

INTRODUCTION A PHP

Université Virtuelle de Côte d'Ivoire



Table des matières

| | |
|---|----|
| Objectifs | 4 |
| Introduction | 5 |
| I - Fonctionnement et environnement de PHP | 6 |
| 1. Fonctionnement du langage PHP | 6 |
| 2. Environnement de travail | 8 |
| 3. Premier projet en PHP | 10 |
| II - Exercice : Activité d'auto-évaluation 1 | 11 |
| 1. Exercice | 12 |
| 2. Exercice | 13 |
| 3. Exercice | 14 |
| 4. Exercice | 15 |
| 5. Exercice | 16 |
| 6. Exercice | 17 |
| 7. Exercice | 18 |
| 8. Exercice | 19 |
| III - Les éléments de base du langage | 20 |
| 1. Les variables | 20 |
| 2. Conversion de variable | 21 |
| 3. Les constantes | 22 |
| 4. Les commentaires | 22 |
| 5. Les opérateurs | 23 |
| 6. Les fonctions mathématiques | 26 |
| 7. Les inclusions de fichiers | 28 |

| | |
|---|-----------|
| 8. Travaux dirigés | 29 |
| IV - Exercice | 31 |
| 1. Exercice | 32 |
| 2. Exercice | 33 |
| 3. Exercice | 34 |
| 4. Exercice | 35 |
| 5. Exercice | 36 |
| 6. Exercice | 37 |
| 7. Exercice | 38 |
| 8. Exercice | 39 |
| V - Les structures de contrôle | 40 |
| 1. Les structures conditionnelles | 40 |
| 2. Les structures itératives | 43 |
| 3. Travaux dirigés | 45 |
| VI - Exercice | 48 |
| 1. Exercice | 49 |
| 2. Exercice | 50 |
| 3. Exercice | 51 |

Objectifs



Cette leçon vise à définir le fonctionnement, l'environnement d'exécution, et les éléments de base du langage PHP.

Objectifs d'apprentissage

À la fin de cette leçon, vous serez capable :

- De définir le fonctionnement et l'environnement de PHP
- D'identifier les éléments de base du langage
- D'utiliser les structures de contrôle

Introduction



PHP (officiellement, ce sigle est un acronyme récursif pour PHP Hypertext Preprocessor, anciennement appelé Personal Home Page) est un langage de scripts Open Source, spécialement conçu pour le développement d'applications web dynamique*. Il peut être intégré facilement au HTML et s'exécute principalement via un serveur HTTP. Il est à la fois procédural et orienté objets

, on dit que c'est un langage impératif orienté objets. PHP a des concurrents tels que ASP.Net, Python et Java EE, en 2016 PHP occupe 82% du marché des langages de programmation côté serveur des sites web. Il a beaucoup évolué et sa version actuelle est la version 7.

Pour plus d'information consultez la page : <https://fr.wikipedia.org/wiki/PHP>

* : On dit qu'une page web est dynamique lorsque son contenu peut varier d'un moment à l'autre.

Fonctionnement et environnement de PHP



1. Fonctionnement du langage PHP

Notion de client et de serveur

Pour comprendre le fonctionnement de PHP, il faut comprendre la notion de client et de serveur, en effet internet un réseau basé sur une architecture client-serveur. En informatique :

- le client peut être représenté par une application ou une machine. On dit qu'une application est client lorsqu'elle envoie une demande ou requête à une autre application (Ex : navigateur Firefox, Google chrome, Safari etc.). L'ordinateur sur lequel cette application est installée est appelé ordinateur client.
- Un serveur représente une entité qui reçoit des demandes et les traite. Le serveur peut être représenté par une application ou une machine. On dit qu'une application est serveur lorsqu'elle reçoit des requêtes, les traite, puis fournit un résultat aux clients. L'ordinateur sur lequel cette application est installée est appelé ordinateur serveur. Il existe plusieurs type de serveur

Un protocole est un ensemble de règles permettant à deux applications ou machines de communiquer. C'est un moyen par lequel le serveur et le client communiquent. Il existe plusieurs types de protocoles en fonction du service rendu. Nous avons le protocole HTTP (HyperText Transfert Protocol) pour le web, le protocole FTP (File Transfer Protocol) pour les transferts de fichiers, le protocole SMTP (Simple Mail Transfer Protocol) et POP3 pour la messagerie etc. A un protocole donné correspond un serveur et un client donné.

Dans le cadre de ce cours, il s'agit du protocole http, utilisant des clients http (navigateurs), et les serveurs http ou web tels que apache (apache tomcat et apache HTTP server), NodeJS, Nginx, Internet Information Services (IIS) de Microsoft, etc.

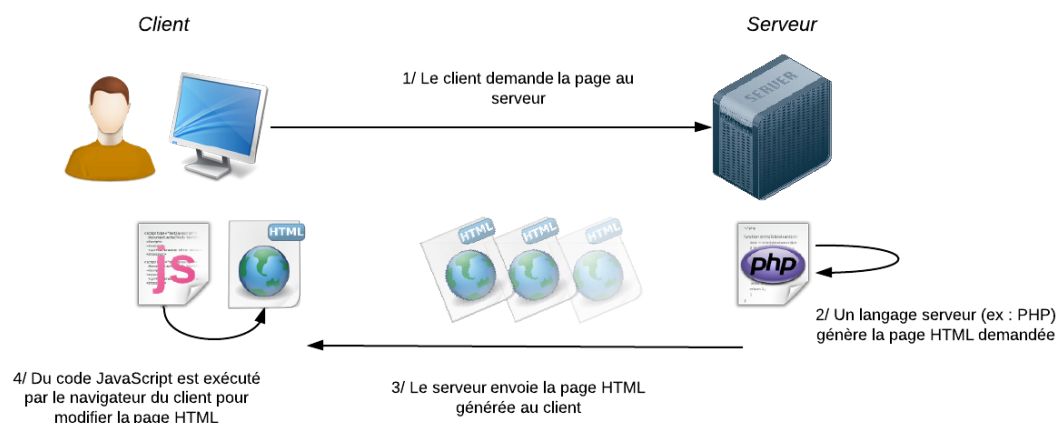
Nous utiliserons le serveur http apache qui prend en compte par défaut le langage PHP.

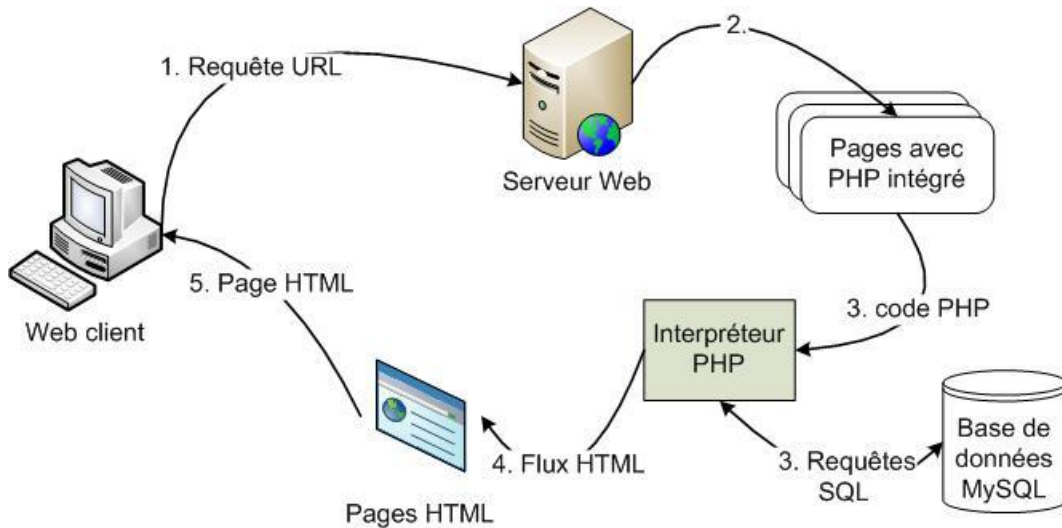
Définition : Serveur Web

Un serveur web est une application qui reçoit des demandes http provenant des clients web (navigateur) et les traite, puis leur fournit des pages web

Exécution d'une page PHP

- Lorsqu'une page web est demandée via un navigateur, le navigateur cherche où se trouve le serveur hébergeant (disposant) la page demandée. Cela est possible grâce à des serveurs DNS (Domaine Name Server). Dès que le serveur est trouvé, le navigateur web lui envoie une requête contenant l'adresse de la page demandée.
- Lorsque le serveur reçoit la requête, il recherche immédiatement le fichier demandé, récupère son contenu, puis le renvoie au navigateur. Selon le type de fichier demandé par le client (fichier HTML, fichier PHP, etc.), le serveur ne va cependant pas procéder de la même manière.
- Dans le cas d'un fichier HTML, le serveur web se contente de récupérer le fichier et de le renvoyer tel quel au navigateur, qui se chargera ensuite de l'afficher dans le navigateur. En revanche, si on demande au serveur l'accès à une page PHP, une étape de plus va être nécessaire, car le navigateur web ne peut pas lire du code PHP. Celui-ci devra être interprété d'abord, c.à.d. traduit en code html.
- Lorsque le serveur récupère le fichier PHP, il utilise un interpréteur chargé d'exécuter le code PHP et de ne renvoyer que le résultat sous forme de code HTML (et éventuellement CSS et JavaScript). Une fois le résultat obtenu, le serveur web le renvoie au navigateur qui peut l'afficher sans problème. Pendant l'interprétation, le moteur web PHP peut interagir avec une base de données pour échanger des informations.





2. Environnement de travail

Les logiciels nécessaires :

Pour développer un projet en PHP, il faut les logiciels suivants :

- Un éditeur de texte, on a des gratuits et des payants :
 - Editeurs gratuits : Sublime Text, notepad++, Visual studio code, netbeans, etc.
 - Editeurs payants : phpdesigner, dreamwaver, etc.

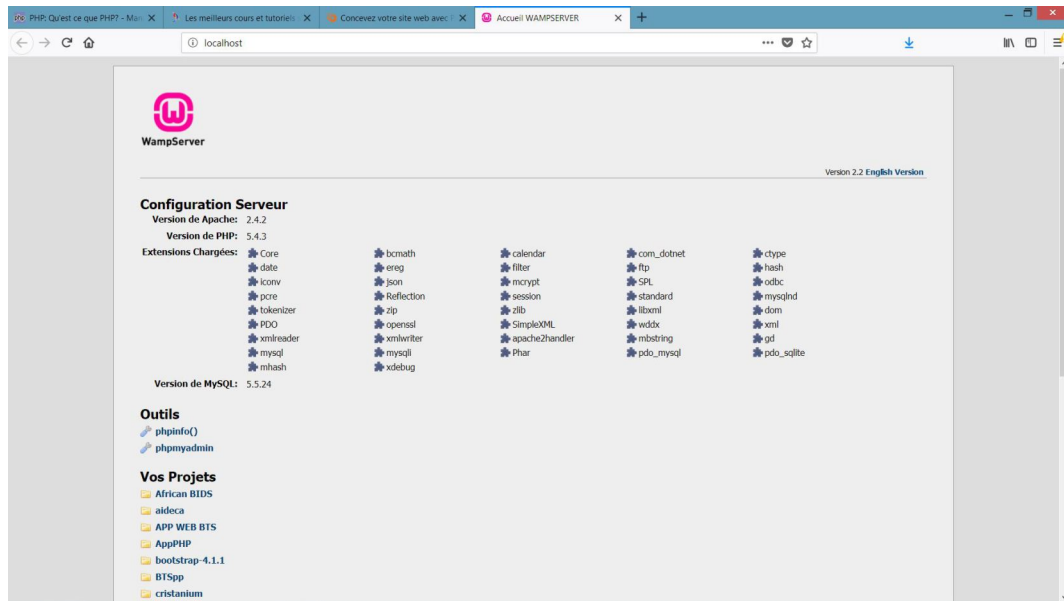
Je vous conseille d'utiliser Visual studio code pour travailler. Vous pouvez avoir en parallèle notepad++ et sublime text.

- D'un serveur web avec éventuellement une base de données. Dans le cadre de ce cours, nous utiliserons le serveur web apache et MySQL comme base de données. Pour les avoir, on peut télécharger et installer le logiciel WAMP ou XAMPP.
 - Pour WAMP, je conseille une version 2.5, la version 3 est beaucoup plus délicate et ne nous intéresse véritablement pas pour ce cours.
 - Pour XAMPP, je conseille la version 3.

Installation et configuration

- *Installation et configuration de WAMP*
 - Télécharger le logiciel à l'adresse suivante : <https://www.commentcamarche.net/download/telecharger-34055064-wampserver>
Ou à : <https://sourceforge.net/projects/wampserver/files/WampServer%20/Wampserver%202.5/>
 - Installer le, par défaut wamp va s'installer sur la partition C dans un dossier wamp (C:\wamp)

- Une fois l'installation finie, vous allez voir son icône apparaître à droite dans la barre des tâches. Vous cliquez sur l'icône et un menu va apparaître, dans le menu vous cliquez sur localhost, vous allez voir une page comme suit dans le navigateur par défaut:



Vous pouvez aussi accéder à wamp server en saisissant localhost dans la barre d'adresse de n'importe quel navigateur.

- Chaque fois que vous allumez votre ordinateur, si vous voulez utiliser wamp, vous devez lancer wamp à partir de son icône sur le bureau ou bien dans les applications. Ensuite, vous procédez comme précédemment.
- *Quelques préoccupations :*

Lorsque wamp est lancé, si le serveur est en marche, l'icône de wamp dans la barre des tâches. Si l'icône est rouge ou orange, c'est que le serveur rencontre quelques problèmes pour démarrer les services. Cela peut être occasionné par plusieurs cas :

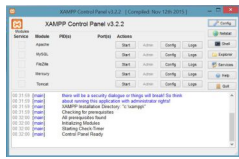
- Soit le port 80 utilisé par wamp server est occupé par une application tierce telle que skype ou IIS. Dans le cas il faut désactiver cette application. Pour savoir quelle application utilise le port 80 vous cliquez sur l'icône de wamp dans la barre des tâches, puis sur apache, puis service, et enfin vous cliquez sur tester le port 80. Une autre astuce est de changer le port 80 utilisé par wamp en un autre port. Mais nous ne verrons cette manière ici.
- Si l'icône de wamp est vert mais vous ne pouvez pas accéder à la page d'accueil de wamp en passant par localhost, alors, il faut saisir 127.0.0.1 dans la barre d'adresse.

Nb : Un port est un numéro permettant d'identifier une application ou service sur un ordinateur.

- *Installation et configuration de XAMPP*

- Télécharger le logiciel à l'adresse suivante : <https://www.apachefriends.org/fr/download.html> ou bien <https://xampp.fr.uptodown.com/windows>

- Installer le, par défaut dans le répertoire C:\xampp. Il faut désactiver tout antivirus pour éviter des problèmes d'installation.
- Une fois l'installation finie et xampp lancé, vous aurez cette fenêtre :



- Pour démarrer le serveur apache vous cliquez sur le bouton start devant apache, pareil pour le serveur mysql.
- Il faut toujours prendre le soin de démarrer le serveur avant d'exécuter du code PHP
- *Lien utile* : <https://www.1and1.fr/digitalguide/serveur/outils/tutoriel-xampp-creerun-serveur-de-test-local/>

3. Premier projet en PHP

- Créez un dossier test dans C:\wamp\www\ pour ceux qui utilise wamp ou bien dans C:\xampp\htdocs\ pour ceux qui utilisent xampp. Utilisez l'un ou l'autre et non les deux à la fois.
- Avec un éditeur de texte, créer une page appelée hello et enregistrez là avec l'extension .php dans le dossier test.
- Mettez dans votre page le code suivant et enregistrez les modifications:

```
<!DOCTYPE HTML> <html>

<head>

<title>Exemple PHP</title>

</head>

<body>

<?php echo "Bonjour les Etudiants !"; ?>

</body>

</html>
```

- Allez dans votre navigateur et saisissez l'adresse : localhost/test/hello.php
- Vous verrez un message « Bonjour les Etudiants » dans le navigateur.
- Chaque fois que vous devez afficher du code PHP, vous devriez procéder de cette manière.
- *Remarques* :
 - Le code php s'écrit à l'intérieur d'un code html grâce à la balise php qui s'écrit de la manière suivante : <?php ?>
 - *echo* est une fonction php permettant d'afficher du texte dans une page web.
 - Remplacez la balise php par la suivante : <?php echo date("d-m-Y H:m:s") ; ?>
 - Vous verrez la date et l'heure de votre machine apparaissent. L'heure changera chaque fois que vous rafraîchissez la page, ça c'est du code dynamique.

Exercice : Activité d'auto-évaluation 1



0. Exercice

Une application client est :

- ☐ une application qui est conçu avec des langages client
- ☐ une application conçu par un client
- ☐ une application qui reçoit des requêtes client
- ☐ une application qui envoie des requêtes à un serveur
- ☐ une application pour des utilisateurs

0. Exercice

Un langage qui s'exécute côté serveur est appelé langage 

0. Exercice

Une application qui reçoit des demandes http provenant de clients web (navigateur) et qui les traite, puis leur fournit des pages web est appelée :

(consigne : tous les caractères doivent être saisis en minuscule. La réponse est constituée de deux mots séparés par un espace)

0. Exercice

Parmi ces logiciels lesquels sont des éditeurs de texte ? (5 réponses)

- ☐ Sublime text
- ☐ Big note
- ☐ PHP executor
- ☐ Netbeans
- ☐ Visual studio code
- ☐ Xampp app
- ☐ Dreamwaver
- ☐ Wordpress
- ☐ Joomla
- ☐ Notepad++

0. Exercice

Pour exécuter des pages web en PHP, on installe un logiciel qui comprend MySQL, Apache et PHP, ce logiciel s'appelle :

(consigne : tous les caractères doivent être saisis en majuscule.
deux réponses possibles)

0. Exercice

On dit qu'une page web est dynamique lorsque :

- ☐ son contenu dans le navigateur peut changer sans qu'on ne modifie le code source de la page
- ☐ son contenu dans le navigateur se rafraîchit à chaque instant
- ☐ Lorsqu'il est possible de modifier le code source de la page lorsqu'elle est affichée dans le navigateur
- ☐ lorsque la page est créée avec l'extension .php

0. Exercice

La démarche pour créer et afficher une page php sous XAMPP est : (4 propositions)

- ☐ après avoir démarré lancer xampp, on démarre les serveurs (apache, mysql)
- ☐ on crée un répertoire (dossier) dans le répertoire htdocs de xampp
- ☐ on crée un répertoire (dossier) dans le répertoire www de xampp
- ☐ on attribue un login et password à l'admin root de xampp
- ☐

on crée à l'aide d'un éditeur de texte une page d'extension .php dans laquelle on met notre code et que nous enregistrons dans notre répertoire créé dans www

- ☐ on crée à l'aide d'un éditeur de texte une page d'extension .php dans laquelle on met notre code et que nous enregistrons dans notre répertoire créé dans htdocs
- ☐ on définit un identifiant pour notre projet
- ☐ à l'aide d'un navigateur, on saisie dans la barre d'adresse : localhost/répertoire/page.php
- ☐ à l'aide d'un navigateur, on saisie dans la barre d'adresse : xampp/dossier/page

0. Exercice

On aimerait écrire un script php qui affiche un texte contenu dans une variable en format titre, centré, en italique, et de couleur bleu. Veuillez compléter le code.

```
<?php  
$message= "Soyez les bienvenues" ;  
[ ] "<h1 [ ] ='color :blue ; [ ] :italic ;'> <center> [ ] <  
/center></h1>" ;  
?>
```



Les éléments de base du langage

1. Les variables

Une variable représente une mémoire permettant de stocker une information pendant l'exécution d'un programme. Elle est représentée par un nom (identifiant qui est définie conformément aux règles d'identification* prônées par tous les langages de programmation) et un type (la nature de la variable).

En php :

- toute variable est précédée du caractère \$ (prononcez dollar) ;
- le typage d'une variable est dynamique, c'est-à-dire pas besoin de préciser le type d'une variable, php va le deviner en fonction de l'information qu'on y stocke.
- une variable peut être déclarée dans n'importe quelle partie du programme (pas besoin d'une partie déclaration pour déclarer une variable à l'instar des langages comme Pascal, C et Java)
- très souvent, la déclaration de la variable se fait automatiquement avec son initialisation

Syntaxe de déclaration d'une variable

\$nomVariable = valeur ;

Ou bien

```
$nomVariable1= valeur1 , $nomVariable2=valeur2 ;
```

Exemple :

```
$nom="Koffi" ; $age=25 ; $x=17, $y=20.5 ;
```

NB :

- En php, toute instruction se termine par un point-virgule (;)
- Php est un langage qui respecte la casse (ou est de case sensitive), c.-à-d. elle fait une distinction entre les minuscules et les majuscules au niveau des identificateurs (variable, fonction, procédure, les mots réservés, etc.)
- * les règles d'identification disent qu'un identificateur doit être unique, ne doit pas commencer par un chiffre, ne doit pas comprendre de caractère accentué, ni d'espace, etc.)
- Pour détruire une variable on utilise la fonction *unset()*. *Ex : unset (\$x) ;*
- L'affectation se fait avec le symbole = ; *Ex : \$x = 20 ;*

Les types de variable

Comme tout langage de programmation, en php les variables sont typées, mais de façon dynamique (c.-à-d., l'utilisateur n'a pas besoin de préciser le type de la variable, php le saura en fonction de l'information qui sera stockée). Une variable peut au départ contenir un entier, puis contenir un réel et une chaîne de caractère à la suite.

Quelques types de variables que nous avons sont :

- string (chaîne de caractère)
- integer (entier)
- double, float (réel)
- boolean (booléen)
- array (tableau)

Remarques

- Pour connaître le type d'une variable on utilise la fonction `gettype()`. Ex : `echo gettype ($x) ;`

2. Conversion de variable

Pour changer le type d'une variable, il existe plusieurs astuces.

- Forcer la conversion

Php peut forcer la conversion d'une variable avec la fonction `settype()` qui prend en charge deux paramètres. Le premier est la variable à convertir et le second est le type de conversion.

La fonction inverse `gettype()` fournit l'information de type d'une variable

```
<?php
$val = "22";

settype($val,'integer'); // on force la conversion

echo gettype($val); // affichera integer

?>
```

Il existe aussi des fonctions dédiées à la conversion, comme `intval()` (pour convertir une valeur en entier), `floatval()`, `strval()`, etc

```
<?php

echo gettype( strval(13) ); // affichera string

?>
```

Modification de type

En PHP une variable peut récupérer la valeur d'une autre variable tout en modifiant son type. Il suffit d'affecter la nouvelle variable par l'ancienne précédée entre parenthèses du type

```
<?php
$ancienne = 1.3;
$nouvelle = (int)$ancienne;
echo $nouvelle; // affichera 1
?>
```

Les différentes valeurs à fournir dans les parenthèses sont :

- (int), (integer) : modification en integer
- (bool), (boolean) : modification en boolean
- (float), (double), (real) : modification en float
- (string) : modification en string
- (array) : modification en array
- (object) : modification en object

3. Les constantes

Une constante se définit comme étant une variable non modifiable. Son contenu est figé.

Pour déclarer une constante en php, on procède de la manière suivante :

```
define("nomVariable", valeur);

<?php

define("TVA", 21.5);

echo TVA;

?>
```

4. Les commentaires

Les commentaires permettent d'apporter des explications au code ou de rendre une partie du code non exécutable. En php, pour faire des commentaires, nous avons deux possibilités :

- Sur une seule ligne la syntaxe est : *//texte en commentaire*
- Sur plusieurs lignes :

```
/* ligne 1 en commentaire  
.....  
ligne N en commentaire */
```
- *Exemple1 :*

```
// Nous allons calculer le prix TTC du produit  
$prixTTC = $prix + $TVA
```
- *Exemple2 :*

/ *

Cette partie du code permet de récupérer deux entiers

Les deux entiers sont saisis par l'utilisateur dans un formulaire

`*/`

```
$a= $_GET['nb1'];
```

```
$b= $_GET['nb2'];
```

5. Les opérateurs

Un opérateur est symbole qui permet de manipuler des variables. Il en existe différents types dont les principaux sont détaillés ci-dessous.

- Les opérateurs de calcul

| <i>Symbole</i> | <i>Action</i> |
|----------------|-------------------------------|
| + | Addition |
| - | Soustraction |
| / | Division |
| * | Multiplication |
| = | Affectation |
| % | Modulo (Reste de la division) |

Les opérateurs de comparaison

| <i>Symbole</i> | <i>Appellation</i> | <i>Action</i> |
|----------------|-----------------------------|--|
| == | Comparaison | Vérifie si 2 valeurs sont égales. A ne pas confondre avec "=" qui permet d'affecter une valeur à une variable. |
| === | Égalité stricte (identique) | Renvoie un booléen pour indiquer l'égalité à la fois de la valeur et du type. |
| != | Différent | Vérifie si 2 valeurs sont différentes. |
| !== | Différent strict | Renvoie un booléen pour indiquer la différence ou de la valeur ou du type |

| <i>Symbole</i> | <i>Appellation</i> | <i>Action</i> |
|----------------|--------------------|--|
| <i>OR</i> | OU logique | Vérifie qu'au moins une des conditions est réalisée. |
| | OU logique | Autre écriture possible pour <i>OR</i> . |
| <i>AND</i> | ET logique | Vérifie que les 2 conditions sont réalisées. |
| && | Et logique | Autre écriture possible pour <i>AND</i> . |

- | <i>Symbole</i> | <i>Action</i> |
|----------------|------------------------------------|
| . (point) | Concatène 2 chaînes de caractères. |

■ ■ ■ ■ ■ ■ ■ ■ ■ ■

Mon université avec moi, partout et à tout moment.

24


```
$signe = "Capricorne";

// Ici j'écris mes 2 variables d'un coup
// et j'ajoute un espace entre elles grâce à 2 concaténations:
echo ( $age." ".$signe);

?>
```

Les opérateurs d'assignation

Ces opérateurs permettent de simplifier des opérations telles que : ajouter une valeur dans une variable et stocker le résultat dans la variable.

Une telle opération s'écrirait habituellement de la façon suivante

par exemple: $x = x + 2$

Avec les opérateurs d'assignation il est possible d'écrire cette opération sous la forme suivante: $x += 2$

| Opérateur | Effet |
|-----------|---|
| += | addition deux valeurs et stocke le résultat dans la variable (à gauche) |
| -= | soustrait deux valeurs et stocke le résultat dans la variable |
| *= | multiplie deux valeurs et stocke le résultat dans la variable |
| /= | divise deux valeurs et stocke le résultat dans la variable |
| %= | donne le reste de la division deux valeurs et stocke le résultat dans la variable |
| .= | Concatène deux chaînes et stocke le résultat dans la variable |

Les opérateurs d'incrément

Ce type d'opérateur permet de facilement augmenter ou diminuer d'une unité une variable. Ces opérateurs sont très utiles pour des structures telles que des boucles, qui ont besoin d'un compteur (variable qui augmente de un en un).

Un opérateur de type $x++$ permet de remplacer des notations lourdes telles que

$x = x + 1$ ou bien $x += 1$

| Opérateur | Dénomination | Effet | Syntaxe | Resultat (avec x valant 7) |
|-----------|----------------|----------------------------------|---------|----------------------------|
| ++ | Incrémentation | Augmente d'une unité la variable | \$x++ | 8 |
| -- | Décrémentation | Diminue d'une unité la variable | \$x-- | 6 |

Opérateur ternaire

L'opérateur ? permet de renvoyer une valeur ou une autre en fonction d'un test. Par exemple :

```
< ?php $appreciation = ($note>10) ? "bon" : "mauvais"; ?>
```

qui est équivalent au bloc PHP suivant :

```
< ?php
if ($note>10)
    $appreciation="bon";
else
    $appreciation="mauvais";
?>
```

6. Les fonctions mathématiques

Quelques fonctions mathématiques sont :

- *abs* : Valeur absolue. *Ex* : \$a = abs(-7) ; // \$a contient 7
- *base_convert* : Convertit un nombre entre des bases arbitraires.

EX :

```
<?php
$hexadecimal = 'a37334';

echo base_convert($hexadecimal, 16, 2);//convertit de base 16 en 2
?>
```

Affiche 101000110111001100110100

- *bindec* : Convertit de binaire en décimal. *Ex*: `echo bindec('110011')` ; // 51
- *ceil* : Arrondit au nombre supérieur. *Ex*: `echo ceil(4.3)`; // affiche 5
- *cos* : Cosinus. L'angle est en radian.

- *decbin* : Convertit de décimal en binaire. *Ex* : `echo decbin(12)`
- *dechex* : Convertit de décimal en hexadécimal
- *deg2rad* : Convertit un nombre de degrés en radians. *Ex* : `echo deg2rad(45)`
- *exp* : Calcul l'exponentielle de e. *Ex* : `echo exp(12)`
- *floor* : Arrondit à l'entier inférieur. *Ex* : `echo floor(9.99); // 9`
- *hexdec* : Convertit de hexadécimal en décimal
- *intdiv* : division entière. *Ex* : `echo intdiv(7, 2) ; // affiche 3`
- *log* : Logarithme naturel (népérien)
- *pi* : Retourne la valeur de pi
- *pow* : Calcule la puissance. *Ex* : `echo pow(2, 3) ; // affiche 8`
- *rad2deg* : Conversion de radians en degrés
- *rand* : Génère une valeur aléatoire.
Ex :

```
<?php
$r= rand() ; echo $r ;

echo rand(5,50); // génère une valeur aléatoire comprise entre 5 et
50

?>
```

- *round* : Arrondit un nombre à virgule flottante.

Ex :

```
<?php
echo round(3.4); // 3
echo round(3.5); // 4
echo round(3.6); // 4
echo round(1.95583, 2); // 1.96
echo round(1241757, -3); // 1242000

?>
```

- *sin* : Sinus
- *sqrt* : Racine carrée. *Ex* : `echo sqrt(16) ; // affiche 4`
- *tan* : Tangente

7. Les inclusions de fichiers

Dans la pratique, lorsque vous créez un site web, les morceaux de contenu qui sont communs à toutes vos pages sont enregistrés dans différents fichiers (entête de page, pied de page, menu principal...). Lorsque vous créez une nouvelle page vous pouvez appeler ces fichiers et les replacer au bon endroit grâce à la fonction `include()` ou `require()`.

```
<!DOCTYPE html>

<html>

<head>

<title>Titre de la page</title>

</head>

<body>

    <?php include("entete_inc.php"); ?>

    <?php include("menu_inc.php"); ?>

    <p>contenu html</p>

    <?php include("pied_inc.php"); ?>

</body>

</html>
```

Aux endroits où les fichiers sont inclus, PHP recopiera le contenu de chaque fichier appelé. Cela permet de gagner du temps et surtout de pouvoir répercuter sur toutes les pages les modifications qui seront apportées par la suite sur les parties communes.

- *Différence entre include() et require()*

La différence importante entre `require()` et `include()` se situe au niveau de la gestion d'erreur. `Include()` renvoi qu'un warning en cas d'erreur, alors que `require()` provoquera une erreur fatale() qui empêchera l'exécution des scripts qui suivront.

- Inclure avec `include_once()` et `require_once()`

Il existe deux autres fonctions pour inclure des fichiers, *include_once()* et *require_once()*. Le comportement est similaire à *include()* et *require()*, mais la différence est que si le code a déjà été inclus, il ne le sera pas une seconde fois. Cette méthode permet d'éviter la déclaration de classes ou de fonctions qui engendre des erreurs.

Lien utile : <https://openclassrooms.com/fr/courses/918836-concevez-votre-site-web-avec-php-et-mysql/4239271-inclure-des-portions-de-page>

8. Travaux dirigés

Affichage texte formaté

Libellé

Ecrire un script PHP qui affiche un texte contenu dans une variable.

Le texte doit être affiché en grand titre, en rouge et centré.

Correction

- Première méthode

```
<?php
$texte="Bienvenue aux étudiants";
echo "<h1 style='color:red; text-align:center;'> $texte </h1>";
?>
```

On pourrait aussi faire comme ça :

```
<?php
$texte="Bienvenue aux étudiants";
echo "<h1 style='color:red;'> <center> $texte </center> </h1>";
?>
```

Explication du code :

- Pour formater un texte à afficher à l'écran en PHP, on utilise les balises comme en HTML
- On peut à ces balises définir des attributs et du css comme il en ait en HTML
- Dans le deuxième exemple, au lieu d'utiliser une propriété css text-align pour centrer le texte, on a préféré utiliser la balise HTML <center> </center> pour le faire. Puis, au lieu d'écrire tout le texte sur une ligne, on a fait des retours à la ligne

Calcul mathématique

Libellé

On considère la formule ci-dessous :

$$(x + a)^n = \sum_{k=0}^n C_n^k x^k a^{n-k}$$

On aimerait pour trois variables \$x\$, \$a\$ et \$n\$ calculer \$(x + a)\$ puissance \$n\$ et afficher le résultat à l'écran. Toutefois, avant d'afficher le résultat :

- On doit d'abord afficher la formule ci-dessus en gras, centrée, et on l'encadre

- Puis on réaffiche à nouveau la formule, mais en remplaçant les inconnues x , a , et n par les variables x , a et n .
- Enfin, on affiche le résultat du calcul à l'ordre 2, en gras, en rouge, taille 28px, police helvetica, souligné, et on la fait défiler à l'écran.

Exercice



0. Exercice

Donnez un symbole représentant la fonction OU logique en PHP

0. Exercice

On aimerait diviser \$x par \$y et mettre le reste de la division dans \$r. Veuillez compléter le code. \$r= \$x
_____ \$y

0. Exercice

Pour définir une constante pi de valeur 3.14, on fait : `const pi = 3.14` ("pi",3.14)

0. Exercice

\$nom et \$prenom sont deux variables chaînes de caractères qu'on aimerait concaténer. veuillez compléter le code : \$nomentier= \$nom \$prenom

0. Exercice

\$x\$ et \$y\$ sont deux variables contenant des valeurs entières, on aimerait additionner \$x\$ et \$y\$ puis stocker le résultat dans \$x\$. Veuillez compléter l'instruction : `$x` `$y`

0. Exercice

\$x\$ est une variable de type réel (float), on aimerait convertir \$x\$ en entier (integer) puis stocker le résultat dans \$y\$. veuillez compléter l'instruction : `$y = (\quad)$x`

0. Exercice

Pour inclure une page php nommée entete.php dans une autre page, on fait : `xxx("entete.php")`, où `xxx` représente une fonction, laquelle ?

0. Exercice

\$x\$ est une valeur décimale avec plusieurs chiffres après la virgule. on aimerait afficher \$x\$ à l'ordre 2.
Veuillez compléter l'instruction : `echo _____ ($x, 2)`

Les structures de contrôle



Une structure de contrôle est un ensemble d'instructions qui permet de contrôler l'exécution du code.

Il en existe essentiellement deux types :

- Les structures de contrôle conditionnelles qui permettent d'exécuter certaines parties du code si une condition spécifique est remplie.
- Les structures de contrôle de boucle qui permettent d'exécuter en boucle certaines parties du code (généralement jusqu'à ce qu'une condition soit remplie).

1. Les structures conditionnelles

La structure conditionnelle simple if

- *Syntaxe*

```
<?php
if (condition) {
instructions au cas où la condition est réalisée;
}
//si la condition n'est pas réalisée, il ne se passe rien
?>
```

- *Example*

```
<?php
if ($a > $b) {
echo "a est plus grand que b";
$b = $a;
}
?>
```


*La structure conditionnelle alternative if ... else**- Syntaxe*

```
<?php
if (condition) {
instruction au cas où la condition serait réalisée;
} else {
instruction au cas où la condition ne serait pas réalisée;
}
?>
```

- Exemple

```
<?php
if ($a > $b) {
echo "a est plus grand que b";
} else {
echo "a est plus petit que b";
}
?>
```

*La structure conditionnelle imbriquée else if / elseif**- Syntaxe*

```
<?php
if(condition 1){
code 1;
}elseif(condition 2){
code 2;
}elseif(condition 3){
code 3;
}
...
else
{ code par défaut; }
?>
```

- *Example*

```
<?php
if ($a > $b) {
    echo "a est plus grand que b";
} elseif ($a == $b) {
    echo "a est égal à b";
} else {
    echo "a est plus petit que b";
}
?>
```

La structure à choix multiple switch

- *Syntaxe*

```
<?php
switch (variable)
{
case valeur_1 : instructions1;
break;

case valeur_2 : instructions2;
break;

....

case valeur_N : instructionsN;
break;

default : instructions;
}
?>
```

- *Example*

```
<?php
$nom = "LA GLOBULE";
switch ($nom) {
case 'Jean' : echo 'Votre nom est Jean.';
break;
```

```

case 'Benoît' : echo 'Votre nom est Benoît.';
break;

case 'LA GLOBULE' : echo 'Votre nom est LA GLOBULE.';
break;

default : echo 'Je ne sais pas qui vous êtes !!!';
}

?>

```

2. Les structures itératives

La boucle while

while est un mot anglais signifiant "tant que" en français. Le programme exécute une routine tant que la condition est vraie.

- Syntaxe

```

<?php
while (condition) {
instructions;
}

?>

```

- Exemple

```

<?php
$i = 1;
while ($i <= 10) {
echo $i;
$i++;
}

?>

```

La boucle do ... while

idem avec premier passage obligatoire : la condition est vérifiée en fin de bloc.

- Syntaxe

```

<?php
do while (condition) {

```

```
instructions;
}

?>
```

- *Example*

```
<?php
$i = 1;

do {

echo $i;

$i++;

}while($i <= 10)

?>
```

La boucle for

for est un mot anglais signifiant "pour" en français. Le programme exécute une routine pour des valeurs d'une variable qui vérifient une certaine condition.

- *Syntaxe*

```
<?php
for(initialisation ; condition d'arrêt ; incrémentation) {
instructions;
}
?>
```

Attention la structure dans la parenthèse est `for(; ;)`

- *Example*

```
<?php
for ($i = 1; $i <= 10; $i++) {
echo $i;
}
?>
```

NB : Il existe une boucle *foreach*, mais nous la verrons un peu plus loin avec les tableaux.

3. Travaux dirigés

Table de multiplication

Libellé

Écrire un programme permettant de faire la table de multiplication d'un entier x.

Correction

- *Première méthode*

```
<?php
$x=5;
for($i=1;$i<=10;$i++){
echo $x." X ".$i." = ".$x*$i." <br/> ";
}
?>
```

Explication du code :

- La balise
 permet de faire un retour à la ligne
- Les points (.) permettent de faire la concaténation, c.à.d. coller ou additionner les chaînes de caractères
- Les caractères ou chaîne de caractère sont entre quote (' ') ou entre double quote (" ").
- Les variables sont en dehors des quotes ou doubles quotes. Mais il est possible de les mettre dans des quotes ou doubles quotes en PHP grâce au symbole \$.

Le code va afficher à l'écran

```
5 X 1 = 5
5 X 2 = 10
.....
5 X 10 = 50
```

- *Deuxième méthode*

```
<?php
$x=5;
for($i=1;$i<=10;$i++){
    $resultat= $x*$i;
    echo "$x X $i = $resultat <br/>";
}
?>
```

Explication du code :

- Ici, on décide de mettre les variables dans des doubles quotes. Toute fois le calcul $\$x*\i ne peut pas se faire dans des quotes, donc on fait d'abord ce calcul et on le stocke dans une variable `$resultat` avant d'afficher le message.
- *Deuxième méthode*

```
<?php
$x=5;
$i=1;
while($i<=10){
    $resultat= $x*$i;
    echo "$x X $i = $resultat <br/>";
    $i++;
}
?>
```

Explication du code :

- Dans cette approche, au lieu d'utiliser une boucle for, nous utilisons une boucle while. Attention avec les boucles while, n'oubliez jamais d'incrémenter le compteur (faire \$i++), car le compteur ne s'incrmente pas automatiquement comme les boucles for. Si vous oubliez, vous aller créer une boucle infinie.

Prénom baoulé

Libellé

Écrire un programme permettant d'afficher le prénom baoulé d'un individu à partir de son jour de naissance et de son sexe stockés dans des variable \$jour et \$sexe.

| Jour | Prénom | |
|----------|---------|-----------------|
| Sexe | m | f |
| lundi | Kouassi | Akissi |
| mardi | Kouadio | Adjoua |
| mercredi | Konon | Amenan ou Amlan |
| jeudi | Kouakou | Ahou |
| vendredi | Yao | Aya |
| samedi | Koffi | Affoué |
| dimanche | Kouamé | Amin |

Correction

Il existe plusieurs méthodes possibles, nous allons ici montrer une méthode

- *Explication du code*

Ici, nous avons travaillé avec des structures conditionnelles imbriquées. Toutefois, il est possible de travailler uniquement avec des structures à choix multiples, ou-bien de faire une combinaison des deux.

The logo consists of three overlapping squares in shades of green and purple, with the Roman numeral 'XXII' in white on the largest, front-most square.

0. Exercice

On aimerait écrire un script qui calcule le pgcd de deux entiers a et b puis affiche le résultat. Veuillez compléter le code :

```
<?php
$a=12 ;
$b=8 ;

while ( $i=1 while $i<=$a while $i++){
if( ($a while $i==0) && ($b%$i while 0) ) { $pgcd=$i ; }
}

echo "Le pgcd de $a et $b est : $pgcd" ;

?>
```

0. Exercice

On aimerait écrire un programme qui calcule la somme des nombres pairs non nul compris entre 0 et 100 et affiche le résultat. Veuillez compléter le code :

```
<?php
$i=2 ;

$som=0 ;

while ( $i <= 10 ) {

    $som = $som + $i ;

    $i = $i + 1 ;

}

echo "La somme est : " . $som ;

?>
```

0. Exercice

On aimerait écrire un programme qui calcule le factoriel d'un entier n puis affiche le résultat. Veuillez compléter le code.

```
<?php
$n=10 ;
$facto=1 ;
for($i=1 ;$i<= _____ ;$i++) _____
$facto *= _____ ;
_____
_____ "Le factoriel de $n est : $facto" _____
?>
```