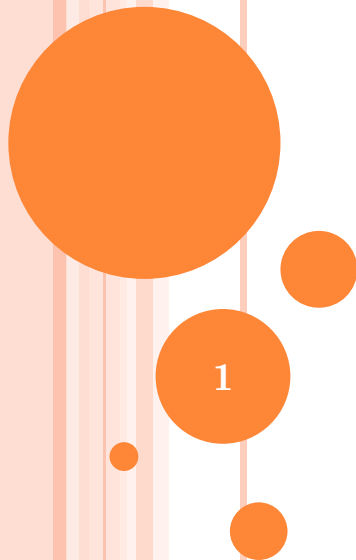




# PROGRAMMATION WEB

## Cours 1: Présentation du langage



# INTRODUCTION AU WEB

- Le terme « web » désigne le service internet permettant de naviguer à travers des pages web.
- Le protocole utilisé pour ce type de communication est le protocole HTTP (HyperText Transfer Protocol), ce qui explique le fait que vous saisissiez des URL commençant par http://.
- Le protocole HTTP en est actuellement à sa version 1.1 (spécifié par le consortium W3C).



## INTRODUCTION AU WEB

- Il existe aussi le protocole HTTPS (protocole HTTP sécurisé) qui est de plus en plus utilisé avec l'utilisation de systèmes de commerce électronique, car il s'agit d'un protocole sécurisé permettant de chiffrer les requêtes échangées entre le navigateur et le serveur.



## W3C

- Le W3C ou World Wide Web Consortium désigne un organisme international à but non lucratif. Il était créé par Tim Berners-Lee le principal inventeur du World Wide Web



- Le W3C unifie les usages et supervise également la mise en place de standards auprès des développeurs web, tels que le HTML5, HTML, XHTML, XML, RDF, SPARQL, CSS...

## W3C

- Il peut s'avérer utile de respecter les normes W3C pour différentes raisons :
  - Garantir un bon référencement et une lisibilité de votre site par les moteurs de recherche
  - Garantir un bon affichage de votre site sur les navigateurs de visiteurs



# LES PAGES STATIQUES ET LES PAGES DYNAMIQUES

- Sur le WEB, il existe deux grandes forme de pages: les pages statiques et les pages dynamiques

## Page statique:

- ces sont des pages qui s'exécute coté navigateur, elles n'ont pas besoin du serveur pour s'exécuter.
- Elle sont généralement composées par HTML, CSS et on peut les animer des langages du script comme JavaScript
- Généralement, Les pages statique portent l'extension .HTML ou .HTM

# LES PAGES STATIQUES ET LES PAGES DYNAMIQUES

## Page dynamique:

- les pages dynamiques se sont à la base des pages statiques et elles contiennent aussi des langages qui ont besoin d'un serveur pour s'exécuter.
- Les langages qu'on peut les utiliser pour créer des pages dynamiques sont: Php, Java, Python, Ruby, C#...etc



# PRÉSENTATION DU LANGAGE PHP:

- Le langage PHP est créé en 1994 par Rasmus Lerdorf pour son site web.



## Les fameux sites créés avec PHP

- PHP a permis de créer un grand nombre de sites web célèbres, comme Facebook, Wikipédia, etc



## La dernière version de PHP

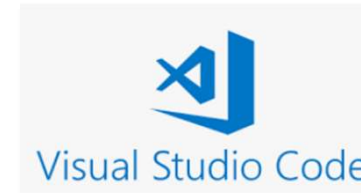
- La version actuelle est la version 8.3, sortie le 23 novembre 2023.



## OUTILS DU TRAVAIL POUR PROGRAMMER EN PHP:

- Dans nos TPs, nous allons utiliser les logiciels suivants pour programmer:

- L'éditeur Visual Studio Code:



- La plateforme de développement en PHP: WampServer 3.2



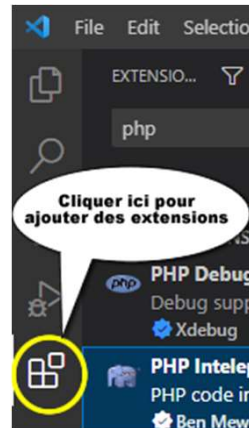
## CONSIGNES À SUIVRE POUR UTILISER CONVENABLEMENT WAMPSERVER

- Essayer d'installer WampServer sous la racine principale du système.
- Arrêter toutes les applications qui utilisent le port de communication du logiciel
- Si wampServer fonctionne correctement, il prend la couleur verte dans la barre de tâche rapide

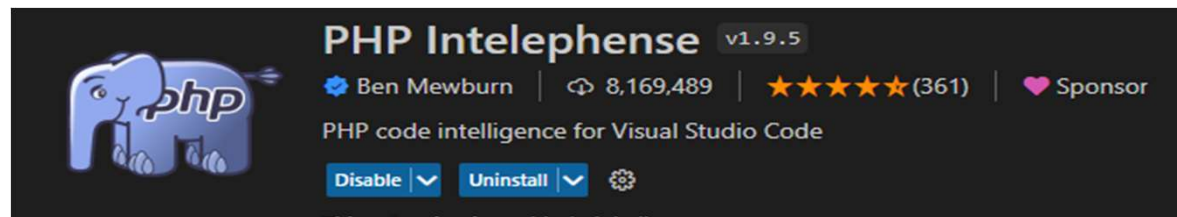


# AJOUTER DES EXTENSIONS À VSCODE

- Une fois l'éditeur VSCODE est installé passer à l'option suivantes



- Puis ajouter cette extension qui nous permet de coder sagement en PHP



## PREMIER SCRIPT EN PHP

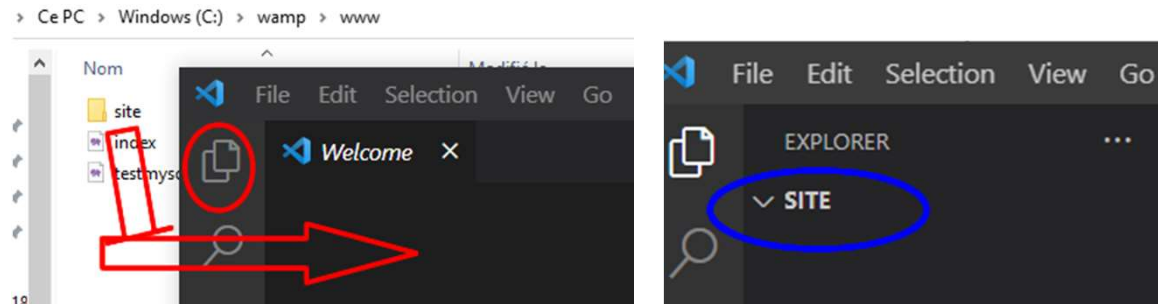
- Si votre logiciel est installé correctement, accéder au dossier www qui existe sous le dossier wamp sous la racine principale du système et créer un nouveau dossier



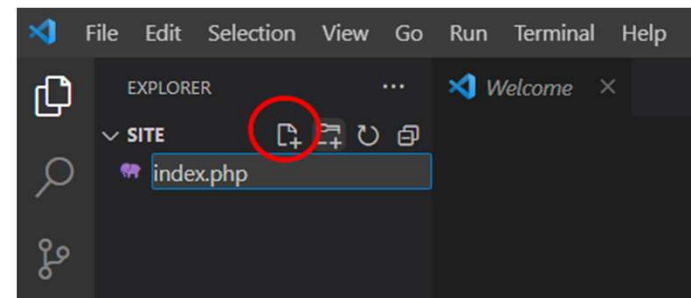
- Ce dossier va représenter votre site et il va contenir toutes ses pages web

## PREMIER SCRIPT EN PHP

- Une fois, le dossier de site est créé, vous pouvez écrire votre premier script en PHP
- Ouvrir l'éditeur VSCode, puis tirer votre dossier du site vers lui



- Une fois que votre site est ouvert sur VSCode, créer la première page index.php



## PREMIER SCRIPT EN PHP

- Voici le code de ma première page PHP:

```
<!DOCTYPE html>
<html>
<head>
    <title> Ma première page php</title>

</head>
<body>

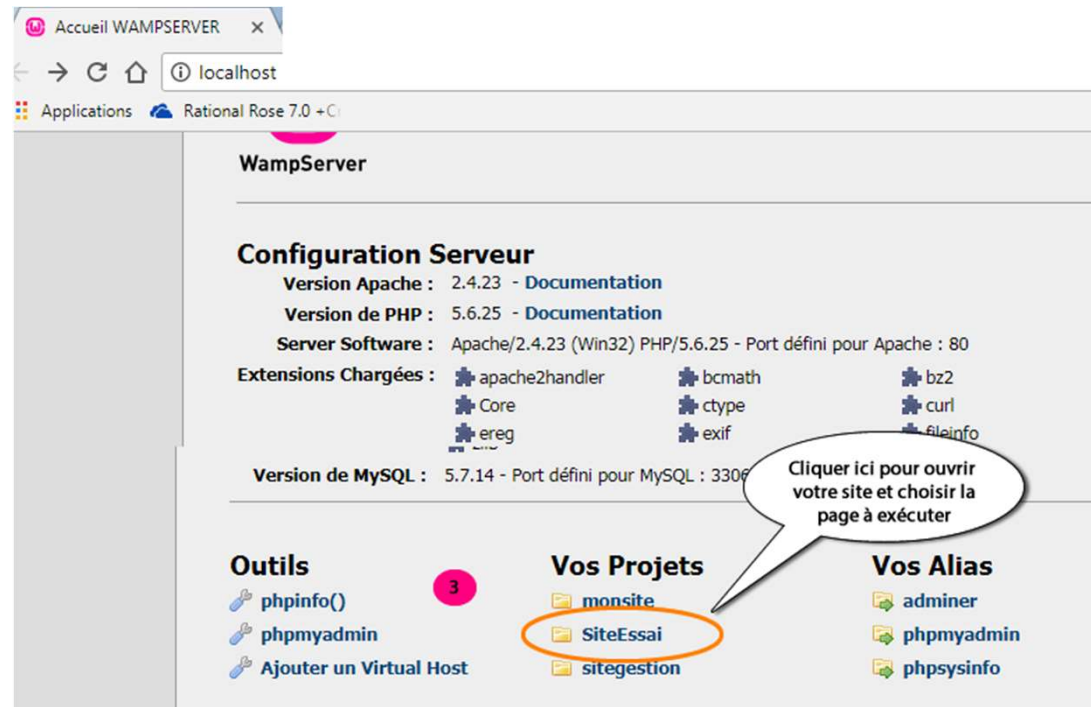
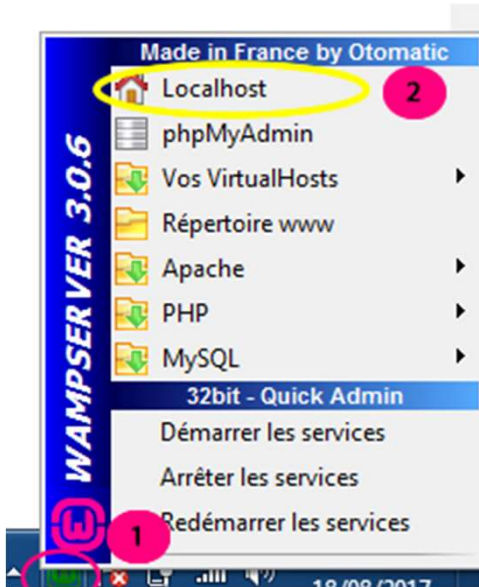
    <?php
        echo "Ma première page PHP";
    ?>
</body>
</html>
```

la balise php qui nous permet d'écrire le script

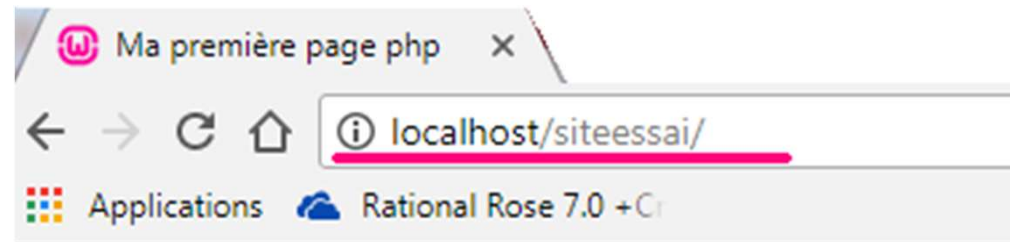
instruction d'affichage.

## L' EXÉCUTION DE LA PAGE PHP

- Une fois, votre page php est bien enregistrée dans son emplacement adéquat déjà cité dans les diapos précédentes, vous pouvez suivre les graphiques suivants pour l'exécuter:



# L' EXÉCUTION DE LA PAGE PHP



Ma première page PHP

4



## PREMIER SCRIPT EN PHP

- Pour écrire le script PHP, vous devez utiliser cette balise spéciale, voir le graphique ci-dessous:

```
<?php  
  
?>
```

- Les scripts PHP peuvent être écrits n'importe où dans le code HTML
- Les instructions PHP se terminent par un point virgule

```
<?php  
    echo 'Hello World';  
?>
```

## PREMIER SCRIPT EN PHP

- Les commentaires en php sont deux types:
  - // un commentaire sur une seule ligne
  - /\* \*/ un commentaire sur plusieurs lignes

```
<?php
//echo "Ma premiere page PHP";
/*echo "Ma premiere page PHP";
*/
?>
```

- L'instruction d'affichage de texte en PHP est echo
- Pour afficher d'un simple texte, on utilise les guillemets ou les apostrophes

```
<?php
echo 'Hello World <br>';
echo "PHP est un langage intéressant";
?>
```

# INSTRUCTIONS D’AFFICHAGE EN PHP

- Vous pouvez injecter dans le message à afficher, des balises HTML

```
<?php
    echo '<p>Hello World </p>';
?>
```

- Vous pouvez formater les balises HTML avec le CSS3

```
echo 'message avec retour à la ligne <br> voila le retour';
echo '<p style="color:blue;font-weight:bold;"> Message formaté avec css3 et html</p>';
```

- Vous pouvez appeler des scripts JavaScript en PHP, voir le code:

```
<?php
    echo "<script> alert ('ok') </script>";
?>
```

# INSTRUCTIONS D’AFFICHAGE EN PHP

- Affichage de texte qui contient des variables, de résultats des fonctions, etc.

```
<?php  
  
echo 'Nous sommes le : ' . date ("d-m-Y") . ' bonne journée';  
  
?>
```

Appel de fonction prédéfinie

## LES VARIABLES EN PHP

- Contrairement à des langages très typés comme C, C++ ou Java; PHP ne porte pas d'importance au typage des variables.
- Par exemple, pour une même variable, le programmeur est libre de lui affecter une valeur de type entier à un instant T1 puis de lui affecter une chaîne de caractères à un instant T2.
- On dit que PHP est un langage de typage « faible et dynamique ». Cela rend son utilisation plus souple par les développeurs,

## LES VARIABLES EN PHP

- Les variables en PHP doivent commencer par \$. Voir l'exemple:

```
<?php
    $var = 4;
?>
```

- Ci-dessous les règles à suivre pour bien nommer les variables en PHP

Correct	Incorrect	Explications
\$variable	variable	Une variable doit commencer par \$
\$Variable1	\$Variable 1	Les espaces sont interdits
\$variable_suite	\$variable-suite	Le tiret est interdit
\$_variable	-\$variable	Le tiret est interdit
\$variable2	\$2variable	Il ne peut y avoir de chiffre après le \$
Déclaration de variables PHP		

## LES VARIABLES EN PHP

- Autre règles de nomination:
  - Les noms de variables sont sensibles à la casse. Par exemple, \$nom et \$NOM seront considérées comme deux variables complètement distinctes.
  - Par convention, les noms de variables composés de plusieurs mots (exemple : \$estEtudiant) doivent avoir le premier mot en minuscules et les autres mots avec la première lettre en majuscule.

## LES TYPES DES VARIABLES EN PHP

- Les variables sont capables de stocker différents types d'informations. Voici les principaux types à connaître:
  - string : on entoure la chaîne de caractères de guillemets ou d'apostrophes.
  - entier, réel ou flottant : on écrit la valeur telle qu'elle. Pour les flottants, la virgule est remplacée par un point (écriture à l'américaine).
  - booléen : on écrit true ou false directement.
  - sans type : si l'on ne souhaite pas typer la variable, on lui affecte la valeur null ou NULL



## LES TYPES DES VARIABLES EN PHP

- Voir l'exemple ci-dessous:

```
<?php  
  
    $nomEtudiant = "ali";  
    $nomMatiere= 'Programmation web';  
    $nbSeance=2;  
    $present=true;  
    $autreInfo=null;  
  
?>
```

# AFFICHAGE ET CONCATÉNATION DES VARIABLES

- Concaténer avec des guillemets doubles:
  - Avec des guillemets doubles, c'est le plus simple. Vous pouvez écrire le nom de la variable au milieu du texte et il sera remplacé par sa valeur.
  - Voir l'exemple:

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

# AFFICHAGE ET CONCATÉNATION DES VARIABLES

- Concaténer avec des guillemets simples:
  - La concaténation avec des guillemets simples se fait avec l'opérateur .
  - Voir l'exemple:

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

## QUELQUES OPÉRATEURS EN PHP:

### ○ Les opérateurs arithmétiques:

Opérateur	Opération	Exemple	Résultat
-	Négation	-\$a	Opposé de \$a
+	Addition	\$a + \$b	Somme de \$a et \$b
*	Multiplication	\$a * \$b	Produit de \$a et \$b
-	Soustraction	\$a - \$b	Différence de \$a et \$b
/	Division	\$a / \$b	Quotient de \$a et \$b
%	Modulo	\$a % \$b	Reste de \$a / \$b

### ○ Les opérateurs d'incrémentation/ décrémentation:

Opérateur	Opération	Exemple	Résultat
++	Pré-Incrémentation	++\$a	Incrémente \$a, puis retourne \$a
++	Post-Incrémentation	\$a++	Retourne \$a, puis incrémente \$a
--	Pré-Décrémentation	--\$a	Décrémente \$a, puis retourne \$a
--	Post-Décrémentation	\$a--	Retourne \$a, puis décrémente \$a

## QUELQUES OPÉRATEURS EN PHP:

### ○ L'opérateur d'assignation:

Opérateur	Opération	Exemple	Résultat
=	Assignation	\$a = 3	Affecte la valeur 3 à \$a

### ○ Les opérateurs de chaîne de caractères

Opérateur	Opération	Exemple	Résultat
.	Concaténation	\$a . \$b	Concatène les valeurs de \$a et \$b
.=	Assignation concaténant	\$a .= \$b	Ajoute la valeur de \$b à la suite de celle de \$a

### ○ Les opérateurs logiques

Opérateur	Opération	Exemple	Résultat
&&	ET	\$a && \$b	TRUE si \$a ET \$b sont vrais
AND	ET	\$a AND \$b	Alias de &&
	OU	\$a    \$b	TRUE si \$a OU \$b est vrai
OR	OU	\$a OR \$b	Alias de
XOR	OU exclusif	\$a XOR \$b	TRUE si \$a OU \$b est vrai mais pas les deux
!	NON	!\$a	TRUE si \$a est faux

## QUELQUES OPÉRATEURS EN PHP:

- Les opérateurs de comparaison:

Opérateur	Opération	Exemple	Résultat
==	Egalité en valeur	\$a == \$b	Vérifie que les valeurs de \$a et \$b sont identiques
===	Egalité en valeur et type	\$a === \$b	Vérifie que les valeurs et types de \$a et \$b sont identiques
!=	Différence en valeur	\$a != \$b	Vérifie que les valeurs de \$a et \$b sont différentes
!==	Différence en valeur et type	\$a !== \$b	Vérifie que les valeurs et types de \$a et \$b sont différents
<>	Différence en valeur	\$a <> \$b	Alias de !=
<	Infériorité stricte	\$a < \$b	Vérifie que \$a est strictement inférieur \$b
<=	Infériorité ou égalité	\$a <= \$b	Vérifie que \$a est strictement inférieur ou égal à \$b
>	Supériorité stricte	\$a > \$b	Vérifie que \$a est strictement supérieur \$b
>=	Supériorité ou égalité	\$a >= \$b	Vérifie que \$a est strictement supérieur ou égal à \$b

## LES CONSTANTES EN PHP

- Tout comme en C, PHP peut utiliser des variables dont on précise la valeur une fois pour toute et qui ne pourra être modifiée jusqu'à la fin de l'exécution du code.
- Le nom d'une constante suit les mêmes règles que celles pour les variables, mis à part que le nom d'une constante n'est pas précédé par \$.
- Ci-dessous, vous voyez comment déclarer une constante en PHP:

```
<?php  
define("CONSTANTE", "Texte ici");  
echo CONSTANTE; // affiche "Texte ici"  
?>
```

## QUELQUES FONCTIONS PRÉDÉFINIES SUR LES VARIABLES EN PHP

- `isset()`: une fonction prédéfinie qui teste de l'existence de la variable. Voir l'exemple:

```
$prenom = 'Hugo';  
  
echo isset($prenom); // Retourne TRUE -> affiche 1  
  
echo '<br/>';  
  
echo isset($nom); // Retourne FALSE -> n'affiche rien
```

- `unset()`: une fonction qui permet de détruire des variables en cours d'exécution du programme

```
$prenom = 'Hugo';  
  
unset($prenom); //détruit la variable $prenom  
  
echo isset($prenom); // Retourne FALSE -> n'affiche rien
```



## QUELQUES FONCTIONS PRÉDÉFINIES SUR LES VARIABLES EN PHP

- Void `var_dump($var)` — Affiche les informations (type, contenu, etc.) de la variable `$var`. voir l'exemple:

```
$prenom = 'Hugo';  
var_dump($prenom);
```



```
C:\wamp\www\SiteEssai\fonctionvar.php:31:string 'Hugo' (length=4)
```

## LES CONDITIONS EN PHP

- Comme le langage C, Le langage PHP introduit 4 constructions conditionnelles : if, elseif, else et switch.
- L'instruction conditionnelle if()

### Principe de fonctionnement de l'instruction if()

```
if(expression)
{
    bloc d'instructions;
}
```

- Exemple:

```
<?php
    $var =8;
    if($var < 10 ){
        echo '<p style="color:red;font-weight:bold;">'.$var. ' :Note faible</p>';
    }

?>
```

## LES CONDITIONS EN PHP

- La clause else:

```
$var =12;  
  
if($var < 10 ){  
    echo '<p style="color:red;font-weight:bold;">'.$var. ' :Note faible</p>';  
}  
  
else {  
    echo '<p style="color:yellow;font-weight:bold;">'.$var. ' :Bonne note</p>';  
}
```

- On peut omettre les accolades d'une structure, si on a une seule instruction

```
$var =12;  
  
if($var < 10 ) echo '<p style="color:red;font-weight:bold;">'.$var. ' :Note faible</p>';  
else echo '<p style="color:yellow;font-weight:bold;">'.$var. ' :Bonne note</p>';
```

# LES CONDITIONS EN PHP

- L'instruction elseif()

```
<?php
// Déclaration d'une variable
$vitesse = 82;

// On teste la valeur de la variable
if($vitesse > 50 && $vitesse < 70)
{
    echo 'Les policiers m\'arrêtent !';
    echo 'Je perds 1 point pour excès de vitesse !';
    echo 'Le policier me dit : "Roulez doucement maintenant !"';
}
elseif($vitesse > 70 && $vitesse < 80)
{
    echo 'Les policiers m\'arrêtent !';
    echo 'Je perds 2 points pour excès de vitesse !';
    echo 'Le policier me verbalise par une amende';
}
elseif($vitesse > 80 && $vitesse < 90)
{
    echo 'Les policiers m\'arrêtent !';
    echo 'Je perds 3 points pour excès de vitesse !';
    echo 'Le policier me verbalise par une grosse amende !';
}
elseif($vitesse > 90)
{
    echo 'Les policiers m\'arrêtent !';
    echo 'Je perds 4 points pour excès de vitesse !';
    echo 'Le policier me retire mon permis !';
    echo 'Je finis à pied !';
}
else
{
    echo 'Les policiers me laissent suivre ma route !';
}
?>
```

# LES CONDITIONS EN PHP

- L'instruction elseif()

```
<?php
// Déclaration d'une variable
$vitesse = 82;

// On teste la valeur de la variable
if($vitesse > 50 && $vitesse < 70)
{
    echo 'Les policiers m\'arrêtent !';
    echo 'Je perds 1 point pour excès de vitesse !';
    echo 'Le policier me dit : "Roulez doucement maintenant !"';
}
elseif($vitesse > 70 && $vitesse < 80)
{
    echo 'Les policiers m\'arrêtent !';
    echo 'Je perds 2 points pour excès de vitesse !';
    echo 'Le policier me verbalise par une amende';
}
elseif($vitesse > 80 && $vitesse < 90)
{
    echo 'Les policiers m\'arrêtent !';
    echo 'Je perds 3 points pour excès de vitesse !';
    echo 'Le policier me verbalise par une grosse amende !';
}
elseif($vitesse > 90)
{
    echo 'Les policiers m\'arrêtent !';
    echo 'Je perds 4 points pour excès de vitesse !';
    echo 'Le policier me retire mon permis !';
    echo 'Je finis à pied !';
}
else
{
    echo 'Les policiers me laissent suivre ma route !';
}
?>
```

## LES CONDITIONS EN PHP

- L'instruction `switch()`
- Il existe une autre alternative à la structure `if()` / `elseif()` / `else` ou bien aux imbrications de blocs `if()`. Elle se nomme `switch()`. Cette instruction conditionnelle permet de tester toutes les valeurs possibles que peut prendre une variable. Voici sa syntaxe:

```
switch(expression)
{
    case valeur1 :
        instructions;
        break;

    case valeur2 :
        instructions;
        break;

    default :
        instructions;
        break;
}
```

# LES CONDITIONS EN PHP

- L'instruction switch()

```
Exemple d'utilisation de l'instruction conditionnelle switch()
<?php
// Le joueur a choisi son personnage ('elfe')
$personnage = 'elfe';

// Test de la valeur du personnage
switch($personnage)
{
    case 'sorcier' :
        echo 'Vous avez choisi le sorcier !';
        echo 'Vous disposez de pouvoirs magiques.';
        break;

    case 'guerrier' :
        echo 'Vous avez choisi le guerrier !';
        echo 'Vous disposez d\'armes blanches.';
        break;

    case 'roi' :
        echo 'Vous avez choisi le roi !';
        echo 'Vous vivez dans un château entouré de gardes.';
        break;

    case 'elfe' :
        echo 'Vous avez choisi l\'elfe !';
        echo 'Vous pouvez voler et vous déplacer rapidement.';
        break;

    case 'ogre' :
        echo 'Vous avez choisi l\'ogre !';
        echo 'Vous avez beaucoup de force dans les bras.';
        break;

    default :
        echo 'Veuillez choisir votre personnage svp !';
        break;
}
?>
```

## LES CONDITIONS EN PHP

- Remarque:
- Vous pouvez réduire les tests booléens, en utilisant les écritures suivantes:

```
if($var == true ) /*traitement*/;
```

équivalent

```
if($var) /*traitement*/;
```

```
if($var==false) /*traitement*/;
```

équivalent

```
if(! $var) /*traitement*/;
```



## LES BOUCLES EN PHP

- **for()**: cette boucle a la même structure de celle qui existe en langage C :
- Exemples:

```
for ($i=0; $i<10; $i++) {  
    echo 'Ceci est une boucle for en PHP'; // afficher 10 fois ce message  
}
```

```
for ($i=5; $i<9; $i++) {  
    echo $i; // afficher les valeurs de la variable i de 5 à 8  
}
```

## LES BOUCLES EN PHP

- **Do ... While() et while()...**: ces deux boucles ont la même structure que celles qui existent en langage C :

- Exemple:

```
$i = 5;  
do {  
    echo $i;  
    $i++;  
} while ($i<9)
```

- **Foreach()**: cette boucle sera étudiée profondément dans le prochain cours

## LES BOUCLES EN PHP

- **Les instructions d'arrêt et de continuité:** PHP introduit également deux instructions particulières des boucles. Il s'agit de continue et break.
- La première permet de forcer le passage à l'itération suivante en sautant tout ou une partie du bloc d'instructions à exécuter.
- **Voir l'exemple:**

```
Exemple d'utilisation de continue
<?php
for($i=0; $i<=10; $i++)
{
    // On n'affiche pas les 5 premiers nombres
    if($i<=5)
    {
        continue;
    }
    // Affichage des nombres de 6 à 10
    echo $i , '<br/>';
}
?>
```

## LES BOUCLES EN PHP

- La seconde, quant à elle, permet de forcer l'arrêt d'une structure conditionnelle telle que for(), while(), foreach() ou switch().
- Voir l'exemple:

### Exemple d'utilisation de break

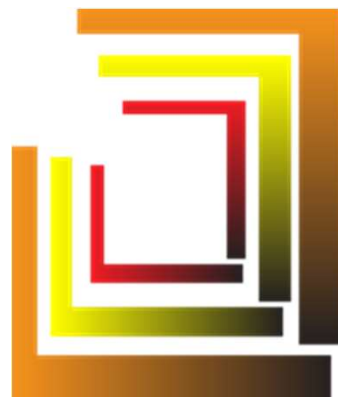
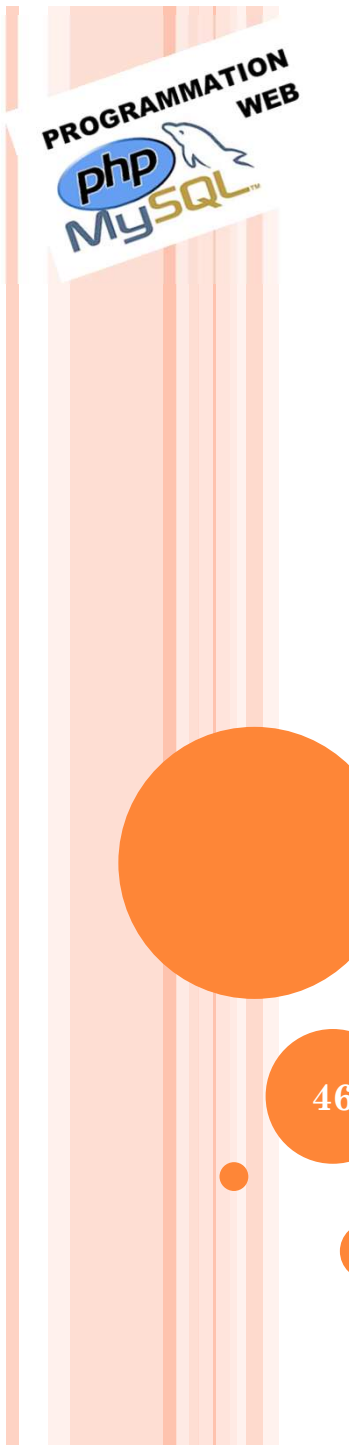
```
<?php
for($i=0; $i<=10; $i++)
{
    // On affiche la valeur du compteur
    echo 'Compteur = ';
    // On sort de la boucle si on atteint le chiffre 5
    if($i>5)
    {
        break;
    }
    // Affichage des nombres de 0 à 5
    echo $i , '<br/>';
}
?>
```

# LES BOUCLES EN PHP

- Les instructions d'arrêt et de continuité:

Exemple d'utilisation de continue

```
<?php
for($i=0; $i<=10; $i++)
{
    // On n'affiche pas les 5 premiers nombres
    if($i<=5)
    {
        continue;
    }
    // Affichage des nombres de 6 à 10
    echo $i , '<br/>';
}
?>
```



***FIN***  
***Mansour Sihem***