HELHa
Haute École Louvain en Hainaut

Syllabus

Catégorie Économique

Informatique de gestion

ECIG1B10IGDE2f
Concepts fondamentaux de technologie Internet
1IG

Année académique 2014 - 2015

PHP- INTRODUCTION

1. Les sites dynamiques
2. Qu'est-ce que PHP?
3. Le rôle du PHP
4. L'environnement de travail
5. Création d'un projet
6. Le code PHP
7. Inclusion de fichiers externes
8. Ajout de commentaires
9. Séparateur d'instructions – bloc d'instructions
10. Minuscules-majuscules

PHP- INTRODUCTION

1. Les sites dynamiques

Afin de comprendre l'existence et le fonctionnement de PHP, nous allons prendre l'**exemple du restaurant**:

- il y a des clients et des serveurs;
- le client passe commande;
- le serveur apporte la commande du client.

Et en PHP?

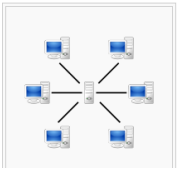
- le client est la combinaison des codes HTML, CSS, JS, JQuery et XML;
- le serveur est le code PHP;
- le PHP répond aux attentes du client.

PHP- INTRODUCTION

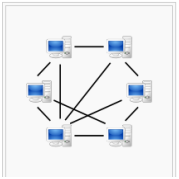
Attention!!!

Dans le monde des réseaux, nous voyons apparaître **deux architectures** possibles basées sur l'exemple précédent:

- l'architecture client/serveur;



- l'architecture peer to peer.



PHP- INTRODUCTION

Cette couche réseau nous permet de créer un nouveau type de site: le **site dynamique**.

	Site Statique	Site dynamique
Langages utilisés	HTML/CSS, XML, JS, Jquery …	PHP, ASP.Net, JSP …
Serveurs utilisés		Apache, IIS, Tomcat + MYSQL, SQL Server …
Modification	Manuelle (Webmaster)	Automatique
Type de site	Vitrine (Présentation)	E-commerce, E-learning …

Le site dynamique utilise un **serveur WEB** et (éventuellement) un **serveur de base de données**.

PHP- INTRODUCTION

Le **serveur Web** est:

- d'un point de vue matériel, un ordinateur;
- d'un point de vue logiciel, un **serveur HTTP** (HyperText Transfer Protocol) installé sur un OS, HTTP étant le principal protocole de communication employé par le World Wide Web via le port 80.

Le **serveur de base de données** permet d'accéder à une base de données qui stocke et retrouve un ensemble d'informations de plusieurs natures ainsi que les liens qui existent entre les différentes informations.

Un site Internet sera consulter grâce au protocole HTTP mais sera garni sur un hébergeur par le **protocole FTP** (File Transfer Protocol) avec le port 21.

PHP- INTRODUCTION

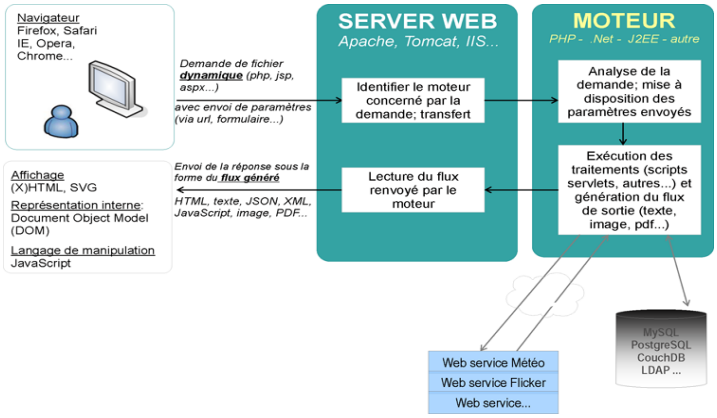
Revenons au fonctionnement de notre site dynamique et voyons comment le client et le serveur communiquent:

- Le client demande au serveur à voir une page web;
- Le serveur prépare la page spécialement pour le client;
- Le serveur lui envoie la page qu'il vient de générer.

La page web est générée à chaque fois qu'un client la réclame. C'est précisément ce qui rend les sites dynamiques vivants: le contenu d'une même page peut changer d'un instant à l'autre.

PHP- INTRODUCTION

En analysant de plus près le fonctionnement de notre serveur, nous observons ceci:



PHP- INTRODUCTION

2. Qu'est-ce que PHP?

PHP (*PHP: Hypertext Preprocessor*) est :
- un langage de programmation.
- destiné à être interprété sur un serveur WEB
- offre la possibilité de développer des sites WEB dynamiques

Ce langage permet d'insérer des instructions de programmation puissantes dans des pages de type HTML.

Une page dynamique PHP est un document HTML envoyé par le serveur vers le poste client.

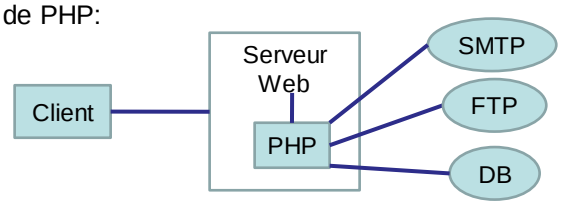
PHP- INTRODUCTION

3. Le rôle du PHP

Le langage PHP possède les mêmes fonctionnalités que les autres langages (C#/VB.Net, Java …):

- collecte et analyse de données
- génération dynamique de pages Web
- envoi et réception de cookies
- gestion de sessions…

Rôles de PHP:



PHP- INTRODUCTION

4. L'environnement de travail

Chez vous:

Pour que votre ordinateur puisse lire du PHP, il faut qu'il se comporte comme un serveur. Il suffit simplement d'installer les mêmes programmes que ceux que l'on trouve sur les serveurs qui délivrent les sites web aux internautes.

La combinaison Apache + PHP + MySQL est la plus courante sur les serveurs web, à tel point qu'on a créé des « packs » tout prêts qui contiennent tous ces éléments.

Commencez par télécharger WAMP sur son site web officiel et suivez les étapes d'installation.

PHP- INTRODUCTION

- **Apache**: c'est un serveur web. Il s'agit du plus important de tous les programmes, car c'est lui qui est chargé de délivrer les pages web aux visiteurs.

Cependant, Apache ne gère que les sites web statiques. Il faut donc le compléter avec d'autres programmes.

- **PHP**: c'est **un plug-in** pour Apache qui le rend capable de traiter des pages web dynamiques en PHP. En clair, en combinant Apache et PHP, notre ordinateur sera capable de lire des pages web en PHP.

- **MySQL**: c'est le logiciel de gestion de bases de données. Il permet d'enregistrer des données de manière organisée.

PHP- INTRODUCTION

En Labo:

Lancer le serveur Wamp (serveur PHP-MySql)
L'icône du serveur passe de l'orange au vert lorsque le Wamp est lancé.

Pour publication:

Trouver un hébergeur (gratuit ou payant) et charger votre site en FTP sur cet hébergeur.

PHP- INTRODUCTION

5. Création d'un projet

- Créer un répertoire pour votre projet dans le dossier ***c:/wamp/www;***
- Créer les scripts PHP à l'aide d'un éditeur (Notepad ou Dreamweaver);
- Sauvegarder les scripts dans le répertoire que vous avez créé préalablement.

- Exécuter les scripts dans un navigateur (Firefox, Internet Explorer, Opéra …) soit en tapant l'adresse de votre projet en utilisant le **numéro de port 8080** (80 chez vous) ou soit à partir de l'icône wamp dans la zone de notification de votre barre des tâches.

Ex: http://localhost:8080/MonProjet/index.php.

PHP- INTRODUCTION

A partir de l'icône Wamp, vous devez:

- cliquer sur l'icône du Wamp
- affichage du menu ci-dessous

PHP- INTRODUCTION

Une page web (ci-dessous) devrait s'ouvrir dans votre navigateur favori (Firefox, par exemple). Si la page s'affiche chez vous, cela signifie qu'Apache fonctionne.

PHP- INTRODUCTION

Dans la rubrique « **vos projets** », les répertoires créés dans ***c:/wamp/www*** apparaissent. Lorsque vous cliquez sur un de ces répertoires, la fenêtre suivante apparaît avec la liste des scripts PHP contenus dans le répertoire sélectionné (si aucun s'appelle index.php):

PHP- INTRODUCTION

Vous êtes en train de simuler le fonctionnement d'un serveur web sur votre propre machine. Et donc, vous êtes le seul internaute à pouvoir y accéder. On dit que vous travaillez « en local ».

L'URL affichée par le navigateur dans la barre d'adresse **est http://localhost/**, ce qui signifie que vous naviguez sur un site web situé sur votre propre ordinateur.

Sauvegarde permanente du projet au laboratoire

- Créer un répertoire PHP sur W:
- Copier le répertoire créé dans ***c:/wamp/www*** dans le répertoire PHP créé ci-dessus.
- Réaliser cette opération de copie plusieurs fois par séance de laboratoire ou appliquer vos connaissances du MS-DOS ☺.

Pour rappel, le contenu du disque C: est perdu chaque fois que la machine virtuelle est arrêtée.

PHP- INTRODUCTION

6. Le code PHP

Le code PHP est identifié par la balise suivante **<?php** mon code **?>**.

Peut-on placer une balise PHP n'importe où dans le code?

Oui! Vraiment n'importe où. Pas seulement dans le corps de la page d'ailleurs: vous pouvez placer une balise PHP dans l'en-tête de la page.

PHP- INTRODUCTION

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/x,!
html1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
<head>
<title>Notre première instruction : echo</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
</head>
<body>
<h2>Affichage de texte avec PHP</h2>
<p>
Cette ligne a été écrite entièrement en HTML.<br />
<?php echo "Celle-ci a été écrite entièrement en PHP."; ?>
</p>
</body>
</html>
```

Nous remarquons que l'instruction echo se termine par un **;**.

PHP- INTRODUCTION

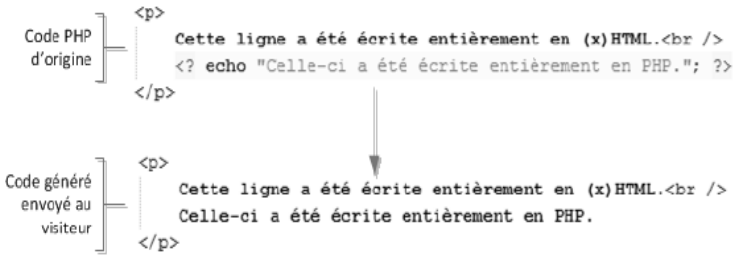
Comment PHP génère du code HTML ?

Le code PHP est exécuté en premier et l'ordinateur fait ce qu'on lui demande.

Ici on lui a dit « Affiche ce texte ici ».

Une fois toutes les instructions PHP exécutées , la page qui sort est une page qui ne contient que du HTML! C'est cette page de «résultat» qui est envoyée au visiteur, car celui-ci ne sait lire que le HTML.

PHP- INTRODUCTION



PHP- INTRODUCTION

7. Inclure des fichiers externes

Cela permet de moduler le code, de créer une bibliothèque de fonctions d'utilisation courante ou encore mettre en place le MVC.

Les fichiers de code php peuvent avoir l'une des extensions suivantes:

.php .inc .inc.php

Fonction	Description
`include("nom_fichier.ext")`	Lors de son interprétation par le serveur, cette ligne est remplacée par tout le contenu du fichier précisé en paramètre, dont vous fournissez le nom et éventuellement l'adresse complète. En cas d'erreur, par exemple, si le fichier n'est pas trouvé, `include()` ne génère qu'une alerte, et le script continue.
`require("nom_fichier.ext")`	A désormais un comportement identique à `include()`, à la différence près qu'en cas d'erreur, `require()` provoque une erreur fatale et met fin au script.
`include_once("nom_fichier.ext")` `require_once("nom_fichier.ext")`	Contrairement aux deux précédentes, ces fonctions ne sont pas exécutées plusieurs fois, même si elles figurent dans une boucle ou si elles ont déjà été exécutées une fois dans le code qui précède.

PHP- INTRODUCTION

8. Ajout de commentaires

```
// ceci est un commentaire

/* commentaire
 sur plusieurs lignes */

# commentaire
```

9. Séparateur d'instructions – bloc d'instructions

Séparateur ; { bloc }

10. minuscules-majuscules

PHP est case sensitive et fait la distinction entre les caractères en minuscules et en majuscules.

PHP- INTRODUCTION

```
<html>
<head>
<title> Premier programme PHP</title>
</head>
<body>
<?php
   $name="Paul";
   $Name="Pierre";
   echo   " hello $name,<br/> " ;
   echo    " où est $Name ? " ;
?>
</body>
</html>
```

PHP- CONCEPTS DE BASE

1. Les données
2. L'instruction « echo »
3. Les instructions conditionnelles
4. Les instructions de boucle
5. Les tableaux et la boucle foreach
6. La gestion des erreurs

PHP- Les données

1.1 Les données

Une variable est une information qui reste stockée en mémoire le temps de la génération de la page PHP. Elle a un nom et une valeur.

Les noms de variables :
- commencent par $ suivi de lettres min-maj, chiffres, _
- sont sensibles à la casse
- ne sont pas déclarées (son type est déterminé par sa valeur)

- On peut créer des variables n'importe où. La déclaration des variables n'est pas obligatoire.
- L'initialisation des variables n'est pas obligatoire.
- La portée d'une variable est GLOBALE (elle concerne tout le script à partir de sa déclaration)

```
$a = 1;
$b = $a + 5;
$c = "hello";
$c = $b = $a = 1;
```

PHP- Les données

1.1 Les données

```
<html>
<head>
<title> e2_1.PHP</title>
</head>
<body>
<?php
   $name="Paul";
   $Name="Pierre";
   echo  " hello $name,<br/>" ;
   echo   " où est $Name ? " ;
?>
</body>
</html>
```

```
<html>
<head>
<title> e2_2.PHP</title>
</head>
<body>
<?php
   $color1 = "#ff0000";
   $color2 = "#00ff00";
   $color3 = "#0000ff";
   echo "<font
        color=\"$color1\">red</font><br/>";
   echo "<font
        color=\"$color2\">green</font><br/>";
   echo "<font
        color=\"$color3\">blue</font><br/>";?>
</body>
</html>
```


PHP- Les données

1.2 Les types de données

int : codés sur 32 bits, allant de -2^{31} à $+2^{31}-1$

float : précision de 14 chiffres (affichage sous forme Décimale si le nombre a moins de 15 chiffres, sinon sous forme exponentielle

bool : vaut TRUE ou FALSE

Type de données	Exemple de valeur
string	"Du texte"
int	42
float	14.738
bool	true false
NULL	

```
$a = 80;
$b= ($a<95);
echo $b
```

Affiche 1

```
$bool= TRUE;
$bool= FALSE;
```

```
$a=15;
If ($a)…
```

True si $a existe et a une valeur !=0

PHP- Les données

1.2 Les types de données

string : chaîne de caractères entre ‘ ’ ou " "

```php
<?php
$nom_du_visiteur = "Mateo21";
$nom_du_visiteur = 'Mateo21';
?>
```

```php
<?php
$variable = "Mon \"nom\" est Mateo21";
$variable = 'Je m\'appelle Mateo21';
?>
```

```php
<?php
$variable = 'Mon "nom" est Mateo21';
$variable = "Je m'appelle Mateo21";
?>
```

Type de données	Exemple de valeur
string	"Du texte"
int	42
float	14.738
bool	true false
NULL	

PHP- Les données

1.2 Les types de données

déterminer le type :

string gettype ($mavar)

vérifier le type d’une variable :

is_int ($var) , is_float( ), is_string ( ), is_bool ( ), is_numeric ( ), is_array ( ), ...

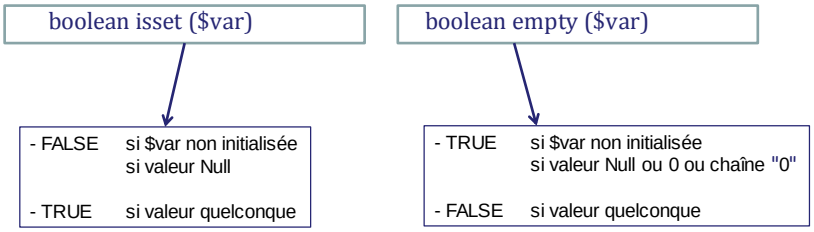
convertir le type d’une variable :

$result = (type-desire) $var

PHP- LES DONNEES

1.2 Les types de données

Contrôler l'état d'une variable :



PHP- LES DONNEES

1.3 Les variables prédéfinies

les variables prédéfinies contiennent des infos sur le serveur et sur les données qui peuvent transiter entre le poste client et le serveur.

$GLOBALS
$_COOKIE
$_ENV
$_FILES
$_GET
$_POST
$_REQUEST
$_SERVER
$_SESSION

PHP- Les données

1.4 Les constantes personnalisées

Sensible à la casse
ou non

```
// définition
define ( "PI", 3.14159, TRUE);
//utilisation
echo "la constante PI vaut " , PI, "</br>« ;
```

non précédée de $,
donc la séparer avec ,
de ce qui précède dans echo

PHP- Les données

1.5 Les opérateurs mathématiques, logiques, de comparaison

opérateurs	Signification
+, ++	Addition
-, - -	Soustraction
*	Multiplication
/	Division
%	modulo

OR ||
XOR
AND &&
!

+= -= *=
/= %= .=

== != ou < >
< <= > >=
=== !==

Teste l'identité
- des valeurs
- des types

PHP- Les données

1.6 Les fonctions mathématiques

double/integer `abs` (double/integer X)	Valeur absolue de X : `echo abs(-543);` //affiche 543.
double `acos` (double X)	Arc cosinus de X, qui doit être compris entre – 1 et + 1. Le résultat est en radians : `echo acos(0.5);` // affiche 1.0471975511966.
double `acosh` (double X)	Arc cosinus hyperbolique de X. Ne fonctionne pas sous Windows.
double `asin` (double X)	Arc sinus de X, qui doit être compris entre – 1 et + 1. Le résultat est en radians : `echo asin(0.5);` // affiche 0.5235987755983.
double `asinh` (double X)	Arc sinus hyperbolique de X. Ne fonctionne pas sous Windows.
double `atan` (double X)	Arc tangente de X. Le résultat est en radians : `echo atan(5);` // affiche 0.46364760900081.
double `atan2` (double Y, double X)	Arc tangente du rapport Y/X. Le résultat est en radians. Il faut que Y soit différent de 0.
double `atanh` (double X)	Arc tangente hyperbolique de X.
string `base_convert` (string N, integer B1, integer B2)	Convertit le nombre *N* contenu dans une chaîne de la base B1 dans la base B2.
integer `bindec` (string X)	Convertit un nombre binaire X contenu dans une chaîne en base 10.

PHP- Les données

double `ceil` (double X)	Retourne l'entier immédiatement supérieur à X.
double `cos` (double X)	Cosinus de X qui doit être exprimé en radians.
double `cosh` (double X)	Cosinus hyperbolique de X.
string `decbin` (integer X)	Convertit X de la base 10 en binaire.
string `dechex` (integer X)	Convertit X de la base 10 en hexadécimal.
string `decoct` (integer X)	Convertit X de la base 10 en octal.
double `deg2rad` (double X)	Convertit X de degrés en radians.
double `exp` (double X)	Exponentielle de X, soit e^x
double `expm1` (double X)	Retourne l'exponentielle de X – 1, soit ex –1.
double `floor` (double X)	Retourne la partie entière de X, soit l'entier immédiatement inférieur à X.
double `fmod` (double X, double Y)	Retourne le reste de la division de Y par X pour des opérandes de type double.
integer `getrandmax` (void)	Indique la valeur maximale retournée par la fonction `rand()`.
integer `hexdec` (string CH)	Convertit la chaîne hexadécimale CH en décimal.
double `hypot` (double X, double Y)	Retourne la valeur de l'hypoténuse d'un triangle rectangle dont les côtés de l'angle droit sont X et Y, donc la valeur de la racine carrée de $(X^2 + Y^2)$.
boolean `is_finite` (double X)	Retourne `TRUE` si la valeur X est finie, c'est-à-dire dans l'intervalle des valeurs admises pour un double, et `FALSE` dans le cas contraire.
boolean `is_infinite` (double X)	Retourne `TRUE` si la valeur X est supérieure à la valeur maximale admise pour un double, et `FALSE` dans le cas contraire.
boolean `is_nan` (double X)	Retourne `TRUE` si la valeur X n'est pas un nombre, et `FALSE` dans le cas contraire.

PHP- Les données

double `lcg_value` (void)	Retourne un nombre aléatoire compris entre 0 et 1.
double `log` (double X, double B)	Logarithme népérien (de base e) du nombre X
double `log10` (double X)	Logarithme décimal (de base 10) de X
double `log1p` (double X)	Logarithme népérien de (1 + X)
double/integer `max` (double/integer X, double/integer Y)	Retourne la valeur maximale de X et de Y.
double/integer `min` (double/integer X, double/integer Y)	Retourne la valeur minimale de X et de Y.
integer `mt_getrandmax` (void)	Retourne la plus grande valeur aléatoire que peut retourner la fonction `mt_rand()`.
integer `mt_rand` (integer Min, integer Max)	Génère un résultat compris entre Min et Max ou entre 0 et `rand_nombre aléatoiremax` si vous omettez les paramètres.
void `mt_srand` (integer N)	Initialise le générateur de nombres aléatoires pour la fonction `mt_rand()`. Le paramètre *N* est un entier quelconque.
integer `octdec` (string CH)	Convertit un nombre octal contenu dans la chaîne CH en base 10.
double `pi` (void)	Retourne la valeur de `pi`.

PHP- l'instruction echo

2.1 afficher du texte et des variables avec echo

on peut afficher du texte avec 'echo'.
La chaîne de caractères est mise entre " "

```
1 <?php echo "Ceci est du texte"; ?>
```

Le mot « texte » sera affiché en gras grâce à la présence des balises et .

```
1 <?php echo "Ceci est du <strong>texte</strong>"; ?>
```

On peut aussi s'en servir pour afficher la valeur d'une variable.

```
1 <?php
2 $age_du_visiteur = 17;
3 echo $age_du_visiteur;
4 ?>
```

PHP- l'instruction echo

2,2 concaténer du texte et des variables avec echo

on peut afficher du texte et des variables avec echo et les . " "

```php
<?php
$age_du_visiteur = 17;
echo "Le visiteur a $age_du_visiteur ans";
?>
```

Avec les ' ' , cela ne marche pas sans concaténation.

```php
<?php
$age_du_visiteur = 17;
echo 'Le visiteur a $age_du_visiteur ans'; // Ne marche pas
?>
```

Le symbole de concaténation est le **.**

```php
<?php
$age_du_visiteur = 17;
echo 'Le visiteur a ' . $age_du_visiteur . ' ans';
?>
```


PHP- l'instruction echo

2,2 concaténer du texte et des variables avec echo

Conclusion : effet des guillemets simples et doubles

' '

- Variables non substituées
- car \ sans effet

" "

- Variables remplacées par leur valeur
- car \ permet de protéger un car spécial (échappement)

\' affiche '
\ " affiche "
\ $ affiche $
\\ affiche \
\ n saut de ligne
\r retour chariot
\t tabulation horizontale

PHP- les instructions conditionnelles

3.1 if

```
if (condition) instruction;
```

```
if (condition)
{
        // bloc
}
```

```
if (condition)
    {
       // bloc
    }
else
    {
       // bloc
    }
```


PHP- les instructions conditionnelles

```
<?php  // e3_1.php
   $prix = 35;
   if ($prix > 100)
      {
       echo " pour un achat de $prix &euro , la remise est de
      <b>15%</b><br/>";
       $pnet = $prix * 0.85;
       echo " le prix net est de $pnet";
      }
elseif ($prix > 50)
      {
       echo " pour un achat de $prix &euro , la remise est de
      <b>10%</b><br/>";
       $pnet = $prix * 0.90;
       echo " le prix net est de $pnet";
      }
```

PHP- les instructions conditionnelles

```
else
      {
       echo " pour un achat de $prix &euro , la remise est de
      <b>O5%</b><br/>";
       $pnet = $prix * 0.95;
       echo " le prix net est de $pnet";
      }
?>
```

3.2 opérateur ?

```
<?php  //e3_2.php
    $ch=" bonjour ";
    $sexe="M";
    $ch .= ($sexe=="F") ? "Madame" : "Monsieur";
    echo "<h2>$ch</h2> <br/>";
?>
```


PHP- les instructions conditionnelles

3.3 switch

```
<?php
    $dept=75;
    switch ($dept)
    {
    case 75:
      echo "<h2>Paris</h2> <br/>";
      break;
    case 83:
      echo "<h2>Var</h2> <br/>";
      break;
    default :
      echo "departement inconnu";
    }
?>
```

PHP- les instructions répétitives

4.1 La boucle for

```
<?php   //e3_4.php
 for ($i=1;$i<7;$i++)
   {
      echo "<h$i>titre de niveau $i</h$i> ";
   }
?>
```

```
<?php   //e3_5.php
   for ($i=1,$j=9;$i<10,$j>0;$i++,$j--)
   {
      echo "<span style =\" border-style:double;border-width:3;\">  $i + $j =
      10</span> ";
   }
?>
```


PHP- les instructions répétitives

4.2 La boucle while

```
<?php    //e3_6.php
   $n=1;
   while ($n%7!=0)
   {
      $n = rand (1,100);
      echo $n,"  ";
   }
?>
```

PHP- les instructions répétitives

4.3 La boucle do while

```
<?php    //e3-7.php
   do
    {
       $n = rand (1,100);
       echo $n,"  ";
    }
    while ($n%7!=0);
?>
```


PHP- les tableaux

5.1 notion de tableau

Les tableaux (arrays) permettent de stocker plusieurs valeurs sous un même nom de variable.

Chaque élément de tableau peut être de type différent (entier, chaîne, booléen, date, array, objects, etc...)

On distingue :

- les tableaux indicés
  les éléments sont identifiés par un indice numérique
  ex. $tab[0] $tab[2] $tab[$i + 1]

- les tableaux associatifs
  les éléments sont identifiés par une étiquette (de type string) appelée CLE (Key)
  ex. $tab["nom"]

PHP- les tableaux

5.2. les tableaux à une dimension

Définir directement les valeurs du tableau

```
$tab[0] = 25;
$tab[1] = " php " ;
$tab[20] = 10.50;   // les éléments 2 à 19 n'existent pas (valeur NULL)
$tab[21] = TRUE;
$tab[ ] = " exemple ";   // si indice non précisé → indice suivant (ici 22)
$ind = 8;  $tab[$ind] = 50;   // l'indice est une variable integer

$tab2[" nom " ] = " Dupont ";  // tableau associatif
$prenom= " Jean ";
$tab2[" prenom "] = $prenom;
```

ATTENTION : Les clés des tableaux associatifs sont sensibles à la casse

PHP- les tableaux

la fonction array() pour garnir un tableau

```
$tab = array(val0,val1,val2,……,valN);

$tab = array("cleA " =>valA, " cleB " => valB, … " cleZ " =>valZ);
```

la fonction count($tableau)

La fonction count retourne le nombre d'éléments actifs (non nuls) d'un tableau $tableau donné en paramètre.

PHP- les tableaux

La fonction array_count_value (array)

Cette fonction compte le nombre de valeurs différentes dans un tableau

$resu = array_count_values($tab);

$resu sera un tableau associatif ayant pour clé les valeurs du tableau $tab et pour valeur associée le nbre d'occurrences de cette valeur dans $tab

Attention : ne s'utilise qu'avec un tableau une dimension

Exemple :

```
<?php
$tab = array (" Web ", " Internet ", " PHP ", " Javascript ", " PHP ", " ASP
    ", " PHP ", " ASP ");
$result=array_count_values($tab);
echo   " le  tableau \$tab contient ",count($tab) , " éléments <br> ";
echo   " le  tableau \$tab contient ",count($result) , " valeurs différentes
    <br> ";
print_r($result);
?>
```


PHP- les tableaux

5.3 Les tableaux multi dimensions

En PHP, pas de méthode explicite pour la création d'un tableau à n dimensions car un élément d'un tableau peut être un tableau lui-même.

```
$tabmulti = array (
            array(" Li0Col0 "  , " Li0Col1 " , " Li0Col2 " ),
            array(" Li1Col0 "  , " Li1Col1 " , " Li1Col2 " ),
            array(" Li2Col0 "  , " Li2Col1 " , " Li2Col2 " )    );
```

Ecriture : $tabmulti[2][1] = 25; // modifie contenu Li2Col1

Lecture : $el = $tabmulti[2][1] ;

NB : En test , utilisez les fonctions print_r($tabmulti) ou var_dump ($tabmulti) qui afficheront le contenu de toute la table.

PHP- les tableaux

```
for ($i=0;$i < count($tabmulti);$i++)
{
      for ($j=0;$j < count($tabmulti [ $i ] );$j++)
      {
       print $tabmulti[$i][$j] . '<br>' ;
       }
}
```

la fonction range(début,fin) pour créer des suites

Suite de nombres : $tab = range(1,10); // nbres de 1 à 10
Suite de lettres : $tab=range("A" , " M "); // lettres A à M

PHP- les tableaux

la fonction explode(sép, ch) pour extraire les éléments d'une chaîne

$chaine= " La cigale et la fourmi" ;
$tabl = explode(" " ,$chaine); // -> $tabl[0] = La , $tabl[4] = fourmi

La fonction count ($tableau)

```
$nbre = 0;
for ($i=0;$i< count($tm);$i++)
{
  if gettype($tm[$i] == " array ")
       {
          $nbre+= count($tm[$i]);
       }
  else
       {
          $nbre++;
       }
}
```

PHP- les tableaux

parcourir un tableau avec while

```
$i=0;
while (isset($tab[$i])) // isset() retourne TRUE si l'élément existe
        {   echo $tab[$i];
            $i++;    }
```

```
$clients=array (array ("Lechien", "Paris"), array ("Lechat", "Vincennes"),
          array("Lavache", "Saint-Cloud"));
$i=0;
while (isset($clients[$i])) // isset() retourne TRUE si l'élément existe
        {   $j=0;
            while (isset $clients [$i] [$j])
              {   echo $clients [$i] [$j];
                  $j++;
              }
            $i++;
        }
```


PHP- les tableaux

Parcourir un tableau avec la fonction EACH()

$element = each($tableau)

$element est un tableau de 4 éléments qui contient les infos sur l'élément courant du tableau passé en paramètre puis de pointer sur l'élément suivant :

$element[0] -> indice de l'élément courant (utilisé pour les tab indicés)

$element[1] -> valeur de l'élément courant (utilisé pour les tab indicés)

$element["key"] -> clé de l'élément courant (utilisé pour les tab associatifs)

$element[" value"] -> valeur de l'élément courant (utilisé pour les tab associatifs)

```
while($element=each($tableau))
{  /* les 2 lignes affichent  le même résultat */
echo " l'élément indice  .$element[0] . Contient . $element[1] " ;
echo " l'élément de clé  .$element['key'] . Contient . $element['value'] " ;
}
```

Attention la fonction each() déplace le pointeur sur l'élément suivant. Pour réinitialiser le pointeur, ,utilisez la fonction reset($tableau)

PHP- les tableaux

Parcourir un tableau avec la fonction list(variables) affecte à N variables la valeur des N premiers élts

```
list($x,$y,$z)=$tab;          // $x=$tab[0]   $y = $tab[1] etc …
list($x,,$y,,$z)=$tab;        // $x=$tab[0]   $y=$tab[2]  $z=$tab[4]
```

Attention, la fonction list() ne s'applique pas aux tableaux associatifs

La fonction FOREACH() (ne doit pas connaître le nombre d'élts)

Tableaux indicés

```
foreach($tab as $valeur)      {bloc de code utilisant $valeur }
foreach($tab as $indice=>$valeur)   {bloc de code utilisant $indice $valeur }
```

Tableaux associatifs

```
foreach($tab as $cle=>$valeur)   { bloc utilisant $cle et $valeur}
```


PHP- les tableaux

```
foreach($tab as $cle=>$valeur)
{ echo  " l'élément de clé  .$cle . contient . $valeur <br>" ; }
```

```
$clients = array(
"client1"=>array("nom1"=>  "Leparc" ,  "ville1"=>  "Paris"),
"client2"=>array("nom2"=>  "Laforet" ,  "ville2"=> " Bruxelles"),
"client3"=>array("nom3"=>  "Lechamp" ,  "ville3"=> " Madrid"));
foreach($clients as $cle=>$tab)
{
      echo " <B> $cle </B>" ;
      foreach($tab as $key=>$valeur)
      {
         echo " $key - $valeur" ;
      }
}
```

PHP- les tableaux

Manipuler des tableaux

array_slice() retourne un tableau qui est un sous-ensemble d'un tableau initial

```
$soustab=array_slice($tab, $indicedeb, $indicefin);
```

array_unique($tab) retourne un tableau ne contenant que la dernière occurrence des doublons.

Rem : les indices associés à chaque élément conservent leurs indices. Le tableau retourné comporte des trous dans la suite des indices.

array_merge($tab1, $tab2, …$tabn) renvoie en un seul tableau, l'union des différents tableaux passés en paramètres.

Rem : les éléments présents dans plusieurs paramètres sont présents en double dans le tableau final. Pour les éliminer, utiliser array_unique ( )

PHP- les tableaux

array_combine($tabcles,$tabvaleurs) crée un tableau associatif. $tabcles et $tabvaleurs étant des tableaux indicés contenant respectivement les clés et les valeurs associées.

array_intersect($tab1, $tab2) renvoie un tableau contenant les éléments communs aux 2 tableaux $tab1 et $tab2.

Attention : Les indices conservés correspondent à ceux du premier tableau

array_diff($tab1, $tab2) renvoie en un tableau, reprenant les éléments présents dans le premier tableau et pas dans le deuxième.

Rem : Ces 2 fonctions peuvent être appliquées à des tableaux associatifs
L'ordre des paramètres fourni est important

PHP- les tableaux

Tri des tableaux indicés

sort($tabind) trie le tableau $tabind en ordre croissant ASCII

rsort($tabind) trie le tableau $tabind en ordre décroissant

natsort($tabind) trie le tableau $tabind en ordre croissant naturel

natrsort($tabind) trie le tableau $tabind en ordre décroissant

natcasesort($tabind) trie le tableau $tabind en ordre croissant naturel sans tenir compte de la casse

NB Il ne sera pas possible de récupérer l'ordre initial, mais on peut si nécessaire dupliquer un tableau avant de le trier $tabbis = $tab

$tab2 = array_reverse(tab) inverse l'ordre des valeurs dans un nouveau tableau

shuffle($tab) mélangera de façon aléatoire les valeurs contenues dans la table.

PHP- les tableaux

Tri des tableaux associatifs

asort($tabas) trie les valeurs du tableau $tabas en ordre croissant ASCII

arsort($tabas) trie les valeurs du tableau $tabas en ordre décroissant

$boolean = ksort($rtabas) trie le tableau associatif sur la valeur croissante de ses clés et renvoie TRUE si le tri s'est bien passé (sinon FALSE)

$boolean = krsort($rtabas) idem en ordre décroissant

array_change_key_case($tabas,CASE_LOWER / CASE_UPPER) permet de transformer la casse des clés

PHP- les tableaux

Opérer une sélection des éléments

$tab2 = array_filter($tab, "userFunctionName") permet de réaliser une sélection pour ne retenir dans $tab2 que les éléments de $tab répondant aux critères de la UDF. Cette UDF reçoit 1 paramètre, représentant la valeur courante de $tab

Exemple : ne retenir les valeurs commençant par la lettre 'P' ou 'p'

```
Function init($ville)
{
 if($ville[0] == 'P' || ($ville[0] == 'p')) return $ville ;
}
```

Appel : `$tab2= array_filter($tab, "init")`

PHP- la gestion des erreurs

6. La gestion des erreurs

Un bon script ne doit pas générer d'erreurs.

Les actions de l'utilisateur peuvent générer des erreurs :
- Saisies erronées provoquant l'arrêt du script (ex : division par 0)
- Tentative d'accès à une ressource inexistante

La gestion des erreurs a pour buts :
- Éviter l'affichage des messages bruts tels que PHP les envoie au navigateur
- Signaler « proprement » les problèmes au visiteur (vu en 2ème)

PHP- la gestion des erreurs

La suppression des messages d'erreur

```
<?php  //e3_9.php
 $a = 10;
 $b = 0; echo $a/$b;
fopen(" inconnu.txt","r");
?>
```

Warning : division par zéro in c:\wamp\e3_9.php on line 14
Warning: (fopen inconnu.txt) [function open] : failed to open to stream : nosuch file or directory….

Pour éviter les messages :

1) Ajouter @ devant l'appel d'une fonction .

```
@fopen(" inconnu.txt "," r ");
```


PHP- la gestion des erreurs

Pour éviter les messages :

2) Utiliser la fonction error_reporting ( ) qui permet de n'afficher que les erreurs d'un certain niveau.

error_reporting (E_WARNING | E_PARSE) affiche les erreurs de type alerte ou syntaxe

- Error_reporting (0) bloque tous les messages d'erreur

Tableau 3-1 Liste des niveaux d'erreur courants

Constante	Valeur	Niveau d'affichage
E_ERROR	1	Erreur fatale qui provoque l'arrêt du script, par exemple, l'appel d'une fonction qui n'existe pas.
E_WARNING	2	Avertissement ne provoquant pas l'arrêt du script, par exemple, une division par 0.
E_PARSE	4	Erreur de syntaxe détectée par l'analyseur PHP et provoquant l'arrêt du script, par exemple l'oubli du point-virgule en fin de ligne.
E_NOTICE	8	Avis que le script a rencontré un problème simple qui peut ne pas être une erreur.
E_ALL	4095	Toutes les erreurs