

PHP

C'est un langage qui est utilisé pour la réalisation de sites. Il faut donc installer Easyphp pour avoir un serveur local. L'installation fournit également la base de données Mysql. Avoir à sa disposition le traitement de texte dédié à la programmation : Notepad++ est une bonne chose.

Si nous ouvrons le dossier de Easyphp nous trouvons un dossier eds-www c'est dans ce dossier que doivent se trouver tous les sites qui sont créés.

Pour fonctionner Easyphp doit être lancé. Mais on doit aussi lancer tous les serveurs. Un clic sur l'icône qui est apparue dans la zone de notification permet de le faire.

Pour lancer un programme il faut impérativement passer par le navigateur avec l'adresse 127.0.0.1 terminez par ENTREE, le dossier des sites est ouvert cliquez sur le site voulu.

La mise en œuvre n'est pas très complexe mais c'est un passage obligé.

Table des matières

Particularité du langage.....	1	La liste.....	3
L'instruction conditionnelle.....	1	La liste associative.....	3
Le choix.....	2	La variable de session.....	3
La boucle for.....	2	L'instruction include();.....	3
La boucle while.....	2	La fonction.....	3
Quelques fonctions de bases.....	2	Portabilité des variables.....	3
Chaîne de Caractères.....	2	L'attribut global.....	3
Tableau.....	2	L'attribut static.....	4
Nombres.....	2	Procédure.....	4
Style.....	3	Exemple.....	4
Sécurité.....	3	Script 1.....	4
Fichiers.....	3	Script 2.....	4

Particularité du langage

Il est sensible à la casse.

Toutes les instructions se terminent par un point-virgule.

Toute variable est préfixée par \$

Des instructions peuvent être mises sur une même ligne si elles sont séparées par un point-virgule.

`a+=3` signifie que l'on ajoute 3 à a => `a=a+3`

La concaténation se fait par le point.

"La pleine"." est plate" => "La plaine est plate"

les variables ne sont pas typées elles peuvent être employées comme chaîne ou comme nombre.

`$a=3; $b="enfants";`

`$c=$a." ".$b; // renvoie "3 enfants"`

Différence d'utilisation des simples et doubles guillemets.

`echo ('La couleur est $coul');` sera écrit tel quel, la variable ne sera pas traduite.

`echo ("La couleur est $coul");` ici le texte s'affichera avec le nom de la couleur contenue dans \$coul.

L'instruction conditionnelle

`if(x==1) {liste d'instruction1} else {liste d'instruction2}`

La condition est mise en parenthèse, un bloc d'instructions est formé par toutes les instructions qui se trouvent entre les deux accolades.

Si la condition est vraie la liste1 sera exécutée sinon c'est la liste2 qui sera prise en compte.

C'est l'une ou l'autre jamais les deux.

les opérateurs de comparaison sont `== != < <= > >=`

égale, différent, plus petit, plus petit ou égale, plus grand, plus grand ou égale

Les conditions peuvent être liées par des mots AND - OR - NOT

AND dans le sens ensemble l'une et l'autre conditions doivent être vraies pour que l'ensemble le soit.

OR il suffit que l'une des conditions soit vraie pour que l'ensemble le soit.

NOT inverse le résultat ce qui est vrai devient faux.
Les parenthèses peuvent être utilisées pour imposer les priorités.
La liste2 n'est pas obligatoire.

Le choix

```
switch($var){
    case 1: $a='vert'; break;
    case 2: $a='bleu';break;
    default: $a='inconnue';break;
}
```

Si la variable \$var a une valeur autre que 1 ou 2 alors "inconnue" sera pris en compte.
Le mot **break** doit terminer chaque choix.

La boucle for

```
$n=1;
for($i=0;$i<$n;$i++){
    $n1+=1;
    print("$n1 ème tour");
}
```

La boucle while

Pour une liste.

```
foreach( $array as $value )
    echo $value . '<br />';
```

Pour une liste associative

```
$array = array(
    'premier' => 'N° 1',
    'deuxieme' => 'N° 2',
    'troisieme' => 'N° 3');
foreach( $array as $key => $value )
    echo 'Cet élément a pour clé "' . $key . '" et pour valeur "' . $value . '"<br />';
```

Quelques fonctions de bases

Chaîne de Caractères

intval(\$ch);	transforme des caractères en valeur entière
floatval(\$ch);	transforme des caractères en valeur flottante
strlen(\$chn);	donne le nombre de caractères de la chaîne
\$a=substr(\$var,\$deb,\$long);	Extraction (indice zéro => 1ère lettre)
\$a=str_replace("à chercher", "remplacer par", "dans");	
\$a=str_replace(['#','*','&'],[' ','',''],\$c);	
\$a=str_repeat(\$car,\$nb);	Remplace des signes
\$pos=strpos(chaine,sous-chaine);	Répète des signes
	Renvoie la position de la première occurrence

Tableau

explode(':', \$chn);	chaîne avec séparateur, renvoie un tableau
implode(':', \$chn);	tableau, renvoie une chaîne avec séparateur
\$nli=count(\$tb);	donne le nombre d'éléments d'un tableau
var_dump(array);	renvoie les éléments du tableau

Nombres

round(\$val,nbdécimales);	zéro ou rien sont admis pour entier
number_format (\$nombre,2,',.' '');	nombre, décimales, signe Décimal, signe des milliers
\$k=rand(1,15);	renvoi un entier entre 1 et 15

Style

`strtoupper();` met en majuscule
`ucfirst();` Première lettre majuscule

Sécurité

`strip_tags(addslashes($a));` Supprime les balises, et échappe les apostrophes
`stripslashes($a);` enlève les apostrophes
`htmlentities($chn);` Récupère des données avec apostrophe
`htmlspecialchars($chn);` ????

Fichiers

`rename(ancien_nom,nouveau_nom);` Renomme un fichier texte

La liste

```
$animaux= array("Chien","Chat","Cheval");
Nous avons ici une liste.
Le premier indice est toujours zéro.
echo $animaux[1]; renvoie "Chat"
$animaux[2]='âne';
la liste devient "Chien, Chat, âne".
```

La liste associative

```
malist=array(
    légume1 => Carottes,    peut aussi s'écrire =>
    légume2 => Choux,
    légume3 => Navets
)
```

```
malist=[
    légume1 => Carottes,
    légume2 => Choux,
    légume3 => Navets
]
```

Nous avons une liste associative ou un tableau à double entrée. `légume1` représente la clef (key) Carottes la valeur de la clef "légume1"
`echo "malist['légume1']";` // renverra "Carottes"

La variable de session

C'est une variable qui va durer pendant toute la session. Elle est reconnue dans toutes les pages visitées. C'est très pratique pour passer une valeur d'une page à l'autre.

L'instruction `include()`;

Elle permet d'incorporer des lignes de code dans un script. C'est une instruction pratique.

La fonction

C'est l'élément qui va permettre de définir un processus simple ou complexe ne revoyant aucune valeur ou, au contraire, renvoyant une valeur. On doit garder à l'esprit qu'une fonction doit être définie avant son premier emploi. Elles s'écrivent en tête des pages quand elles sont définies dans une page d'utilisation.

Dès que l'on parle de fonction une nouvelle notion s'impose celle de la visibilité des variables.

Portabilité des variables

Une variable utilisée dans la page est visible pendant toute la durée de la page mais uniquement dans la page, pas dans les fonctions. Cela se nomme la portée des variables.

En conséquence une variable portant le même nom dans la page et dans la fonction sont deux variables distinctes et étrangères.

L'attribut global

On peut y remédier en déclarant cette variable "global" Dans ce cas la variable est reconnue dans la fonction mais en plus si elle est modifiée par cette fonction, elle le restera à son retour. La particularité est que "global" peut servir d'initialisation à NULL. Ce qui est assez pratique.

L'attribut static

Un paramètre de fonction déclaré "static" aura la particularité de conserver sa valeur d'un appel à l'autre de la fonction.

Procédure

Ce terme n'existe pas en PHP. Une fonction en PHP est tantôt procédure, tantôt fonction. En principe une procédure ne renvoie rien contrairement à une fonction. Or une fonction en PHP peut ne rien renvoyer comme renvoyer une valeur. Tout dépend du programmeur.

Exemple

postulats : bébé 0-2 ans / enfant 2ans-13ans / adolescent 13ans-20 ans / adulte au-dessus
Limite supérieur incluse, limite inférieure exclue. (A 2 ans 1 mois on est un enfant)

Script 1

```
//-----  
function statut($age){  
    if($age<=$li[0]) $r='Bébé';  
    if($age>$lim[0] and $age<=$lim[1]) $r='Enfant';  
    if($age>$lim[1] and $age<=$lim[2])  
$r='Adolescent';  
    if($age>$lim[3]) $r='Adulte';  
}  
$lim=list(0,2,13,20);  
$age=32;  
echo "statut($age)";  
//-----
```

Script 2

```
//-----  
function statut($age){  
    global $lim;  
    if($age<=$lim[0]) $r='Bébé';  
    if($age>$lim[0] and $age<=$lim[1]) $r='Enfant';  
    if($age>$lim[1] and $age<=$lim[2])  
$r='Adolescent';  
    if($age>$lim[3]) $r='Adulte';  
}  
$lim=list(0,2,13,20);  
$age=32;  
echo "statut($age)";  
//-----
```

Un script fonctionne pas l'autre. Pourquoi ?