Notions de base

en programmation web

avec PHP

Introduction

Internet et les pages web

HTML est un langage de description de page et non pas un langage de programmation. Il permet de concevoir des pages destinées à être publiées sur Internet.

Page html : contient le texte à afficher et des instructions de mise en page => Il n'y a pas d'instructions de calcul

Pour avoir des sites de plus en plus riches en informations

- Il est nécessaire d'améliorer régulièrement le contenu de sites
- Faire des mises à jour manuelles trop complexes

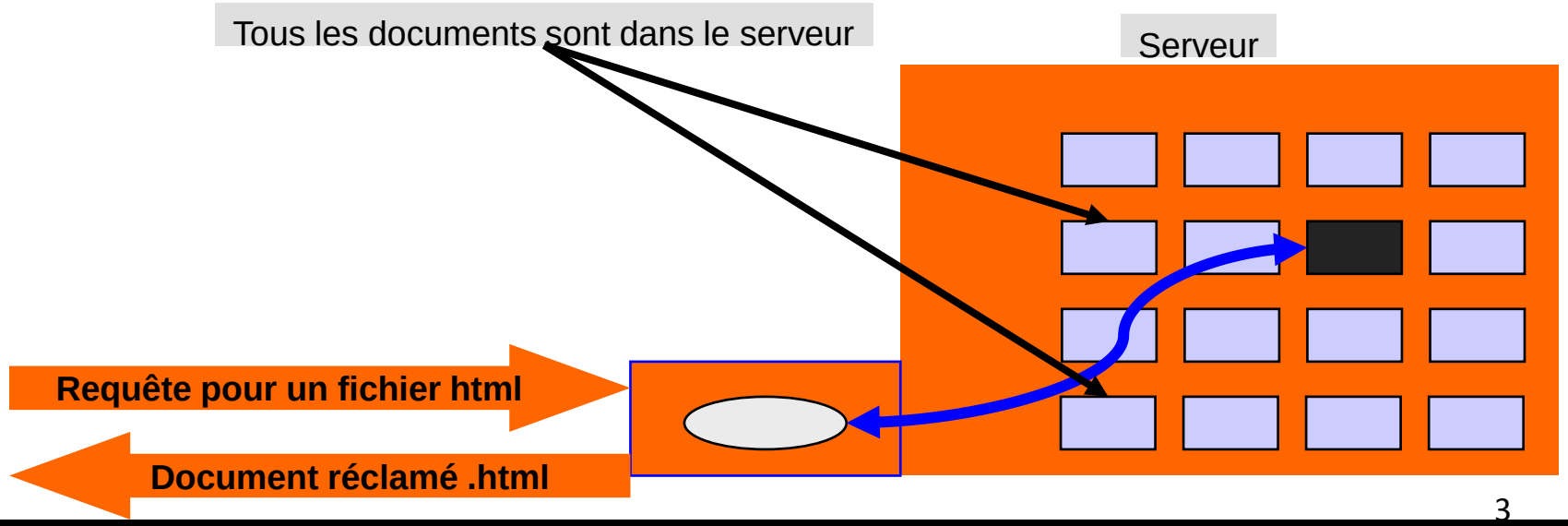
Qusetion => Pourquoi ne pas automatiser les mises à jour ?

Introduction

Fonctionnement des pages web statiques

Leurs contenus ne changent ni en fonction du demandeur ni en fonction d'autres paramètres éventuellement inclus dans la requête adressée au serveur. Toujours le même résultat.

- Rôle du serveur : localiser le fichier correspondant au document demandé et répond au navigateur en lui envoyant le contenu de ce fichier

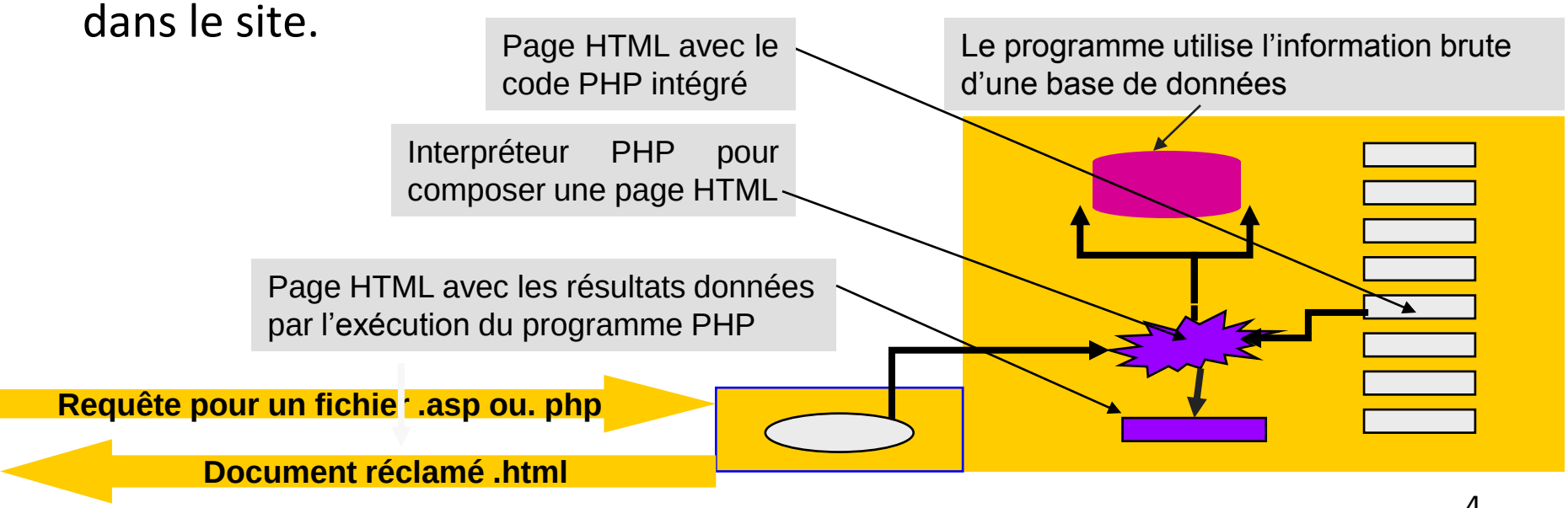


Introduction

Fonctionnement des pages web dynamiques

Dans un site dynamique : le contenu et la forme sont séparés. Le contenu est rangé dans une base de données et il est extrait pour s'afficher dans un modèle de page conçu par le webmaster.

Le webmaster est celui qui va concevoir les modèles de pages et écrire les scripts qui assurent le bon fonctionnement de la navigation dans le site.



Introduction

Pages web dynamiques côté serveur ou côté client

Langage côté client :

- Traité par la machine qui accueille le logiciel de navigation.
- Ses résultats peuvent varier en fonction de plate-forme utilisée. Un programme en JavaScript pourra fonctionner sous Netscape et poser problème sous Internet explorer.
- Les résultats peuvent être différents suivant la machine (PC, Mac)
- Ne permettent pas de masquer les sources du programme
- Sont indépendants du serveur et donc de l'hébergement
- Exemples: JavaScript

Introduction

Pages web dynamiques côté serveur ou côté client

Langage côté serveur :

- Le travail d'interprétation du programme est réalisé par le serveur
- Sont indépendants de la machine et du logiciel de navigation utilisés pour la consultation.
- Sont compatibles avec tous les navigateurs et toutes leurs versions.
- Permettent de masquer les sources de ses programmes
- Exemples: ASP, JSP, PHP….

Initiation à PHP

PHP est un langage de programmation informatique essentiellement utilisé pour produire à la volée des pages web dynamiques. Dans sa version 5 lancée en juillet 2004, PHP s'est imposé comme le langage de référence sur le web en raison de sa simplicité, de sa gratuité et de son origine de logiciel libre.

- Connaît un succès toujours croissant sur le Web et se positionne comme un rival important pour ASP
- Combiné avec le serveur Web Apache et la base de données MySQL, PHP offre une solution particulièrement robuste, stable et efficace
- Gratuité : Tous les logiciels (EasyPHP, WampServer) sont issus du monde des logiciels libres (Open Source).
- PHP est un langage imbriqué dans le code HTML, il est interprété par un module « spécial » par le serveur web.
- Un fichier HTML contenant un script PHP doit avoir l'extension ".php".

Initiation à PHP

- PHP 1.0 (Personnal Home Page), 1994-1995
- PHP/FI 2.0. 1995-1997
- PHP3, 1997-2000

  La version 3 a permis une explosion de l'utilisation de PHP.
- PHP4 (Zend), 2000
  - *Moteur ZEND : le « cerveau » de PHP*
  - *Développé par la compagnie Zend, les créateurs de PHP.*
- PHP5 Juillet 2004

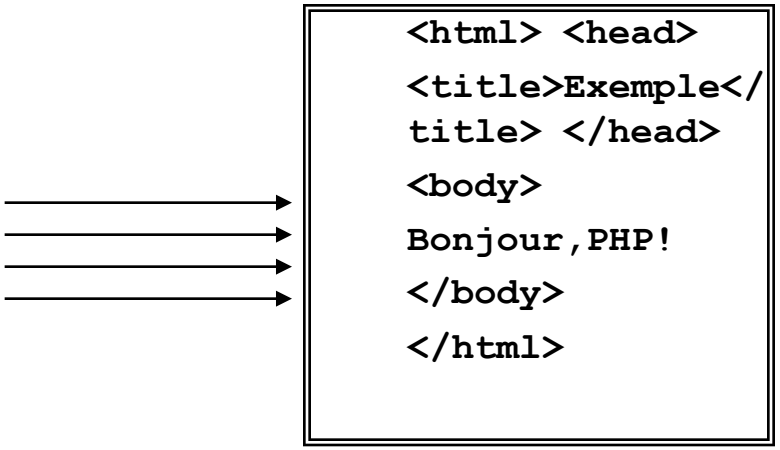
Initiation à PHP

Syntaxiquement, le langage PHP est un mélange de C, Java et Perl.
PHP est un acronyme récursif "Hypertext Preprocessor "

Exemple

```
<html>    <head>
<title>Exemple</title></head>
<body>
<?php
echo ("Bonjour, PHP!");
?>
</body>
</html>
```

Source côté client



Initiation à PHP: intégration du script dans HTML

Un script PHP peut comprendre à la fois du code PHP et du code HTML, non interprété. On doit donc encadrer les parties comportant le code PHP entre 2 balises **<?php** et **?>**. Le reste de la page n'est pas interprété.

```
<html>
<body>
<h1><?php   echo "Bonjour " ;  ?></h1>
<?php
echo "<br/> "  ;
echo  "<b> Tout le monde </b>";
 ?>
</body>
</html>
```

Note: On peut également utiliser les balises **<script language="php">** et **</script>**

Initiation à PHP: les commentaires

- Le langage PHP est composé d'instructions terminées par **;**
- Bloc d'instructions délimité par des **{}**
- Les commentaires sont soit précédés de **//** ou **#** :
  - // Commentaires de fin de ligne
  - soit entourés de /* et */ :
  - /* Commentaires longs */

```
<html>
<head>
<title>Exemple</title>
</head>
<body>
<? php
# La ligne suivante est une instruction PHP
echo "Bonjour, je suis un script PHP!";
?>
</body>
</html>
```

Initiation à PHP: Afficher du texte

La fonction echo affiche un (ou plus) argument.

```
echo 'Hello, World';
echo "Hello, World ";
```

Avec le quotte double ", les variables contenues dans cette chaîne sont interprétées. Avec un simple quotte , les variables contenues dans cette chaîne seront affichées tel quelles.

```
$nom= "Toto";
echo "Hello $nom";
echo 'Hello $nom';
```

Initiation à PHP: Les variables

Visibilité et affectation

PHP n'est pas un langage fortement structuré, il ne contient donc pas de partie déclarative clairement définie. Pour définir une variable, il suffit de l'initialiser.

Les variables sont précédées du signe $, quelque soit leur type. Ainsi pour déclarer une variable var :

```
$var=1;
```

La variable $var est alors définie et vaut 1. Elle devient immédiatement accessible et ce jusqu'à la fin du script.

Initiation à PHP: Les variables

Type de variables

Le typage des variables est implicite en php. Il n'est donc pas nécessaire de déclarer leur type au préalable ni même de les initialiser avant leur utilisation.

C'est PHP qui décide de son type lors de l'affectation.

Il existe six types de données :

- ✓Entier *(int, integer)*
- ✓Décimal *(real, float, double)*
- ✓Chaîne de caractères *(string)*
- ✓Tableau *(array)*
- ✓Objet *(object)*
- ✓Booléen *(boolean, uniquement PHP4)*

Initiation à PHP: Les variables

Type de variables

Il est parfois utile de forcer le type d'une variable. On utilise la fonction settype ou bien les opérateurs de casting à savoir (int) et (string).

La fonction settype renvoie vrai si la conversion a fonctionné, faux sinon.

```
$a= 3.1415;
$result= settype( $a, "integer" );     // => $a = 3  , $result = 1

 $ str = ’’12’’;   //  $ str  vaut la chaîne ‘’12’’
 $nbr = (int)$str; // $nbr vaut le nombre 12
```

=> Le cast est une conversion de type. L’action de caster consiste en convertir une variable d’un type à un autre

Initiation à PHP: Les variables

Type de variables

D'une manière générale, les opérateurs de conversion sont:

- ✓(string) conversion en chaîne de caractères
- ✓(int) conversion en entier, synonyme de (integer)
- ✓(real) conversion en double, synonyme de (double) et (float)
- ✓(array) conversion en tableau
- ✓(object) conversion en objet
- ✓(bool) conversion en booléen

Initiation à PHP: Les variables

Fonctions prédéfinies

Quelques fonctions:

- ✓empty($var) : renvoie vrai si la variable est vide
- ✓isset($var) : renvoie vrai si la variable existe
- ✓unset($var) : détruit une variable
- ✓gettype($var) : retourne le type de la variable
- ✓settype($var, ''type'') : convertit la variable en type type (cast)
- ✓is_long(), is_double(), is_string(), is_array(), is_object(), is_bool(), is_float(), is_numeric(), is_integer(), is_int()…

Initiation à PHP: Les variables

Tests sur les variables

La fonction isset permet de tester si une variable est définie.

La fonction unset permet de supprimer la variable, et de désallouer la mémoire utilisée.

```
$a= " ";
echo isset($a);   // => 1 (vrai)
unset($a);
echo isset($a);
```

La fonction gettype permet de connaître le type de la variable. Elle renvoie une chaîne : ''string'' ou ''integer'' ou ''double'' ou ''array'' ou ''object''.

```
$a= 12;
echo gettype($a) ; // => "integer"
$a= $a / 10;
echo gettype($a) ; // => "double"
unset($a);
echo gettype($a) ; // => "string "
```

Remarque : Si la variable n'est pas définie, elle renvoie "string".

Initiation à PHP: Les constantes

L'utilisateur peut définir des constantes dont la valeur est fixée une fois pour toute. Les constantes ne portent pas le symbole **$** (dollars) en début d'identificateur et ne sont pas modifiables.

PHP permet de définir des constantes à l'aide de la fonction:

define("var",valeur)

```
define(" author ", " Foobar ");
echo author; // affiche 'Foobar'

define(" annee" ,1980);
echo annee; // affiche 1980
```

Contrairement aux variables, les identificateurs de constantes (et aussi ceux de fonction) ne sont pas sensibles à la casse.

Initiation à PHP: Les opérateurs

PHP dispose des opérateurs classiques inspirés des langages C et Perl.

Comparaison

Exemple	Nom
$a == $b	Égal
$a != $b	Différent
$a < $b	Strictement inférieur
$a <= $b	Inférieur ou égal
$a > $b	Strictement supérieur
$a >= $b	Supérieur ou égal

Initiation à PHP: Les opérateurs

PHP dispose des opérateurs classiques inspirés des langages C et Perl.

Logiques

Les opérateurs logiques sont utilisés dans les tests, par exemple dans un « if (condition) »

Exemple	Nom
$a and $b $a && $b	ET
$a or $b $a || $b	OU
$a xor $b	OU Exclusif
! $a	NON

Initiation à PHP: Les opérateurs

PHP dispose des opérateurs classiques inspirés des langages C et Perl.

Arithmétique

Exemple	Nom	Résultat
$a + $b	Addition	Somme de $a et $b
$a - $b	Soustraction	Différence de $a et $b
$a * $b	Multiplication	Produit de $a et $b
$a / $b	Division	Quotient de $a et $b
$a % $b	Modulo	Reste de $a divisé par $b

Remarque : l'opérateur / renvoie un entier si les 2 opérandes sont des entiers, sinon il renvoie un flottant.

Initiation à PHP: Les opérateurs

Les **opérateurs** incrémentaux

Exemple	Nom
$a++	Post-incrémente
++$a	Pré-incrémente
$a--	Post-décrémente
--$a	Pré-décrémente

Les opérateurs d'assignation et d'autres

Exemple	Fonction
$a = $b ; $a = 4 ; $a = "foo" ;	Affectation de la valeur de l'expression de droite dans la variable de gauche.
"bonjour "."Monde!"; $b . $c; $b . "ajout";	Concaténation de chaînes de caractères.
+=, -=, *=, /=, .=	Opérateurs d'assignation combinés

Initiation à PHP: Les chaines de caractères

Opérateur de concaténation de chaînes : . (point)

Exemple1:

```
$foo = "Hello";
$bar = "Word";
echo $foo.$bar;
```

Exemple2:

```
$name = "Henry";
$whoiam = $name. "IV ";
```

Exemple3:

```
$out = "Patati";
$out .= "et patata…";
```

Initiation à PHP: Les chaines de caractères

Quelques fonctions:

- ✓length($str) : retourne le nombre de caractères d'une chaîne
- ✓strtolower($str) : conversion en minuscules
- ✓strtoupper($str) : conversion en majuscules
- ✓trim($str) : suppression des espaces de début et de fin de chaîne
- ✓substr($str,$i,$j) : retourne une sous chaîne (entre les positions i et j)
- ✓strnatcmp($str1,$str2) : comparaison de 2 chaînes
- ✓addslashes($str) : Ajoute des slash dans une chaîne, à la mode du langage C
- ✓ord($char) : retourne la valeur ASCII du caractère

Initiation à PHP: Les chaines de caractères

Les autres fonctions d'affichage :

✓echo() : écriture dans le navigateur

✓print() : écriture dans le navigateur

✓printf([$format, $arg1, $arg2]) : écriture formatée comme en C, i.e. la chaîne de caractère est constante et contient le format d'affichage des variables passées en argument

Exemple1

```
echo  "Bonjour $name";
print("Bonjour $name");
printf("Bonjour %s", $name);
```

Les structures de contrôle

Les structures de contrôle

Les conditions sont les éléments les plus utilisés dans n'importe quel langage. Grâce aux conditions, vous allez pouvoir effectuer des actions telles que « Si le visiteur est un homme, afficher 'Bonjour Monsieur', sinon afficher 'Bonjour Madame‘ ».

On utilise quelques opérateurs très simples en PHP pour comparer deux variables.

Symbole	Signification
$a == $b	$a est égal à $b
$a < $b	$a est inférieur à $b
$a > $b	$a est supérieur à $b
$a <= $b	$a est inférieur ou égal à $b
$a >= $b	$a est supérieur ou égal à $b
$a != $b	$a est différent de $b
$a <> $b	$a est différent de $b

Les structures de contrôle: IF

Les tests IF

Syntaxes :

Test if " basique " :	Test if-else :	Test if-elseif :

if([condition])
{
...
}
	if([condition])
	{
	...
	}
	else {
	...
	}
		if([condition])
		{...}
		elseif([condition])
		{
		...
		}

Dans le cas de plusieurs tests successif portant sur une même variable, on utilisera plutôt le test switch.

Remarque : Si le corps du test ne comporte qu'une instruction, les accolades { } sont optionnels, (contrairement au Perl).

Les structures de contrôle: IF

La structure if / else est la base en PHP. Grâce à elle, vous allez pouvoir effectuer deux choses : une action si la condition est vraie, et une autre action si elle est fausse. On traduit le if par "si" et le else par "sinon".

Exemple1:

```
<?php
    $age = 12;
    if($age >12)  echo 'vous êtes un adolescent';
?>
```

Exemple2

```
<?php
    $age = 12;
    if($age <18)   echo 'vous êtes un enfant ;
    else  echo 'vous êtes un adolescent';
?>
```

Les structures de contrôle: IF

Exemple3:

```
<?php
    $age = 12;
    if($age <18)   echo 'vous êtes un enfant ;
    elseif ($age>18  &&  $age<30)  echo 'vous êtes un adolescent';
    else  echo 'vous êtes un adulte';
?>
```

Exemple4:

```
<?php
    $nombre1 = 12;
    $nombre2 = 13;
    if($nombre1 == $nombre2)  echo 'Les valeurs sont égales';
    else    echo 'Les valeurs ne sont pas égales';

?>
```

Les structures de contrôle: IF

Exemple5:

Les conditions multiples permettent de donner plusieurs conditions pour effectuer une ou plusieurs actions. Par exemple, si on souhaite que les visiteurs aient plus de 13 ans pour visiter le site et qu'ils soient des hommes, on doit avoir une condition de ce style :

```
<?php
   $homme = FALSE;    //il s'agit donc d'une femme
   $age   = 17;
 if($homme ==TRUE AND $age > 13) //Le visiteur est un homme et agé de plus de 13 ans
   {
       echo 'Vous pouvez visiter le site';
   }
   else //Le visiteur est une femme ou alors il a moins de 13 ans
   {
       echo 'Vous ne pouvez pas visiter le site';
   }
?>
```

Les structures de contrôle: SWITCH

Le test SWITCH

Le switch permet de confronter une variable à plusieurs valeurs prédéfinies.

Il permet un code plus compact et lisible qu'un test :

if-elseif-elseif...

Syntaxe :

```
switch( [variable] ) {
case [valeur1] :
[bloc d'instructions]  break;
case [valeur2] :
[bloc d'instructions]  break;
...
default:
[bloc d'instructions]
                                    }
```

Les structures de contrôle: SWITCH

Le test SWITCH

La valeur de [variable] est comparé successivement à chaque case. Si il y a égalité, le bloc d'instruction est exécuté. Il ne faut pas omettre le break en fin de bloc, sans quoi le reste du switch est exécuté.

Enfin, le handler default permet de définir des instructions à effectuer par défaut, c'est à dire si aucun case n'a ''fonctionné''…

```
<?php
$prenom= " Patricia ";
switch( $prenom )
{  case "Bob" :
   case "Toto" :
   case "Julien" : echo "bonjour ", $prénom , " ! vous  êtes un garçon"; break;
   case "Patricia" : echo "bonjour ", $prénom , " ! vous  êtes une fille";
  default: echo "Bonjour $prénom ! Désolé je ne  connais pas beaucoup de  prénoms"
}
?>
```

Les structures itératives: les boucles

Les boucles servent à répéter l'exécution d'un groupe d'instructions un certain nombre de fois. Cette ensemble d'instructions s'appelle le corps de la boucle. Chaque exécution du corps d'une boucle s'appelle une itération.

On distingue trois sortes de boucles en langages de programmation :

- *Les **boucles while (condition) {….}***
  on y répète des instructions tant qu'une certaine condition est réalisée
- *Les **boucles do {…} while (condition);***
  on y répète des instructions jusqu'à ce qu'une certaine condition soit réalisée
- *Les **boucles for** ou avec compteur : on y répète des instructions en faisant évoluer un compteur (variable particulière) entre une valeur initiale et une valeur finale*

Les structures itératives: les boucles

La boucle While

Cette boucle permet d'effectuer une ou plusieurs actions **tant que** la condition que vous placerez à l'intérieur des parenthèses sera vérifiée.

Syntaxe:

```
<?php
    while(condition)
    {
        instruction 1;
        instruction 2;
        ...
    }
?>
```

Exemple:

```
<?php
    $i = 0;
    while($i < 7)
    {
        echo $i.'<br />';

        $i++;
    }
?>
```

Les structures itératives: les boucles

La boucle DO……WHILE :

La condition de sortie est située en fin de boucle. Ainsi la boucle est parcourue une fois au minimum.

Syntaxe:

```
<?php
    do
    {
        instruction 1;
        instruction 2;
        …
    }
    while(condition)
?>
```

Exemple:

```
<?php
$i = 1;
do
{
  echo $i.'<br />';
  $ i++;
 } while ($i <=5);

?>
```

Les structures itératives: les boucles

La boucle FOR :

Syntaxe:

```
<?php
      for(initialisation; condition de continuité ; modification du compteur)
   {
        instruction 1;
        instruction 2;
        ...
   }
?>
```

Exemple:

```
<?php
        for($i=0;$i < 7;$i++)   echo $i.'<br />';
?>
```

Exercices d’application

Exercice1

Rédiger des expressions conditionnelles pour tester si un nombre est un multiple de 3 ou un multiple de 5.

```
<?php
$x=1245;
if(_____)
{

}
?>
```

Exercice2

Écrire une expression conditionnelle utilisant les variables $age et $sexe dans une instruction if pour sélectionner une personne de sexe féminin dont l'âge est compris entre à 21 et 40 ans et afficher un message de bienvenue approprié.

```
<?php
$sexe="F";
$age=43;
if(_____)
{
.....
}
?>
```

Exercice3

1. Ecrire une table de multiplication simple.
2. Ecrire une table de multiplication et la formater dans un tableau html.
3. Ecrire une table de multiplication, la formater dans un tableau et colorer la cellule en rouge si on a un multiple de 3.
4. Une fois le niveau 3 atteint, créer une jolie mise en page pour cette table.

Niveau 1	Niveau 2	Niveau 3	Niveau 4
1 2 3 4 5 6 7 8 9 2 4 6 8 10 12 14 16 18 3 6 9 12 15 18 21 24 27 4 8 12 16 20 24 28 32 36 5 10 15 20 25 30 35 40 45 6 12 18 24 30 36 42 48 54 7 14 21 28 35 42 49 56 63 8 16 24 32 40 48 56 64 72 9 18 27 36 45 54 63 72 81			

Notions de base

en programmation web

avec PHP

Les tableaux

Les tableaux

Les tableaux, aussi appelés ***arrays*** en anglais, sont des types de données structurés permettant de grouper des informations ensemble. A la différence des types primitifs (entiers, réels, flottants, booléens, chaines de caractères), les tableaux peuvent stocker une ou plusieurs valeurs à la fois (de types différents).

Lors de la déclaration d'un tableau, il est inutile de préciser sa dimension et le type de données qu'il va contenir. PHP s'en charge automatiquement. Les tableaux sont dits ***dynamiques***. A chaque nouvelle entrée enregistrée dans le tableau, PHP agrandit sa taille de 1 élément.

Le langage PHP propose également deux types distincts de tableaux :
les tableaux à index numériques et **les tableaux associatifs**

Les tableaux indicés

Principe:

- Création à l'aide de la fonction array()
- Les éléments d'un tableau peuvent appartenir à des types distincts
- Accéder aux éléments par l'intermédiaire de numéros
- L'index d'un tableau en PHP commence de 0
- Pas de limites supérieures pour les tableaux

Les tableaux indicés

Déclaration d'un tableau

La déclaration d'un tableau vide se fait de la même manière qu'une variable, c'est à dire avec un signe dollars ($) et un nom. Le format du nom doit respecter les mêmes règles de déclaration qu'une variable.

Pour déclarer un nouveau tableau, il suffit d'utiliser la structure de langage **array().** Cette fonction prend en paramètres facultatifs (séparés par une virgule), les valeurs que l'on souhaite insérer dans le tableau pour l'initialiser. Si rien n'est précisé en paramètre, le tableau créé sera vide.

Déclarations :

```
// déclaration d'un tableau vide
$fruits= array();
 //Déclaration et Initialisation d'un tableau indicé
$legumes = array('carotte','poivron','aubergine','chou');
```

Les tableaux indicés

Ajout d'une nouvelle entrée dans un tableau

Pour ajouter une nouvelle valeur dynamiquement à la fin des tableaux précédents, il suffit de procéder comme expliqué dans l'exemple suivant :

```
$fruits[0]= "pomme";
$fruits[1]= "banane";
$fruits[2] = "orange";

$legumes[]= ‘tomate ’;
```

Fonctions relatives :

sizeoff : Renvoie le nombre d'éléments d'un tableau.

count (): C’est un équivalent de sizeof()

```
$nbelements= sizeof( $tableau );
```

Les tableaux associatifs

Un tableau associatif est un tableau dont l'index est une chaîne de caractère au lieu d'un nombre. On parle aussi de ''hash array'' ou ''hash''. Il se déclare comme un tableau traditionnel, la distinction se fait lors de l'affectation

Déclarations :

```
$calories= array();
$identite = array('nom' => 'Hamon', 'prenom' => 'Hugo', 'age' => 19)
```

Affectations :

Affectons un nombre de calories moyen aux fruits.

```
$calories["pommes"]= 300;
$calories["banane"]= 130;
$calories["litchie"]= 30;
// Ajout de la taille de la personne dans le tableau associatif identitie
$identite['taille'] = 180;
```

Les tableaux associatifs

Le tableau associatif est apparu pour pallier les faiblesses du tableau à index numériques. Pour ce dernier, il faut absolument connaître son emplacement pour atteindre la valeur et pour un programmeur ce n'est pas toujours le cas.

Un tableau associatif est un tableau composé de couples ***clé chainée / valeur***. A chaque clé est référencée une valeur.

Pour accéder à l'une des valeurs du tableau, il suffit d'y faire référence de la manière suivante :**$tableau['cle']**. Dans notre exemple précédent, nous pourrions afficher l'identité de la personne de cette façon :

```
<?php
// Affichage des valeurs du tableau associatif
echo 'Nom : ', $identite['nom'] ,'<br/>';
echo 'Prénom : ', $identite['prenom'] ,'<br/>';
echo 'Age : ', $identite['age'] ,' ans<br/>';
echo 'Taille : ', $identite['taille'] ,' cm';
?>
```

Les tableaux associatifs

Fonctions relatives :

isset : pour tester l'existence d'un élément, on utilise la fonction *isset()*

```
if( isset( $calories["pommes"] ) )
{ echo "une pomme contient ", $calories["pommes"] , " calories\n"; }
else { echo "pas de calories définies pour la pomme\n"; }
```

asort, arsort: Ces fonctions de tri conservent la relation entre l'index et la valeur, généralement le cas dans un tableau associatif.

- ✓ *asort* trie par valeurs croissantes,
- ✓ *arsort* par valeurs décroissantes,

Les tableaux

Parcours des éléments d'un tableau

Comme dans tout autre langage de programmation, le parcours de tableau se fait à l'aide de boucles.

- **Avec une boucle for**

```
$tab=array(1,2,3,4,5);
 for ($i=0; $i<count($tab) ; $i++)
   {echo $tab[$i]."<br/>"; }
```

- **Avec une boucle while**

```
$i=0;
while ($tab[$i])
 {echo $tab[$i]."<br/> "; }
```

- **Avec La boucle foreach**

```
$i = 0;
foreach($tab as $elt) { echo "La case n°".$i.":".$elt."<br>";
                        $i++; }
```

Les tableaux

Cette structure prend en paramètre le nom du tableau à parcourir puis les données qu'il faut récupérer (valeurs uniquement ou bien valeurs et clés). Dans la première syntaxe, la valeur de l'élément courant du tableau est directement assignée à la variable $valeur. Dans la seconde, la clé courante de l'élément du tableau est affectée à la variable $cle et sa valeur stockée dans la variable $valeur.

Note : foreach() agit sur une copie du tableau d'origine.

```
<?php
// Affichage des valeurs d'un tableau
foreach($leTableau as $valeur)
{ echo $valeur ,'<br/>';}

// Affichage des couples clé / valeur
foreach($leTableau as $cle => $valeur)
{echo $cle ,' : ', $valeur ,'<br/>';}
?>
```

Exercices d’application

Exercice1

Définissez un tableau associatif correspondant aux facettes d'une personne réelle ou imaginaire (nom, prénom, âge, sexe, adresse). Affichez ensuite les valeurs de ce tableau, mais pas les clés.

Exercice2

Ecrivez un script php qui vérifie si deux tableaux sont effectivement identiques (même nombre de cases, mêmes associations).

Exercice3

Ecrire un script php qui vérifie si un tableau simple est un palindrome (la lecture de gauche à droite est égale à la lecture de droite à gauche).

Exercice4

On voudrait réaliser un tableau HTML donnant le nombre de jours de chaque mois de l'année par un script PHP utilisant un tableau associatif $Mois . Les clefs de ce tableau PHP sont les noms des mois de l'année.
La valeur d'un élément du tableau est le nombre de jours du mois indexant cet élément:

Indices	Janvier	Fevrier	Mars	Avril	Mai	Juin	Juillet	Aout	Septembre	Octobre	Novembre	Decembre
Valeurs	31	28	31	30	31	30	31	31	30	31	30	31

Le tableau HTML possède trois colonnes et se présente comme ceci :

-Afficher dans une première étape les informations brutes de ce tableau (clé, numéro, nombre de jours).

-En deuxième étape, ajouter les balises HTML nécessaires permettant de dresser le tableau en HTML.

Mois	Numéro	Nombre de jours
Janvier	1	31
Février	2	28
Mars	3	31
Avril	4	30
Mai	5	31
Juin	6	30
Juillet	7	31
Aout	8	31
Septembre	9	30
Octobre	10	31
Novembre	11	30
Décembre	12	31