol)

SERVICES DE BASE DE L'INTERNET

World Wide Web (ou Web)

- Le web repose sur l'hypertexte
- Une page hypertexte est écrite en
  - Hyper Text Mark-up Language
- Le protocole utilisé est le HTTP
  - Hyper Text Transport Protocol

WEB

Architecture pour accéder à des documents liées entre eux et situés sur des machines reliées par Internet

Architecture basée sur 3 concepts :

- La localisation : URL
- Le protocole : HTTP
- Le langage : HTML

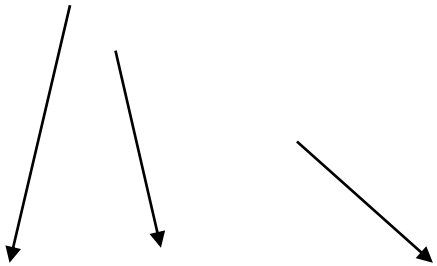
Popularité due à :

- Interface graphique conviviale
- Très grande quantité d'informations
- Grande diversité d'informations

UNE URL

URL : *Uniforme Ressource Locator : Adresse Universelle de Ressource*

- Le protocole (*comment*),
- Le nom DNS (*où*),
- Le nom du document (*quoi*).
- Exemple d'URL :
  - http://www.perdu.com/index.html



Autre exemple d'URL : http://www.perdu.com/index.html?var=1345

- *protocol*://**host_name**:*port*/**path**/*extrapath*?**arguments**
- *Extrapath :* permet de passer des informations à des programmes s'exécutant sur le serveur

FONCTIONNEMENT DU WEB

Le client (*browser/butineur*) dialogue avec un serveur web selon le protocole HTTP

Le serveur vérifie la demande, les autorisations et transmet l'information requise

Le browser interprète le fichier reçu et l'affiche

A ceci peut s'ajouter des contrôles ou des exécutions (côté client ou côté serveur)

- Coté Client (*en javascript* par exemple)

VISION CÔTÉ SERVEUR

Le serveur est en permanence à l'écoute des requêtes formulées par les clients (*qui peuvent être très nombreuses et nombreux !*)

Il vérifie la validité de la requête...

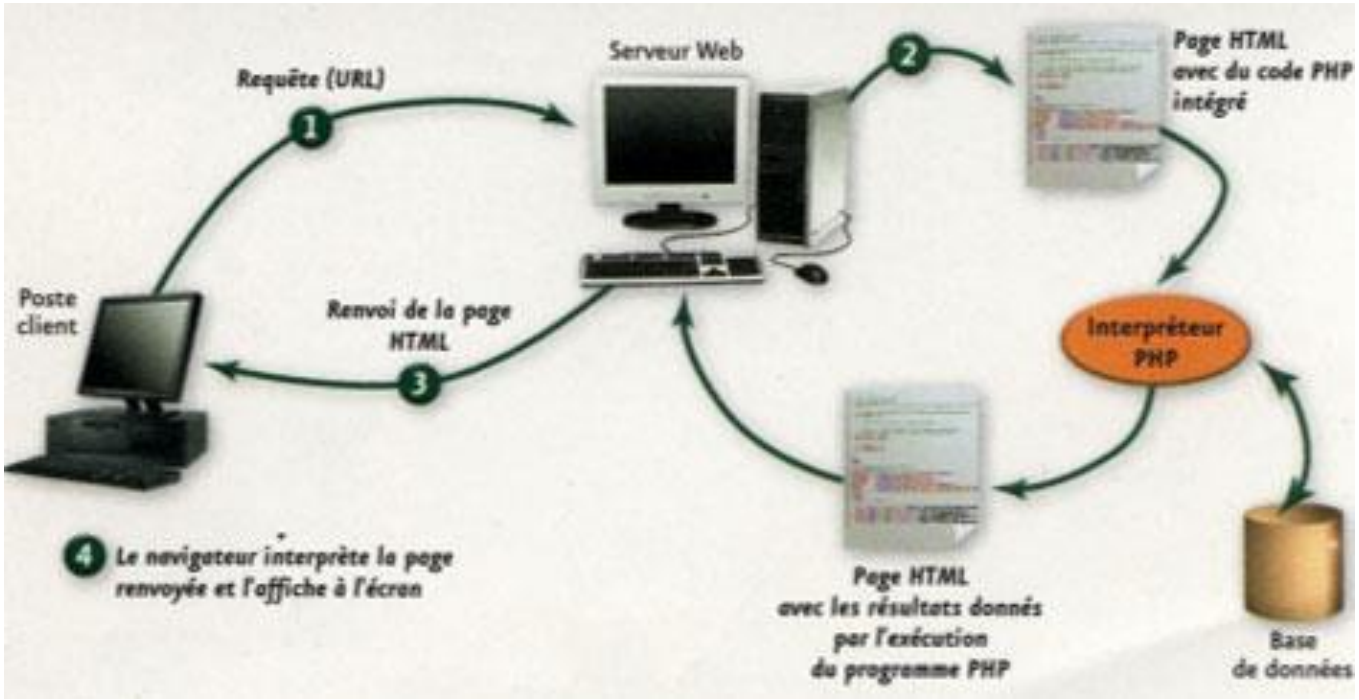
- adresse correcte (URL)
- client autorisé à accéder au document

...et y répond : envoi du texte, des images, du code à exécuter sur le client, d'un message d'erreur, d'une demande d'authentification, ...

Il peut exécuter un programme localement qui va générer une réponse HTML

- pages dynamiques => PHP – CQFD !

ARCHITECTURE GLOBALE



BESOINS DE PAGES DYNAMIQUES

Définition Wikipedia :

- *Un **site Web dynamique** est un site Web dont les pages peuvent être générées dynamiquement, c'est-à-dire à la demande, contrairement au site Web statique.*

Exemples caractéristiques

- Boursorama (http://www.boursorama.com/)
- FNAC (www.fnac.com)

Mais aussi

- Pages dont le contenu est fonction du contexte (matériel, environnemental, choix/préférences des usagers
  - l'heure, du visiteur, de ses interactions précédentes, des précédentes requêtes, du contenu des données saisies dans des formulaires, de la t°, de la position GPS, du niveau de batterie, etc.).

LANGAGE STATIQUE VS DYNAMIQUE

Langage statique ou côté client

- Langage qui n'utilise pas de serveur pour exécuter son code, ex : HTML, Javascript pour exécuter une page en HTML simple qui est sur votre ordinateur.

  - Cible : sites simples, évoluant peu (sites personnels)
  - + : Facile pour le péquin moyen, nécessite juste les bons outils
  - - : Ne permet pas de faire évoluer contenu + présentation facilement ou automatiquement

Langage dynamique ou côté serveur

- C'est un langage qui DOIT utiliser un serveur pour exécuter son code, ex : ASP, PHP, CGI, ASP.net, JSP, etc.,
- Il faut avoir un serveur web (*Nginx, Apache, IIS, etc.*) **PLUS** un module qui retranscrira les commandes du langage associé.
  - Cible : tout le reste !
  - - : Pas facile pour le même péquin moyen que précédemment. Nécessite un apprentissage des langages + connaissances algorithmiques, de BD, de sécurité, etc.
  - + : Permet pas de faire évoluer contenu + présentation facilement ou automatiquement

LE LANGAGE...

INTRODUCTION AU PHP (PERSONNAL HOME PAGE PHP : HYPERTEXT PROCESSOR)

Langage né en 1994 par Rasmus Lerdof pour son CV et pour garder trace de consultations (travaille chez Yahoo! maintenant).

PHP est un disponible dans plusieurs environnements, tels qu'Unix (Linux, AIX), Windows (95, 98, NT) et Macintosh.

PHP est un langage de programmation spécialisé dans la génération de code, dont le langage de prédilection est le HTML (crée en, 1984 par le CERN).

Atout principaux : quantité d'outils

- *manipulation d'images, traitement de fichiers, accès aux bases de données, paiement sécurisé, accès aux annuaires, etc.*

OUTILS NÉCESSAIRES

Wamp (Windows), Mamp (Mac) Lamp (Linux)

- Serveur http + mysql + php

Un bon éditeur de texte

- Emacs, Notepad++

Un éditeur HTML

- Dreamweaver, NVU

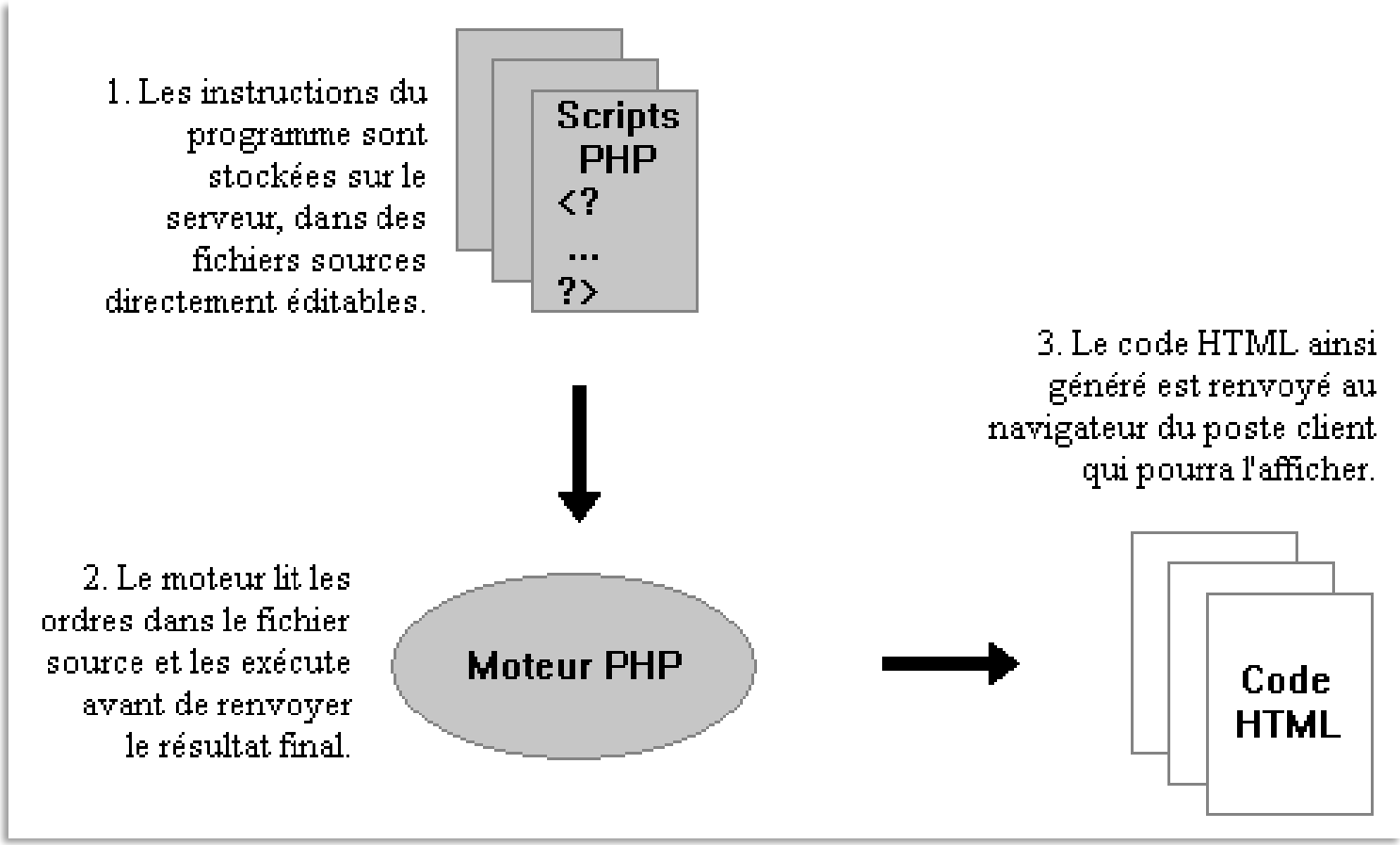
LE LANGAGE

Le PHP est un langage complet, écrit en C, qui reprend une grande partie des spécificités techniques et sémantiques de ce langage.

Le moteur d'interprétation du langage lit un fichier source PHP, puis génère un flux destination, en respectant les définitions et règles suivantes :

- Un bloc PHP est un groupe continu de lignes, encadré par deux balises : <? et ?>, <?php et php?> ou **<?php et ?>** (cas à l'IUT)
- Toute ligne située à l'extérieur de ces balises n'est pas interprétée et est renvoyée telle quelle dans le flux de sortie.
- Toute ligne située à l'intérieur de ces balises est considérée comme une instruction PHP et est donc interprétée par le moteur.
- Les instructions PHP n'apparaissent pas dans le résultat généré.
- Lorsqu'une erreur survient, un message est intégré dans le flux de sortie, et la génération du script est interrompue.

SCHÉMA DE FONCTIONNEMENT GÉNÉRAL D'UN LANGAGE DE SCRIPT



PREMIER PROGRAMME

La conception du script PHP est réalisé avec un éditeur quelconque, Emacs (hé hé hé !) est très bien pour cela.

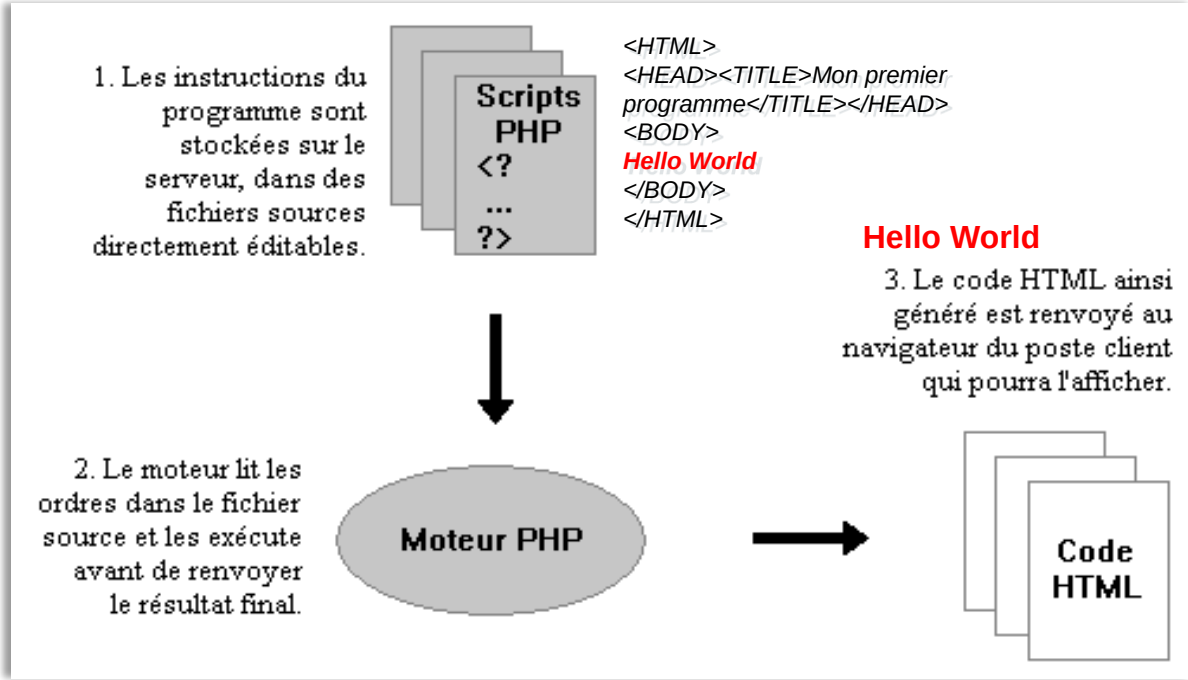
```
<HTML>
<HEAD><TITLE>Mon premier programme</TITLE></HEAD>
<BODY>
<?php
    print "Hello World"; // Interdiction de faire un autre premier programme !
?>
</BODY>
</HTML>
```

L'extension de votre fichier doit être .php afin d'être interprété par le serveur (Apache par exemple)

- *helloworld.php*
- => *~login/WWW/helloworld.php*

MOTEUR EXÉCUTION

```
<HTML>
<HEAD><TITLE>Mon premier programme</TITLE></HEAD>
<BODY>
<?php
  print "Hello World"; // Interdiction de faire un autre premier programme !
?>
</BODY>
</HTML>
```



LES VARIABLES

Contrairement à beaucoup de langages, PHP ne contient de partie déclarative clairement définie. Pour déclarer une variable, il suffit de l'initialiser. Celle-ci sera immédiatement accessible, et le restera jusqu'à la fin du script.

Les variables en PHP sont toutes dotés du préfixe $

```
$toto = « zertyuiop »;
print $totot;
```

Le type d'une variable est également défini par la valeur qui lui à été affecté lors de sa création. Il existe 5 types de données :

- Entier (*int, integer*)
- Décimal (*real, double, float*)
- Chaîne de caractères (*string*)
- Booléen (*boolean*)
- Tableau (*array*)
- Objet (*object*)
- Ressource (*accès ressource physique*)
- NULL (rien !)

PORTÉE DES VARIABLES

Il existe trois niveaux de définition de variables :

- Le niveau global. Il définit des variables dans l'intégralité du code d'une page PHP.
- Le niveau local. Il définit des variables propres à une fonction, dont la durée de vie ne dépasse pas le temps de cette fonction.
- Le niveau static. Il définit des variables propres à une fonction, qui persistent pendant l'intégralité du code de la page PHP.

LES TABLEAUX

Dans PHP, comme dans tout autre langage, les tableaux sont une structure de données incontournable.

La déclaration d'un tableau se fait de la même manière que la déclaration d'une variable avec un indice se trouvant entre [ ].

LES TABLEAUX

$tableau[0] = 1; // on crée un tableau, et sa première valeur est 1

Pour un tableau à deux dimensions, il suffit de mettre un second indice au moment de l'affectation.

$tableau[0][0] = 1; // on crée un tableau, et sa première valeur est 1

- Il n'est pas obligatoire de préciser l'indice pour affecter une valeur.

exemple :

$tableau[] = 1; // équivaut à $tableau[0] = 1;

$tableau[] = 45; // équivaut à $tableau[1] = 45;

$tableau[] = 6; // équivaut à $tableau[2] = 6;

LES TABLEAUX ASSOCIATIFS

Fonction 'array()'. Permet de préciser les indices ainsi que les valeurs du tableau (à l'aide de l'opérateur =>).

$tableau[ ] = array(0=>1, 1=>45, 2=>6);

$tableau[ ] = array("rouge"=>"red", "vert"=>"green", "bleu"=>"blue");

- echo $tableau['rouge'] affichera 'red'

La navigation dans les éléments du tableau s'effectue à l'aide des fonctions 'next()' , 'prev()' et 'each()'.

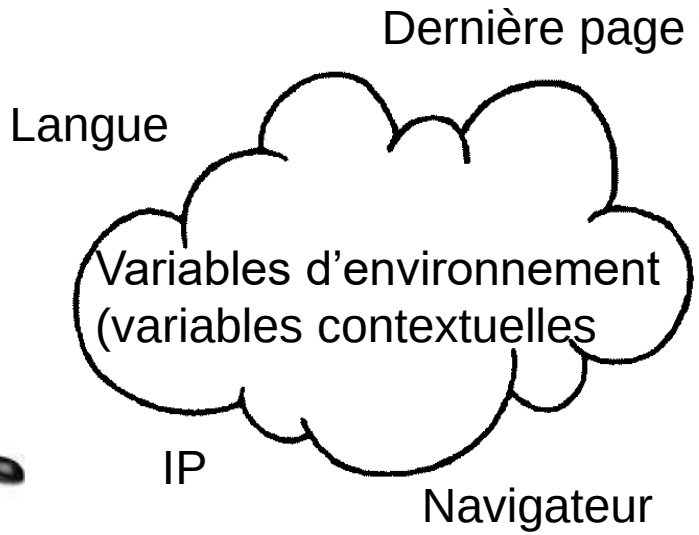
Le nombre d'éléments d'un tableau peut être obtenu à l'aide de la fonction 'count()'. Le tri des tableaux est facilité par de nombreuses fonctions : *asort(), ksort(), sort(), usort(), etc...*

LES TABLEAUX

```
$nombres = array(3, 6, 9);
for($i=0; $i<count($nombres); ++$i) {
 echo $i.' '.$nombres[$i].'<br/>';
}

$nombres = array(3, 6, 9);
foreach($nombres as $nombre) {
 echo $nombre.'<br/>';
}
```

VARIABLES D'ENVIRONNEMENT/SUPERGLOBALES

L'un des aspects fondamentaux d'une application construite sur une architecture de type intranet est l'utilisation des variables d'environnement du serveur, et notamment celles du serveur HTTP.



VARIABLES D'ENVIRONNEMENT/SUPERGLOBALES

Avec PHP toutes les variables d'environnement du serveur sont automatiquement reprises dans les scripts PHP en tant que variables globales. Ainsi il suffit de les utiliser directement dans le code.

- **$_GET** : Les valeurs provenant de l'URL ;
- **$_POST** : Les valeurs envoyées par formulaire ;
- **$_FILE** : Les fichiers envoyés par formulaire ;
- **$_SERVER** : Les valeurs mises en place par le serveur Web (elles peuvent donc changer d'une configuration à l'autre) ;
- **$_ENV** : Les variables d'environnement (système d'exploitation) ;
- **$_SESSION** : Les valeurs mises dans le magasin des sessions ;
- **$_COOKIE** : Les valeurs transmises au moyen de cookies par le navigateur ;
- **$GLOBALS** : L'ensemble des variables du script.

*$IP=$_SERVER['**REMOTE_ADDR**']*

echo $IP; // @ IP de la machine cliente

VARIABLES D'ENVIRONNEMENT/SUPERGLOBALES

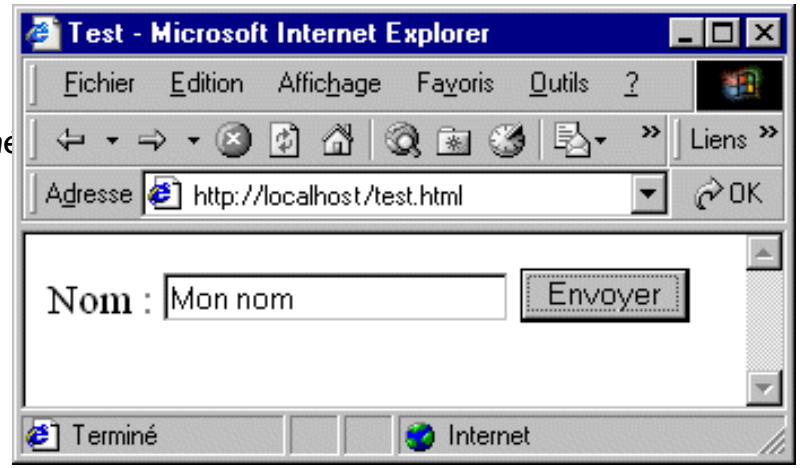
$_SERVER

- **SERVER_NAME** : Le nom du serveur hôte qui exécute le script.
- **PHP_SELF :** Le nom du fichier du script en cour d'exécution.
- **REMOTE_ADDR** : L'adresse IP du client qui demande la page courante.
- …

VARIABLES ISSUES DE FORMULAIRES

Les variables issues de formulaires HTML correspondent aux différents champs positionnés entre les balises <FORM> et </FORM> de ce formulaire. La page qui reçoit ces variables est celle qui est désignée par l'attribut ACTION de la balise <FORM>.

```
<HTML>
<FORM ACTION="test.php" METHOD=POST>
<INPUT TYPE=hidden NAME= "cache" VALUE= " est cache
Nom : <INPUT TYPE=text NAME="nom">
<INPUT TYPE=submit VALUE="Envoyer">
</FORM>
</HTML>
```



VARIABLES ISSUES DE FORMULAIRES

Tableaux :

- $_GET
- $_POST

Exemple (*test.php*)

```
$var1 = $_POST['cache'];
$var2 = $_POST['nom']
print "$var2 $var1" ;

=> Mon nom est caché
```

SELECTIONS

```
<select name="prenom">
    <option value="adriana">Adriana</option>
    <option value="alessandra">Alessandra</option>
    <option value="candice">Candice</option>
    <option value="lili">Lili</option>
</select>
```

```
<?php
   $unprenom=$_POST["prenom"];
   …
?>
```

SELECTIONS MULTIPLES

```
<select multiple name="prenom[]">
    <option value="adriana">Adriana</option>
    <option value="alessandra">Alessandra</option>
    <option value="candice">Candice</option>
    <option value="lili">Lili</option>
</select>
```

```
<?php
foreach($_POST['prenom'] as $valeur)
{
   print "$valeur a été sélectionné<br>";
   …
}
?>
```

TESTS : IF... THEN... ELSE

Test de base que l'on trouve dans la majeure partie des langages. Si une condition est vrai alors on exécute des instructions sinon (facultatif) on en exécute d'autre. On peut changer de condition avec un 'elseif'.

```
if ( condition1 ) {
   Action 1
} elseif ( condition2 ) {
   Action 2
   }
else {
Action 3
}
```

TESTS : IF... THEN... ELSE

```
if ($a==$b) {
   echo "A est égal à B";
} elseif ($a > $b) {
   echo "A est supérieur à B";
} else {
   echo "A est inférieur à B";
}
```

TESTS : SWITCH... CASE... DEFAULT

Si les conditions successives ne portent que sur la valeur d'une variable, on pourra avantageusement remplacer le test 'if... elseif... else' par 'switch'. Dans ce test, la condition est associée à la valeur d'une variable, de plus l'instruction 'break' est primordiale à la fin de chaque bloc de conditions, sinon toutes les conditions seront vérifiées et exécutées.

```
switch ($a) {
   case $b:
      echo "A est égal à B";
      break;
   case >$b:
      echo "A est supérieur à B";
      break;
   default:
      echo "A est inférieur à B";
      break;
}
```

BOUCLES

```
while ( condition ) {
   Action;
}

do {
   Action;
} while (condition);

for (expr1; expr2; expr3) {
   Action;
}
```

```
$i=1;
while ($i <= 10) {
   echo "- $i -";
   $i++;
}

$i=1;
do {
   echo "- $i -";
   $i++;
} while ($i <= 10)

for ($i = 1; $i <= 10; $i++) {
   echo "- $i -";
}
```

BREAK;CONTINUE

Break :Cette instruction permet de sortir de n'importe quelle boucle, à n'importe quel moment.

Continue : Cette instruction permet de ne pas exécuter le code contenu dans la boucle et de passer à l'itération suivante.

```
for ($i=1; $i<=10; $i++) {
   if ($i<=5) { echo $i; }
   else {
     break;
   }
 echo "- ";
}
// cette boucle affichera : 1 - 2 - 3 - 4 - 5
```

LES FONCTIONS

Comme dans tout langage structuré, en PHP, les fonctions sont la base d'une programmation claire et efficace.

Une fonction est une sorte de sous-programme isolé du reste du code, exécutable à tout moment, depuis n'importe quelle partie du code principal ou n'importe quelle autre fonction, par simple appel.

Les avantages des fonctions sont :

- La non répétition de la même séquence de code
  - Le gain de productivité.
  - La meilleure lisibilité du code.
  - La maintenance facilitée.

Déclaration : le mot clé *function*.

- *function nomdefonction ([paramètres éventuels])*

DÉCLARATION, PARAMÈTRES, VALEURS DE RETOUR

exemple :

```
function bonjour( ) { // ceci est une procédure
   echo « Bonjour ! » ;
}
```

Cette fonction ne fait qu'afficher *'bonjour'* et ne retourne aucun résultat, on l'utilisera de la manière suivante :

```
Bonjour( ) ; // affiche 'bonjour' à l'écran
```

Pour que la fonction retourne un résultat, on utilisera le mot clé return

```
function bonjour2( ) { // ceci est une 'vrai' fonction
   return " Bonjour ! ";
}
echo bonjour2( ) ;
$variable=bonjour2();
```

bonjour affiche le résultat elle-même, alors qu'il faut afficher le résultat de bonjour2 pour obtenir une action similaire.

DÉCLARATION, PARAMÈTRES, VALEURS DE RETOUR

```
function dire_texte($qui, $texte = 'Bonjour') {
   if(empty($qui)){ // $qui est vide, on retourne faux
     return false;
   }else{
      echo "$texte $qui"; // on affiche le texte
      return true; // fonction exécutée avec succès
      }
}
```

DÉCLARATION, PARAMÈTRES, VALEURS DE RETOUR

Passage par valeur par référence :

Le passage des paramètres tel qu'on l'a vu précédemment est ce que l'on appelle le passage par valeur. Il existe une autre manière de procéder :le passage par référence. On passe à la fonction la référence (adresse mémoire) d'une variable existante, et la fonction modifie directement la valeur de cette variable.

exemple :

```
function bonjour(&$phrase) {
  $phrase= « bonjour Toto Dupont » ;
}
$chaine = « Phrase qui va disparaître » ;
bonjour($chaine) ;
echo $chaine ; // affiche 'bonjour Toto Dupont' à l'écran
```

OPÉRATEURS LOGIQUES

Ils permettent de combiner plusieurs tests entre eux.

exemple : Ici $a et $b peuvent prendre les valeurs booléennes vrai ou faux.

Opérateur	**Exemple**	**Résultat**
and (&&)	$a and $b $a && $b	vrai si $a et $b sont vrai tous les deux
or (||)	$a or $b $a || $b	vrai si $a est vrai ou $b est vrai, ou encore si $a et $b sont vrai tous les deux
not (!)	not $a !$a	vrai si $a est faux !$a

OPÉRATEURS DE COMPARAISON

Ils permettent de comparer les valeurs de deux variables.

exemple : Ici $a et $b sont du même type de variable.

Opérateur	**Exemple**	**Résultat**
==	$a == $b	vrai si $a est égal à $b
!=	$a != $b	vrai si $a est différent de $b
<	$a $lt; $b	vrai si $a est inférieur $b
>	$a > $b	vrai si $a est supérieur $b
<=	$a <= $b	vrai si $a est inférieur ou égal à $b
>=	$a >= $b	vrai si $a est supérieur ou égal à $b

FONCTIONS SPÉCIFIQUES À PHP

GESTION DE LA DATE

Fonction Date : nombreux paramètres

```
echo date("d-m-Y"); // affiche : "12-12-2000"
```

GESTION DES FICHIERS

La fonction de base est la fonction *fopen( )*. C'est elle qui permet d'ouvrir un fichier, que ce soit pour le lire, le créer, ou y écrire. Sa syntaxe est :

```
entier fopen(chaine nomdufichier, chaine mode);
```

Différents modes disponibles

r : ouverture en lecture seulement

w : ouverture en écriture seulement (la fonction crée le fichier s'il n'existe pas)

a : ouverture en écriture seulement avec ajout du contenu à la fin du fichier (la fonction crée le fichier s'il n'existe pas)

r+ : ouverture en lecture et écriture

w+ : ouverture en lecture et écriture (la fonction crée le fichier s'il n'existe pas)

a+ : ouverture en lecture et écriture avec ajout du contenu à la fin du fichier (la fonction crée le fichier s'il n'existe pas).

GESTION DES FICHIERS

Exemples :

```
$fp = fopen("../fichier.txt","r"); //lecture
```

```
$fp = fopen("ftp://iutbayonne.univ-
 pau.fr./pub/fichier.txt","w");
       //écriture depuis début du fichier
```

```
$fp = fopen("http://www.nexen.com/fichier.txt","a");
       //écriture à partir de la fin du fichier
```

LECTURE DANS UN FICHIER

```
$monfichier = fopen("monfichier.txt", "r") ;
if ( !($monfichier)) {
  print("Impossible d'ouvrir le fichier ") ;
  exit ;
}
while ( !feof($monfichier) ) {
  $ligne = fgets($monfichier,255);
  print "$ligne <BR> "  ;
}
fclose ($monfichier) ;
```

ÉCRITURE DANS UN FICHIER

```
$monFichier = fopen("monfichier.txt","w") ; // ouverture en écriture
if ( !($monfichier)) {
  print("Impossible de créer le fichier \n") ;
  exit ;
}

fputs($monfichier, "ligne 1") ;
fputs($monfichier, "ligne 2") ;
fputs($monfichier, "\n"); // retour à la ligne

fclose($monfichier); // on libère la ressources
```

FONCTION EXPLODE

Il est fréquent d'avoir des fichiers au format CSV (*Comma Separated Value*).

Sa syntaxe est la suivante

explode ("caractère délimiteur", chaîne de donnée)

Fichier csv

Dupont | Pierre | 27

Durand | David | 34

…

GESTION DE FICHIERS & FONCTION EXPLODE

```
if (!file_exists("test.txt")) {
  print "<H3><BR>Erreur, fichier manquant<BR>";
  exit;
} else {
  $fd = fopen($fic,"r");
   while (!feof($fd)) {
    $ligne = fgets($fd,255);
    $tab=explode("|",$ligne);
    print "Nom : $tab[0]<br>";
    print "Prénom : $tab[1]<br>";
    print "Age : $tab[2]<br>";
}
   fclose($fd);
}
```

+ fgetcsv

ACCÈS FICHIERS DISQUE

```
$fd = opendir($Directory)
readdir($fd)// => NULL | Rien
is_dir($Directory)
is_readable($fd)
is_executable($fic)
is_file($fic)
…
```

LECTURE RÉPERTOIRE

```
$directory = ".";
echo "<b>Parcours</b>: $Directory<br>\n";
if (is_dir($Directory) && is_readable($Directory)) {
if($fd= opendir($Directory)) {
 while($Entry = readdir($fd)) {
      echo"<b>Fichier</b>:$Directory/$Entry<br>\n";
 }
}
closedir($fd);
```

FONCTIONS MATHÉMATIQUES

Elles y sont toutes !

- `abs, cos, sin, tan, sqrt, exp, …`
- `pi( )`

CHAÎNES DE CARACTÈRES

`chaîne_result = addSlashes(chaîne)` ajoute un slash devant tous les caractères spéciaux.

`chaîne_result = chop(chaîne)` supprime les espaces blancs en fin de chaîne.

`strpos (chaîne, sous chaîne)` : retourne la position de la sous chaîne dans la chaîne. Dans le cas où la chaîne existe en plusieurs exemplaires, c'est la position de la première occurrence qui est retournée. strrpos retourne quand à elle la position de la dernière occurrence.

`strstr (chaîne, sous chaîne)` retourne la portion de la chaîne à partir de la première occurrence de la sous chaîne.

`strlen (chaîne)` : retourne la taille de la chaîne.

`strtolower|strtoupper (chaîne)` : retourne la chaîne passée en paramètres an minuscules (resp. majuscules).

`str_replace (car d'origine, car de destination, chaîne)` : remplace le caractère d'origine par le caractère de destination dans la chaîne.

`trim ( string str , string charlist )` supprime les caractères invisibles (espaces,\n, …) au début et à la fin de la chaîne.

`ereg(chaîne à chercher, chaîne) :` retourne vrai si la chaîne à chercher (sous forme de chaîne ou sous forme d'expression régulière) est contenue dans chaîne.

MAIL

Il existe une méthode PHP permettant d'envoyer un mail directement, sans appeler un quelconque gestionnaire de courrier.

La fonction *mail* (ou *email* parfois) permet de réaliser cela. Elle nécessite au moins trois paramètres :

- Le destinataire,
- L'objet du message,
- Le corps du message.

```
< ?

mail("dupont@mondomaine.fr", "Test mail","Contenu mail") ;

?>
```

Enverra un mail à *dupont@mondomaine.fr* avec comme sujet de mail « Test de la commande mail », et comme corps du mail : « Voici le corps du mail ».

FTP

Il est possible de se connecter et de s 'authentifier à un serveur FTP pour le transfert (download/upload) de fichiers.

L 'ensemble des fonctions de connexion, récupération de liste de fichiers, upload, download sont disponibles

- `ftp_connect, ftp_quit`
- `ftp_login`
- `ftp_get, ftp_put, ftp_size, etc.`

FTP

```
<?php
$conn_id = ftp_connect($ftp_server);

// Identification avec un nom d'utilisateur et un mot de passe
$login_result = ftp_login($conn_id, $ftp_user_name, $ftp_user_pass);
// Vérification de la connexion
if ((!$conn_id) || (!$login_result)) {
        echo "La connexion FTP a échoué !";
        exit;

}
// Chargement d'un fichier
$upload = ftp_put($conn_id, $destination_file, $source_file, FTP_BINARY);
// Vérification du status du chargement
if (!$upload) {
        echo "Le chargement FTP a échoué!";
    } else {
        echo "Chargement de $source_file vers $ftp_server en tant que
$destination_file";
    }
// Fermeture du flux FTP
ftp_close($conn_id);
?>
```

BASES DE DONNÉES

```
<?
    $bdd= "roose"; // Base de données
    $host= "kartxila.iutbayonne.univ-pau.fr";
    $user= "roose";  // Utilisateur
    $pass= "roose";

    $bdd = mysqli_connect($host, $user, $pass, $bdd))
?>
```

REQUÊTES SQL

```
$query =
 "SELECT num, pays, date, circuit FROM $nomtable ";
$result= mysqli_query($query); // mysqli_query
```

RÉCUPÉRATION RÉSULTATS

```
while($donnees = mysqli_fetch_assoc($resultat)){
$nb = mysqli_num_rows($req);
 print “Nb attributs = $nb<br>”;
 print $donnees[‘pays'];
 print $donnees[‘date'];
 print $donnees[‘circuit'];
}
mysqli_free_result($resultat);
mysqli_close($bdd);
```

PDO - PHP DATA OBJECTS

Permet de réaliser des opérations d'accès à une BDD (Base de Donnée Distante)

- sans avoir à se soucier du SGBD.
- nécessaire d'utiliser la programmation orientée objet
  - **interface PDO (*PHP Data Objects*).**

L'extension PDO n'est pas toujours active

- veiller à ce que l'option soit active dans le php.ini.
  - (dé-commenter la ligne ***extension=php_pdo.dll***).

EXCEPTIONS - PDO

L'utilisation de l'interface PDO peut nécessiter de gérer les exceptions (*mécanisme bien utile pour gérer les erreurs à l'exécution*). Les principales exceptions sont :

- *Erreur "could not find driver" :* PDO non activé, cf. php.ini
- *Erreur "Unknown MySQL server host":* Serveur non accessible, verifier l'URL
- *Erreur "Can't connect to MySQL server" :* Serveur accessible, mais soit y'a un soucis d'accès soit le SGBD est 'occupé'
- *Erreur "Unknown database <db_name>" :* Vérifier le nom de la BDD
- *Erreur "Access denied for user" :* Soucis d'authentification, verifier login/mdp

EXEMPLE

```
$PARAM_hote='localhost'; // le chemin vers le serveur
$PARAM_bdd='roose'; // le nom de votre base de données
$PARAM_user='roose'; // nom d'utilisateur pour se connecter
$PARAM_pw='roose'; // mot de passe de l'utilisateur pour se connecter
 try {
   $connexion = new PDO ('mysql:host='.$PARAM_hote.';dbname='.$PARAM_bdd, $PARAM_user, $PARAM_pw);
   $resultats=$connexion->query("SELECT * FROM bourse");
   $resultats->setFetchMode(PDO::FETCH_OBJ);
   while( $tuple = $resultats->fetch() ) {
      echo 'Utilisateur : '.$tuple->ville.'<br />';
   }
   $resultats->closeCursor();
 } // fin try
 catch(Exception $e){
   echo 'Erreur : '.$e->getMessage().'<br />';
}
```

EXEMPLE

Variante pour parcours résultats

```
// On met les objets dans un tableau
$createurs = $select->fetchAll(PDO::FETCH_OBJ);

while( $tuple = next($createurs) ) {
  echo '<h1>', $ tuple ->ville, ' ', $ tuple ->indice '</h1>';
}
```

MAJ BD

```
$resultats=$connexion->exec("UPDATE/DELETE/ … ");
print resultats ; // affiche le # d'enregistrements
                                                affectés
```

REQUÊTES PRÉPARÉES

Nouveauté avec PDO.

Possibilité de préparer des modèles paramétrés de requêtes,

- requêtes dont certaines valeurs ne seront connues que plus tard.

APPEL AVEC MARQUEURS

Ici, un exemple où les paramètres sont donné sans nom (en respectant l'ordre dans la requête)

```
…
$req = 'SELECT ville, indice FROM bourse WHERE ville = ? AND
indice = ?';
$req_prepare = $connexion->prepare($req);
$req_prepare ->execute(array('NY',10));
$req_prepare ->setFetchMode(PDO::FETCH_OBJ);

while ($tuples1 = $req_prepare->fetch()) {
   print "<BR>Ville : ".$tuples1->ville;
   print "<BR>Indice : ".$tuples1->indice;
…

     }
```

APPEL AVEC PARAMÈTRES NOMMÉS

```
$req = 'SELECT ville, indice FROM bourse WHERE ville = :nomville
AND indice = :valindice';

$req_prepare = $connexion->prepare($req);

$req_prepare ->execute(array('nomville' =>'NY','valindice' => 10));

$req_prepare ->setFetchMode(PDO::FETCH_OBJ);


while ($tuples1 = $req_prepare->fetch()) {

   print "<BR>Ville : ".$tuples1->ville;

   print "<BR>Indice : ".$tuples1->indice;

}
```

PHP ET LES IMAGES

Formats supportés

- JPG, GIF, PNG

Fonctionnalités

- Récupération,
- Création, Fusion
- Modification,
- Manipulation,

FONCTIONS LIÉES AUX IMAGES

Chargement d'images depuis des fichiers existants

- imagecreatefromgif (string filename)
- imagecreatefromjpeg (string filename)
- imagecreatefrompng (string filename)
- imagecreatefromstring (string filename)

// le type est automatiquement détecté

Création d'une image vierge

- imagecreate (int x_size , int y_size)
- imagecreatetruecolor (int x_size , int y_size)

FONCTIONS LIÉES AUX IMAGES

Redimensionnement

- int imagecopyresized (resource dst_image , resource src_image , int dst_x , int dst_y , int src_x , int src_y , int dst_w , int dst_h , int src_w , int src_h)

  imagecopyresized copie une partie rectangulaire d'une image dans une autre image.

- bool imagecopyresampled (resource dst_image , resource src_image , int dst_x , int dst_y , int src_x , int src_y , int dst_w , int dst_h , int src_w , int src_h)

- copie une zone rectangulaire de l'image src_im vers l'image dst_im .

- Durant la copie, la zone est rééchantillonnée de manière à conserver la clarté de l'image . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

FONCTIONS LIÉES AUX IMAGES

Dessiner sur une image

- imagechar (resource image , int font , int x , int y , string c , int color)
- Imagecharup (…)
- imagestring (resource image , int font , int x , int y , string c , int color)
- imagestringup (…)
- imagedashedline(image, x début, y début, x fin, y fin, couleur)
- imagefilledpolygon(image, tableau de points, nombre de points, couleur)
- imagefilledrectangle($image, x haut gauche, y haut gauche, x bas droite, y bas droite, couleur)

FONCTIONS LIÉES AUX IMAGES

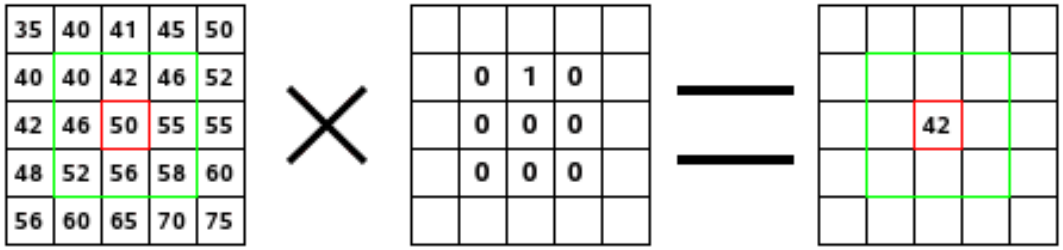
Couleurs

- imagecolorat(image, x, y)
- imagecolorstotal($image)
- imagecolortransparent(entier image, entier couleur)

FILTRES & MATRICES DE CONVOLUTIONS

PHP utilise des matrices de convolution de 3x3 qui suffisent pour la plupart des effets,

Le filtre étudie successivement chacun des pixels de l'image. Pour chaque pixel, que nous appellerons « pixel initial », il multiplie la valeur de ce pixel et de chacun des 8 pixels qui l'entourent par la valeur correspondante dans le noyau. Il additionne l'ensemble des résultats et le pixel initial prend alors la valeur du résultat final.



À gauche se trouve la matrice de l'image: chaque pixel est indiqué par sa valeur. Le pixel initial est encadré de rouge. La zone d'action du noyau est encadrée de vert. Au centre, se trouve le noyau et, à droite, le résultat de la convolution.

https://docs.gimp.org/2.6/fr/plug-in-convmatrix.html

FILTRES & MATRICES DE CONVOLUTIONS

imageconvolution($image, $matrix, $div, $offset)

Box Blur (flou)
$box_blur = array([1, 1, 1], [1, 1, 1], [1, 1, 1]); // Matrice de 3x3
ou $box_blur = array(array(1, 1, 1), array(1, 1, 1), array(1, 1, 1);
imageconvolution($im_php, $box_blur, **9**, 0); // 9 => diviseur du résultat de la convolution; 0 = offset (décalage) de la couleur => permet de jouer sur les couleurs, ombres par exemple.

Le 'Flou fonctionne en 'moyennant l'ensemble des pixels autour.

Sharpen (netteté)
$sharpen = array([0, -1, 0], [-1, 5, -1], [0, -1, 0]);
imageconvolution($im_php, $sharpen, **1**, 0);

La netteté fonctionne en exagérant les differences entre les pixels et ses voisins. Cela rend les bord un peu plus clairs. Dans le cas du 'sharpe' le diviseur est à un car la somme des éléments dans les 3 arrays est égal à 1.

FILTRES & MATRICES DE CONVOLUTIONS

Edge Detect (contours)

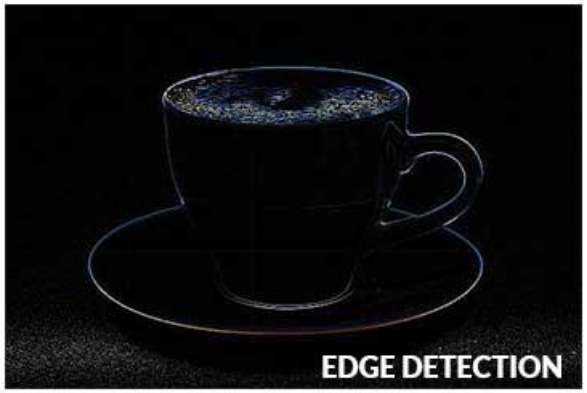
```
$edge_detect = array([-1, -1, -1], [-1, 8, -1], [-1, -1, -1]);

imageconvolution($im_php, $edge_detect, 1, 0);

imageconvolution($im_php, $edge_detect, 1, 255);
```

La détection des contours est similaire la netteté, mais l'effet est encore plus fort. Avec la détection des contours, la somme de tous les éléments du tableau est 0. Cela signifie que l'image que vous obtiendrez sera généralement noire, sauf en cas de brusque changement de couleur, ce qui se produit généralement aux bords des objets. Vous pouvez passer du blanc à l'image noire en définissant le paramètre offset sur 255.


BOX BLUR


SHARPEN


EMBOSS


EDGE DETECTION


EDGE DETECTION

FONCTIONS LIÉES AUX IMAGES

- image[jpg|gif|png] (id image) -> le navigateur.
- imagedestroy(identifiant image) -> suppression fic. temp.

CREATION SIMPLE

```
<?php
    header("Content-type: image/png");
    $nomimage="kalimucho.png";
    $im = imagecreatefrompng($nomimage);
    $red = imagecolorallocate($im, 255, 0, 0);
    imagestring($im, 5, 20, 20, "We did it !", $red);
    imagepng($im); // permet aussi d'enregistrer l'image sur le serveur; de définir un degré de compression
    imagedestroy($im);
?>
```

COPIE DE MORCEAUX D'IMAGES

bool **imagecopy** ($dst_im , $src_im , $dst_x , $dst_y , $src_x , $src_y , $src_w , $src_h)

- dst_im : lien vers la ressource cible de l'image.
- src_im : lien vers la ressource source de l'image.
- dst_x : coordonnées du point de destination.
- dst_y : coordonnées du point de destination.
- src_x : coordonnées du point source.
- src_y : coordonnées du point source.
- src_w: largeur de la source.
- src_h : hauteur de la source.

COPIE DE MORCEAUX D'IMAGES

```
<?php
  header("Content-type: image/png");
  $nomimage="kalimucho.png";
  $src = imagecreatefrompng($nomimage);
  $dest = imagecreatetruecolor(80, 40);
  // Copie
  imagecopy($dest, $src, 0, 0, 20, 13, 80, 40);

  // Affichage et libération de la mémoire
  imagepng($dest, « ../kalimucho_tmp.png"); // sauvegarde sous forme de fichier
  imagedestroy($dest);
  imagedestroy($src);
?>
```

EXEMPLE DE FUSION D'IMAGE

```php
header('Content-type: image/jpeg');

$file_name = $_GET["photo"];
$file_signature="signature.png";

list($largeur, $hauteur)= getimagesize("$file_name"); //dimenseions de l'image originale
list($largeurS, $hauteurS) = getimagesize("$file_signature"); //dimenseions de l'image originale

$image = imagecreatetruecolor($largeur,$hauteur);
$source = imagecreatefromjpeg("$file_name");
$signature=imagecreatefrompng("$file_signature");

imagecopy($source, $signature, $largeur - $largeurS, $hauteur - $hauteurS, 0, 0, $largeur, $hauteur);

imagejpeg($source);
imagedestroy($source);
```

GESTION DE SESSIONS : COOKIES

`setcookie (nom, valeur, expiration, chemin, domaine, securite)` permet de créer un cookie sur le poste client.

Le premier champs nom est obligatoire et défini son nom.

Si l'on souhaite lui associer une valeur on renseignera le second champs.

Le troisième permet de spécifier une date d'expiration;dans le cas où rien n'est précisé, le cookie devient permanent.

Les champs chemin et domaine permettent de préciser les URL et domaines auxquels sont associés le cookie. Enfin le dernier, sécurité permet de n'envoyer le cookie que si la connexion est réalisée via le protocole sécurisé HTTPS (TRUE).

GESTION DE SESSIONS : COOKIES

Une fois le cookie crée, il est possible de le récupérer dans les pages suivantes via le tableau $_COOKIE['nom cookie'].

```
setcookie("TestCookie", $value, time()+3600);
```
// expire dans une heure

La suppression d'un cookie se fait en le recréant...mais avec une date d'expiration passée !

```
setcookie ("TestCookie", "", time() - 3600);
```
// a expiré y'a une heure

Les cookies ne sont malheureusement pas supportés par l'ensemble des navigateurs pour des raisons de confidentialité. Aussi, il est nécessaire de mettre en oeuvre d'autres mécanismes afin de gérer les sessions.

SESSIONS

- La transmission de données d'une page à l'autre est peut se faire avec
  - les divers champs des formulaires(hidden ou non.)
  - passer les variables directement à travers les liens (GET)
  - via des cookies.
  - ou via **les sessions.**
- En gros = mécanisme permettant de mettre en relation les différentes requêtes du même client sur une période de temps donnée.

- Chaque visiteur en se connectant à un site reçoit un numéro d'identification dénommé identifiant de session (SID)

- La fonction `session_start()` se charge de générer automatiquement cet identifiant unique de session et de créer un répertoire. Elle doit être placée au début de chaque page afin de démarrer ou de continuer une session.

```php
<?php
  session_start();
  $Session_ID = session_id();
  // $Session_ID = 7edf48ca359ee24dbc5b3f6ed2557e90
?>
```

SESSIONS

- Un répertoire est créé sur le serveur à l'emplacement désigné par le fichier de configuration php.ini, afin de recueillir les données de la nouvelle session.

```
[Session]
session.save_path= C:\PHP\sessiondata
; Rép session = \sess_7edf48ca359ee24dbc5b3f6ed2557e90
```

- Pour utiliser et créer une variable de session il suffit de l'utiliser via `$_SESSION["nom_variable"]`
- Supprimer une variable : unset($_SESSION['variable'])
- Supprimer toutes les variables : session_unset()
- Supprimer la session : session_destroy()

SESSIONS

Fonction	Signification
session_start	Démarre une session
session_id	Retourne l'id de la session en cours
session_name	Retourne le nom de la session en cours
session_unset	Detruit toutes les variables de la session en cours
session_destroy	Destruit la session en cours

XML

```
<?xml version="1.0" encoding="UTF-8" ?>
<messages>
 <message>
        <id>1</id>
        <user_id>1</user_id>
        <title>Bonjour</title>
        <body>Un bonjour de Paris </body>
 </message>
 <message>
 …
 </message>
</messages>
```

XML

```
//lecture fichier XML et conversion en objet simpleXML
$messages = simplexml_load_file('messages.xml');
foreach($messages as $message) { //parcours du XML comme d'un tableau
 $id = $message->id;
 $body = $message->body;
 print ''$body <br/>'';
}
```

1 - Un bonjour de Paris

DIVERS…ENCODAGE CARACTÈRES

UTF-8 -> ISO

- echo htmlentities($ lachaine, ENT_QUOTES, 'iso-8859-1');

ISO -> UTF-8

- echo htmlentities($lachaine, ENT_QUOTES, 'utf-8') ;

ENT_COMPAT	Convertit les guillemets doubles, et ignore les guillemets simples.
ENT_QUOTES	Convertit les guillemets doubles et les guillemets simples.
ENT_NOQUOTES	Ignore les guillemets doubles et les guillemets simples.
ENT_IGNORE	Ignore les séquences de caractères invalides plutôt que de retourner une chaine vide. L'utilisation de ce drapeau est fortement déconseillée pour des » raisons de sécurité.
ENT_SUBSTITUTE	Remplace les séquences de code invalide avec un caractère de remplacement Unicode U+FFFD (UTF-8) ou &#FFFD; (sinon) au lieu de retourner une chaîne vide.
ENT_DISALLOWED	Remplace les points du code invalides du document fourni avec un caractère de remplacement Unicode U+FFFD (UTF-8) ou &#FFFD; (sinon) au lieu de le laisser tel quel. Ceci peut être utile pour, par exemple, s'assurer du bon formatage de documents XML contenant du contenu externe.
ENT_HTML401	Gère le code comme étant du HTML 4.01.
ENT_XML1	Gère le code comme étant du XML 1.
ENT_XHTML	Gère le code comme étant du XHTML.
ENT_HTML5	Gère le code comme étant du HTML 5.

CONFIGURATION - PHP.INI

Le comportement de PHP est dicté par sa configuration, établie dans le fichier **php.ini**

EVOLUTIONS DE PHP

PHP/FI (Personal Home Page / Form Interpreter) fût la première version officielle de PHP (tout est parti d'une librairie Perl à l'origine).

PHP3 résulte d'une réécriture complète de PHP/FI.

PHP3 et PHP/FI ne sont plus officiellement supportés. Néanmoins, ils persistent encore en particulier chez certains hébergeurs.

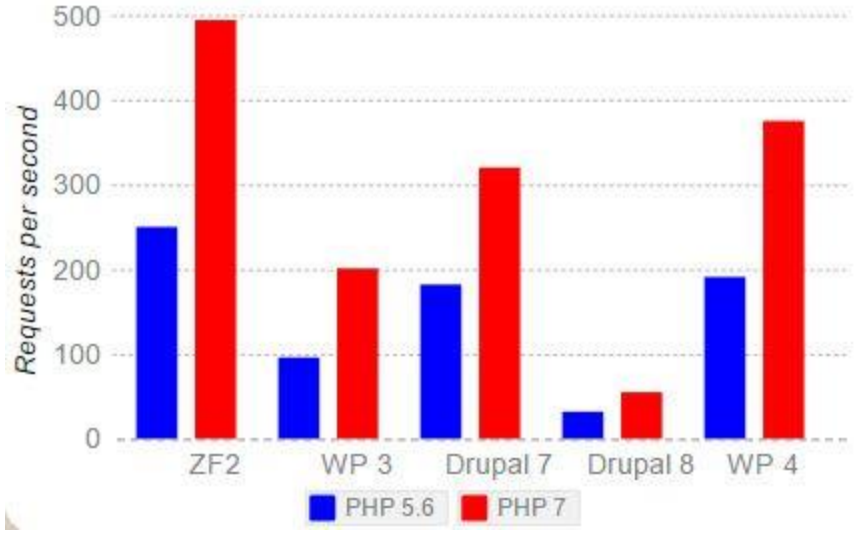
Quant à PHP4, il résulte à son tour d'une réécriture de PHP3 et utilise le moteur Zend. C'est la branche courante de PHP.

PHP 5 : Exceptions (try/catch), JSON, PDO, …

PHP 6 : y'en a pas !

Aujourd'hui : PHP7

PERFORMANCES



PERFORMANCES

Migrer à PHP 7 est un geste écologique et facile.

Cette version intègre un compilateur JIT qui réduit de 25 % à 70 % la puissance informatique nécessaire telle que la quantité de mémoire vive, le nombre de cycle CPU, etc.

Selon Rasmus, PHP motorise environ 50 % des 2 milliards des sites web mondiaux qui sont hébergés sur 10 millions de serveurs physiques.

Au taux d'adoption actuel de 5 %, PHP7 a déjà permis d'économiser
- 200 millions de dollars ;
- 750 millions de kWh d'électricité; (équivalent à la production de 7 barrages électriques = conso moyenne de 70 0000 maisons en France pendant 1 an)

- 750375 millions de kg de gaz à effet de serre.

Avec un taux d'adoption de 100 %, on atteindrait des chiffres énormes :
- 4 milliards de dollars ;
- 15 milliards de kWh d'électricité (2,5 millions de français) ;
- 7,5 milliards de kg de gaz à effet de serre (1 million de français).

ERREURS CLASSIQUES

Oubli des balises **<?php … ?>**

Oubli du **$**

Nom de variable erroné
- => contenu 'vide'

Droits sur les fichiers, erreur de chemin d'accès

Droits sur la BD et/ou login/mp faux

Mauvais fichier mis à jour…pas celui testé

Double clic sur le nom du fichier pour l'exécuter => ne passe pas par le serveur.
- ***file:***//...../*monfichier.php*
- ***http:***//*iparla. … ./monfichier.php*

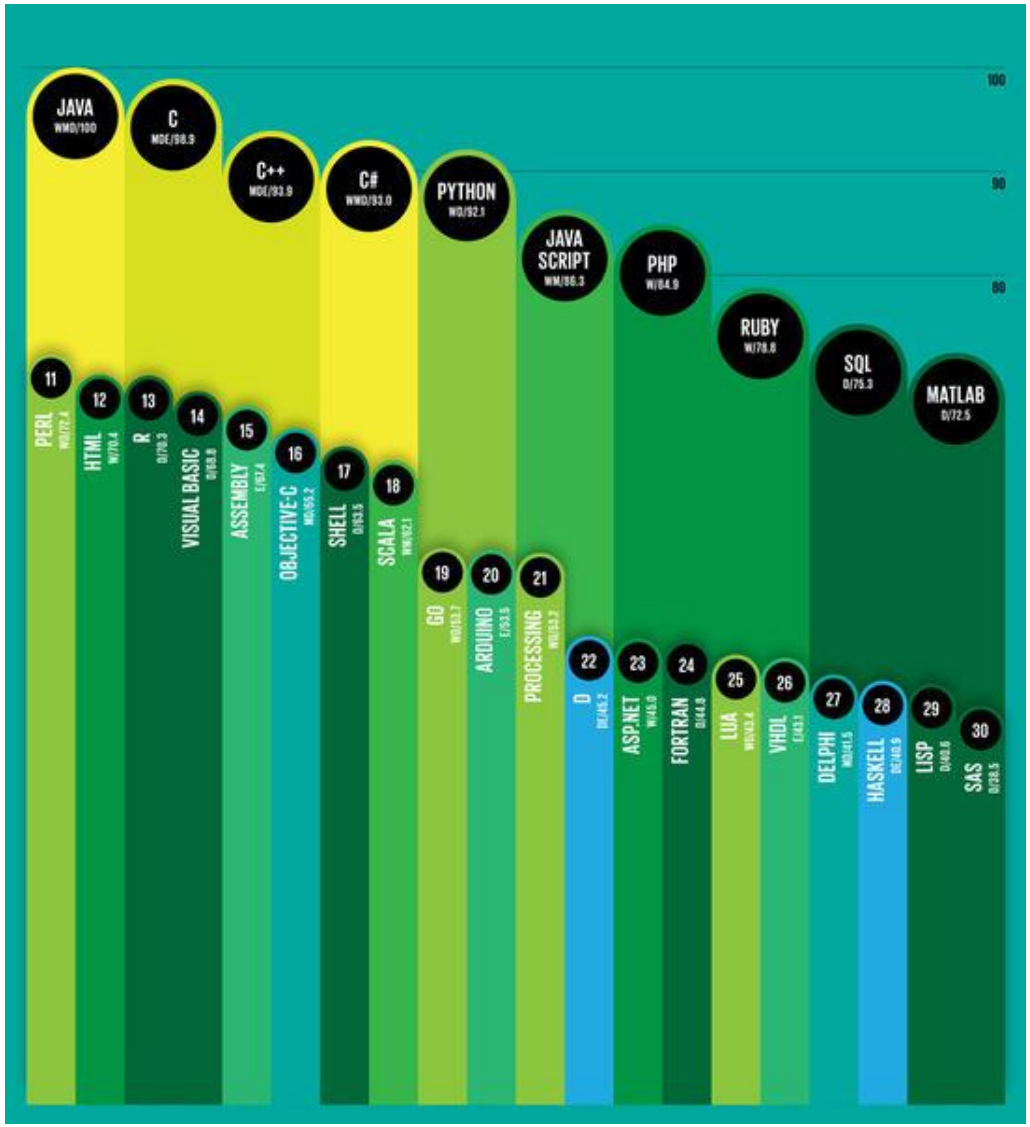
Mauvais header / header « image » pas spécifié
- *Hiéroglyphes à l'écran !*

SITES POPULAIRES AVEC PHP

Facebook.com

Wikipedia.org

Wordpress.com


JAVA
WMD/100
C
MDE/98.9
C++
MDE/93.9
C#
WMD/93.0
PYTHON
WD/92.1
JAVA SCRIPT
WM/86.3
PHP
W/84.9
RUBY
W/79.8
SQL
D/75.3
MATLAB
D/72.5
100
90
80
11 PERL WD/72.4
12 HTML W/70.4
13 R D/70.3
14 VISUAL BASIC D/69.8
15 ASSEMBLY E/67.4
16 OBJECTIVE-C MD/65.2
17 SHELL D/63.5
18 SCALA WM/62.1
19 GO WD/63.7
20 ARDUINO E/63.5
21 PROCESSING WD/63.2
22 D DE/45.2
23 ASP.NET W/45.0
24 FORTRAN D/44.8
25 LUA WD/43.4
26 VHDL E/43.1
27 DELPHI WD/41.5
28 HASKELL DE/40.9
29 LISP D/40.6
30 SAS D/28.5