OUTILS INTERNET II

Présenter par Mme Bensalem Hana

Université de Badji-Mokhtar Annaba

Promotion: MasterI ,GADM

Département d'informatique

CHAPITRE III: PHP LES BASES

- Les Pages statiques: Les pages web en HTML(seulement) sont dites statiques, si elles visualisent toujours le même contenu.

- Les Pages dynamiques: les pages HTML dynamiques visualisent l'information à la demande, selon l'interrogation d'une Base de Données par l'intermédiaire d'un Formulaire.

DYNAMIQUE HTML

- Les pages DHTML: dans cette technologie, les pages contiennent du code d'un langage script(applets java…)
- Le script permet de modifier les propriétés des objets HTML:
  - Contrôler leur positionnement
  - Capturer leur évènements,…
- Le code est incorporer dans les balises HTML et interpréter par le navigateur
- Cette solution permet d'insérer du code JSP, CGI, PHP, ASP…

LES LANGAGES SCRIPT

- Plusieurs langages script sur le côté serveur ont été adoptée:
  - Les anciens langages appelés CGI(Common Gateway Interface):consiste à interpréter des programmes écrit en perl ou en langage C.
  - Le langage ASP(Active Server Page),introduit par Microsoft en 1996,qui a permis de simplifie l'écriture des scripts.
  - Le langage PHP(Hypertext Pre Processor)offre de nombreuses fonctionnalités équivalente à celle de la technologie ASP avec une très bonne liaison aux base de données.
  - etc.

ASP

- Basée sur du Visual Basic, ce qui ne demande pas des compétences de haut-niveau.
- Un serveur web IIS(Internet Information Services) qui offre la possibilité d'intégration avec d'autre outils tel que SQL server.
- Simplicité et rapidité des développements,
- Est un mixe entre les balises HTML et le code VBScript.
- …

PHP

- A une syntaxe accessible proche du C
- Langage riches en manipulations des tableaux et listes
- Accès aux BD est plus simple
- Possibilité d'inclure dynamiquement du code
- serveur Apache(HTTP)
- Englobe la notion d'objet qui est absente dans ASP

ASP/PHP

- ASP

- Installer le PWS(Personal Web Server)
- Exemple:

```
<HTML>
   <HEAD>
      <TITLE> VILLE DE ANNABA</TITLE>
      <BODY>Nous somme le <%Date%> </BODY>
</HTML>
```

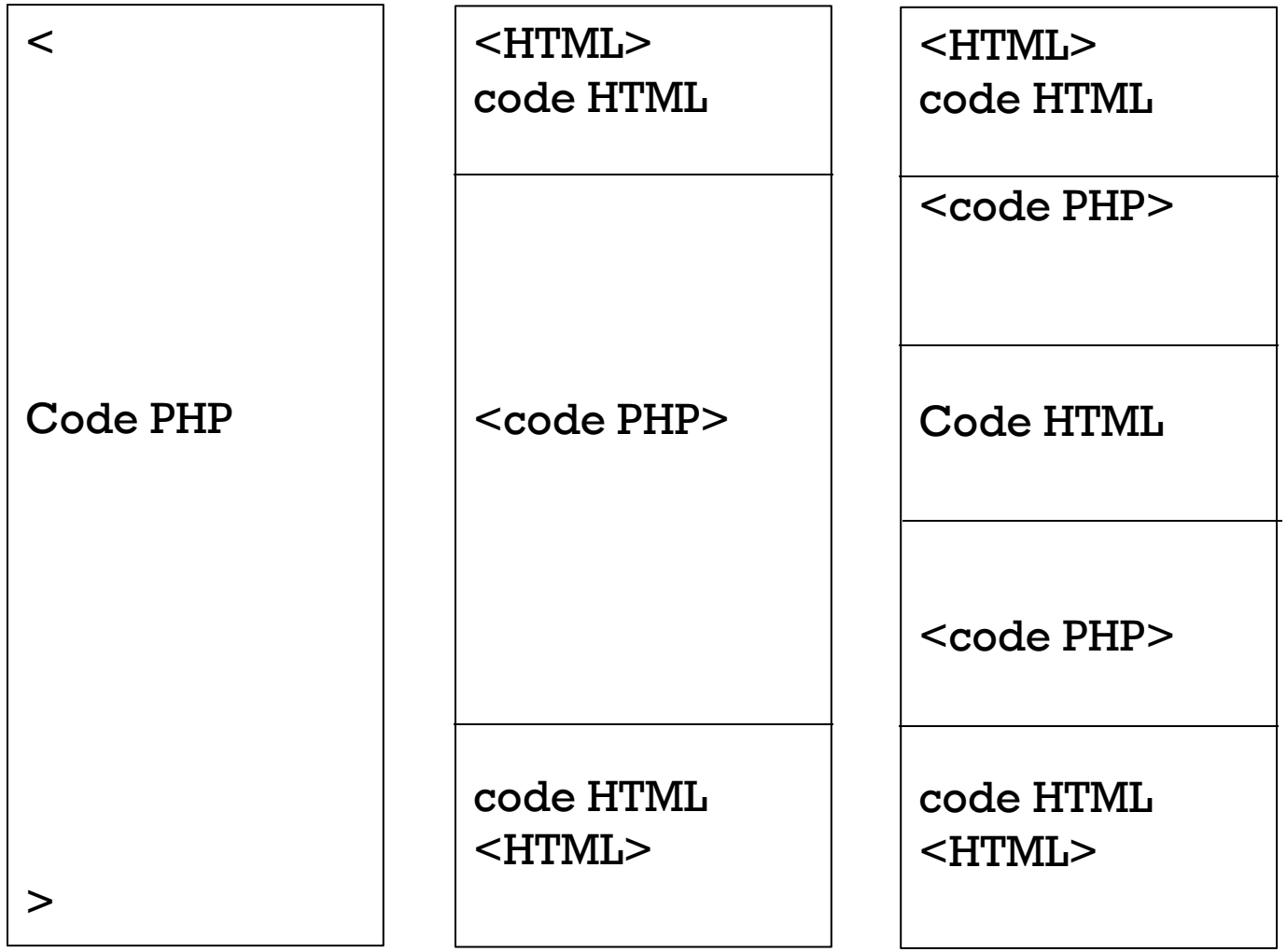
- PHP

- Installer EasyPHP
- Exemple:

```
<HTML>
   <HEAD>
      <TITLE> VILLE DE ANNABA</TITLE>
      <BODY>Nous somme le <?echo date("d/m/y");?> </BODY>
</HTML>
```

PHP

- PHP: est un acronyme désignant initialement « Personal Home Page », est actuellement connu sous le nom de Hypertext Preprocessor.
- Gratuit et open source,
- Intégré dans le serveur apache,
- peut fonctionner avec d'autre serveurs
- très critiqué par les développeurs pour la sécurité
- Mais il est le langage le plus utilisé pour créer des applications web.
- Les versions PHP:version1.0 en 1994,version 2.0 en 1995(intégration de : conditions, boucles, formulaire…),version 3.0 en 1998, version 4.0 en 2000,…
- Actuellement:8.4.4 le 13 Fevrier 2025



Structure d'un script

Exemples de structure d'un script PHP.

```
<HTML>
    <HEAD>
        <TITLE>ca va ? </TITLE>
    </HEAD>

<BODY>

<?php echo " C'est ma première page PHP ";?>

</BODY>

</HTML>
```

PHP LES BASES: LES BALISES

- sont incluses à l'intérieur du code HTML
- sont repérées par les mentions suivantes :
- **< ? … ?>**
- <? php… ?>
- <script language= "**php**" >…</script>
- Si la page porte l'extension ***php*** (plutôt que *html*), le serveur analysera le code avant d'envoyer la page au poste client.

- Fin d'instruction **;**
- Fin du code **?>**
- Un block d'instruction **{…}**
- Les commentaires:
  - Une seule ligne **//** ou **#**
  - Plusieurs lignes **/* …*/**
- Envoi de données vers la sortie standard
  - **print** arg1**;**
  - **echo** arg1**,**arg2**,**…**,**argN**;**

SYNTAXE

Fin d'instruction: **;**

Les commentaires

Séparateurs d'instructions

En cas de chaînes de caractères comme argument=>doit être entourées de "

PREMIER SCRIPT

- <HTML>
  <HEAD>
  <TITLE>ca va ? </TITLE>
  </HEAD>

- <BODY>

```
<?php
Print "exercice1 ";
echo "C'est ma première page PHP  " , " Bonjour! " ;
?>
```

- </BODY>
- </HTML>

LES CONSTANTES

- Définis par la fonction define(" **NOM_CONST** " , val);
- Exemple:
  - define(" **HEURE** –DEJ ", " 12h00 **");**
- Constantes prédéfinies:
  - TRUE
  - FALSE
  - _FILE_ donne le nom du fichier script exécuté
  - _LINE_ fournit le numéro de la ligne du script sur laquelle elle apparait…

Par convention les identifiants des constantes son toujours écrit en majuscule.

- Toute variable commence par $
- L'identifiant d'une variable commence par une lettre ou _ suivi par lettres/chiffres ou _
- Le type de variable est dynamique et repose sur la valeur affectée.
- Le typage est réalisé dés qu'une valeur est affectée à une variable.
- L'opération d'affectation =
- Plusieurs types existent:
  - integer ou int: les valeurs entière positives et négatives
  - double et float: valeurs décimale avec ayant partie décimale et entière séparée par un point.
  - boolean :logique
  - string :chaine de caractères
  - array: les tableaux
  - object:
  - resource:
  - nul:

VARIABLES ET TYPES

- Php est sensible à la casse, sauf pour les noms de fonctions.
- Pour affecter l'adresse d'une variable à une variable: $var1=&$var2
- Var1 peut être considérée comme alias de la variable référencée var2.

- Conversion entre type via (variable) ,exemple: **$nbr = (int)$str;**
- Les opérateurs de conversions:
  - (string),(int),(real)<=>(double) et (float)
  - (array),(object), (bool)

- **Les fonctions tests sur les variables:**
  - **var_dump**(var):retourne le type et la valeur de var.
  - **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()…**

Fonctions sur les variables

Tests et conversion

Les opérateurs

- Les opérateurs arithmétiques:
  - +,- binaire,- unaire ,*,/,% ,++,--.
- Opérateurs relationnels:
  - Ne tenant pas compte des types: ==,!=,<=, >=, <,>.
  - Entre les même types :===,!==.
- Opérateurs de concaténation: c'est le point . ,uniquement pour les chaînes de caractères.
- Opérateurs logiques: ! , || ,&&, xor

- Exemple:

```
<?php
    $nb=3;
    $ch1="$nb portions de frites";
    $ch2='offertes pendant toute la journée' ;   $ch3=<<<promo
   <BR> la veille de ramadhan: <BR>
    <EM> $nb portions de frites</EM>
    <BR>   offertes pendant toute la journée
promo;
echo "$ch1 $ch2<BR>";
echo $ch3;
?>
```

- Accès à un caractère d'après son indice dans la chaine: exemple: le 9iéme élément $ch1{8} donne le caractère 'n' pour notre exemple.
- Les séquences spéciales sont définies pour une chaîne en les précédents par le symbole « \ »,telles que:
  - les guillemets= \",saut de ligne= \n, retour chariot= \r, tabulation horizontale=\t, caractère\=\\.

Les chaînes de caractères

S'écrit entre guillemets ou apostrophes

Syntaxe here-doc:

<<< nom_id …nom_id,si la chaine s'étale sur plusieurs lignes.

Les nom de variables, misent entre apostrophes, ne seront pas substituées par leur contenu.

Fonctions pour les chaines

- Plusieurs fonctions de manipulation de chaînes de caractère:

  - string **strtolower**(string ch),string strtoupper(ch):retourne ch converties respectivement en minuscule et majuscule.
  - string trim(string ch), string rtrim(string ch):retourne ch respectivement sans les espaces de début et fin/sans les espaces de fin.
  - int strcmp(string ch1, string ch2): compare et retourne 0,si les chaines sont égale sinon n positive si ch1 alpha numériquement >ch2,sinon n négative.
  - int strpos(string ch, string sc): retourne la position de la 1iere occurrence de la sous-chaine sc dans ch ou FALSE si sc n'est pas dans ch.
  - int substr_count(string ch, string sc): retourne le nombre d'occurrences de sc dans ch.
  - int strlen(string):retourne le nombre de caractères de ch.
  - string substr_replace(string ch, string rep, int deb, int n): remplace à partir de l'indice deb, la sous chaine de n caractères par rep dans la chaine ch et retourne le résultat.
  - String chr(int ascii): retourne la chaine du code ascii et int ord(string car) fait l'inverse.
  - string strrev(string ch): retourne la chaine inverse de ch.
  - …

```php
<?php
define("x","PHP8");
echo "x vaut :",x." derniére version et est de type ", gettype(x),"<br />";
echo x,"<br>";
$k=strtolower(x);
echo "\$k vaut : $k","<br />";
$a=5 ;
echo "\$a vaut : $a et est de type ", gettype($a),"<br />";
$z=$a;
echo "\$z vaut : $z et est de type ", gettype($z),"<br />";
$y=&$a;echo "\$y vaut : $y et est de type ", gettype($y),"<br />";
$a++;
echo "\$a vaut : $a et est de type ", "<br />";
echo "\$z devient : $z et est de type ", "<br />";
echo "\$y devient : $y et est de type ", "<br />";
 var_dump(isset($w));
?>
```

EXEMPLE 1

x vaut :PHP8 derniére version et est de type string
PHP8
$k vaut : php8
$a vaut : 5 et est de type integer
$z vaut : 5 et est de type integer
$y vaut : 5 et est de type integer
$a vaut : 6 et est de type
$z devient : 5 et est de type
$y devient : 6 et est de type
bool(false)

Les instructions

- if…else

```
if(expression)
    instruction1
else instruction2
```

- l'opérateur conditionnel
  - exp1 ? Exp2 : exp3
  - $a= $b<$c ? $b-$c : 0;
- if…elseif…else

```
if(exp1)
    instruction1
elseif(exp2)
    Instruction2
…
elseif(expN)
    instrutionN
else
    instructionN+1
```

```
if(expression)
    instruction1
```

Instruction switch

```
Switch(exp){
 case val1 : instructions
 case val1 : instructions
  …
 case val1 : instructions
}
```

Les boucles

- while(exp)
  instruction
- do
  instruction
  while(exp);
- for(exp1;exp2;exp3){
  instruction;
  }

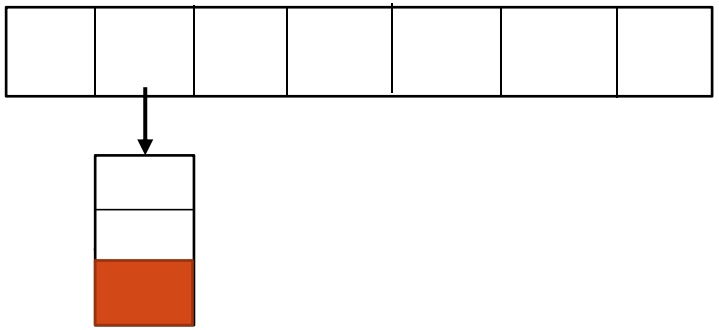
exp1: L'initialisation, exp2: le test de continuation, exp3: évalué à la fin de chaque itération.

- break provoque la sortie des boucles.
- continue provoque le passage à l'itération suivante

Les tableaux 1

- Les éléments d'un tableau peuvent être de types différents.

- **Initialisation**:
  - $tab = array('foobar', 2002, 20.5, $name);
  - ou par case $tab[0]='foobar';
- **Insertion** à la fin:
  - $tab[]='b';
- La fonction **count** retourne le nombre d'éléments non vide d'un tableau.
- Le parcours d'un tableau via
  - les boucles
  - L'instruction **foreach**($tab as $val): accéde dans chaque itération à une case du tableau et met sa valeur, si est non nul, dans $val.
- Tableau à plusieurs dimensions:
  - chaque case peut elle-même être un tableau
  - L'accés exemple : $tab[1] [2]



Un tableau de cardinalité N(fournit par count) peut comporter des cases vides(valeur null) entre les cases non vide.

LES TABLEAUX II

- Les fonctions suivante rendent la valeur d'une case d'un tableau et permettent de se positionner aux:
  - Au début avec: Reset
  - Sur la case suivante avec: next
  - Sur la case précédente avec: prev
  - À la fin avec: end

  → Donnent FALSE S'il n y a pas d'éléments
- key: fournit la clé de la case courante
- current: la valeur de la case courante
- each: retourne un tableau de deux cases la case de la clé et la case de la valeur

LES TABLEAUX ASSOCIATIFS

- Les clés permettent l'accès aux valeurs contenu dans le tableau.
- Si les clés sont des nombres=> tableau avec accès par **indice**
- Si les clés sont des chaînes de caractères=>tableau **associatif.** exemple: $tab['one']
- Initialisation:

Syntaxe-> $nom_tab=array("clé1" =>**val1**, "clé1" =>**val1**,...)

Exemples:

$tab= array(" **one**" =>'foobar', "**two**" =>2002, "**three**" =>20.5, "**viva**" =>$name);

Ou $nom_tab[" **clé** "]=val; $tab[" **one** "]=8;

- Parcours:
  - foreach est la plus simple pour parcourir ces tableaux

```
foreach($personne as $key => $elem)
 {  echo ''$key : $elem"; }
```

Code

```php
<?php
$name='outils internet';
$tab= array(" one"  =>'foobar', "two"  =>2002, "three"  =>20.5, "viva"  =>$name);
foreach($tab as $key => $elem)
 {  echo "$key : $elem <br>"; }
?>
```

interpret

Code interpreted by the server and sent to the browser

one : foobar
two : 2002
three : 20.5
viva : outils internet

FONCTIONS POUR LES TABLEAUX

- Des fonctions
  - de **trie**:
    - void sort(arry t),void rsort(array t)…
    - void asort(array t), void arsort(array t), void ksort(array t)…
  - **Ajouts et suppressions** d'éléments:
    - int array_push(array t,liste arg): empile à la fin de t les élément de liste et retourne le nombre d'éléments de t.
    - type_element array_pop(array t): dépile et retourne le dernier élément de t
    - type_element array_shift(array t,liste arg):supprime et retourne le premier élément de t.
    - …
  - D'**informations** sur le tableau:
    - boolean in_array(type_element rech,array t) retourne TRUE si rech est dans t
    - type_element min(array t): retourne le minimum de t.
    - array array_keys(array t)/ array array_values(array t) : retourne respectivement les clés et les valeurs de t.
    - …
  - **Modification de l'ordre** du contenu(inverser, réordonner aléatoirement…)
    - array array_inverse(array t): retourne le tableau t dans l'ordre inverse
    - void shuffle(array t):mélange aléatoirement les cases de t
    - Array explode(string sep,string ch):transforme une chaîne ch en un tableau selon le separateur sep. Tandis que **string implode(string sep,array t)** fait l'inverse
  - …

croissant:/décroissant

Les tableaux associatifs

```
<?php
for($i=11;$i<=36;$i++)
{$tab[$i]=chr(54+$i);}
//Lecture avec for
for($i=11;$i<=36;$i++)
{echo "Elément d'indice $i : ",$tab[$i]," <br />";}
echo "<hr />";
//Lecture avec foreach
rsort($tab);
foreach($tab as $cle=>$valeur)
{echo "Elément d'indice $cle : $valeur <br />";}
echo "<hr />";
?>
```

EXEMPE2

- Créer un tableau dont les indices varient de 11 à 36 et dont les valeurs sont des lettres de A à Z.
- Lire ensuite ce tableau avec une boucle for puis une boucle foreach
- afficher les indices et les valeurs.

Code

```php
<?php

for($i=11;$i<=36;$i++)
{$tab[$i]=chr(54+$i);}
//Lecture avec for
for($i=11;$i<=36;$i++)
{echo "Elément d'indice $i : ",$tab[$i]," <br />";}
echo "<hr />";
//Lecture avec foreach
rsort($tab);
foreach($tab as $cle=>$valeur)
{echo "Elément d'indice $cle : $valeur <br />";}
echo "<hr />";
?>
```

interpret

Code interpreted by the server and sent to the browser

Elément d'indice 11 : A
Elément d'indice 12 : B
Elément d'indice 13 : C
Elément d'indice 14 : D
Elément d'indice 15 : E
Elément d'indice 16 : F
Elément d'indice 17 : G
Elément d'indice 18 : H

Fonctions Prédéfinis

Date et heures

- timestamp: est le nombre de secondes écoulées depuis 1ier janvier 1970.
- time() retourne le timestamp courant.
- getdate():retourne un tableau associatif contenant

  hours, minutes, seconds, year, mon(de 1à12), month,…
- date():prend en argument un format de date et retourne une chaine de caractères représentant le timestamp selon le format précisé en utilisant une liste de caractères format:
  - H heure de 00à 23,
  - a: ajoute am ou pm ,
  - i: minutes de 00à59,
  - s:secondes de 00à59,
  - Y:années 4chiffres,
  - y:année 2chiffres,
  - m:mois de01à12,
  - d:jour de 01à31…

un caractère ne soit pas interprété comme un caractère de format il faut le préfixer de \

Exemple:

```
<?php
$t=time();
echo date("H\h i\m  s\s" ,$t),"<BR>" ;
echo date("hA i  s" ,$t) ,"<BR>";
echo date(" d m  Y" ,$t),"<BR>";
?>
```

Fonctions prédéfinis

mathématiques

- sin(), cos, tan(), asin()…
- ceil(k): si k=n.r =>ceil(k)=n+1
- floor(k)=retourne la valeur de la partie entière de k.
- round(): arrondis le réel passé comme paramètre et peut prendre un deuxième paramètre comme précision(nombre de décimal à prendre en compte).
- min(),max(): retournent respectivement min et max d'une liste d'arguments.
- string decbin (int n) : convertis du décimal au binaire et le résultat est mis sous forme de chaine de caractères(de même pour decoct(), dechex(), octdec()) et l'inverse de :bindec(),octdec(),hexdec().

- String base_convert(string n,in b1,in b2): convertis n depuis la base b1 vers la base b2 et le résultat est mis sous forme de chaine de caractères.
- rand ()/mt_rand(): retournent un nombre pseudo aléatoire compris entre deux bornes données.
- lcg_value: retourne un réel entre 0et1.
- exp(),log(),log10():retournent respectivement l'exponentielle, logarithme népérien et logarithme à base de 10 d'un nombre réel donnée.
- float sqrt(float x):racine carrée de x
- float pow(float x,float y): x puissance y
- …

```
<?php
$t=base_convert("17",16,10);
echo var_dump($t);
$t=log10($t)+exp($t);
echo $t,"<br>";
?>
```

Les fonctions

- Déclaration :

```
function nom_fonction(param1,param2,…,paramN)
{ …
instruction
…
 return exp;
}
```

- Appel :

```
nom_fonction(liste_Param);
```

- Paramétre passé par adresse
  - function nom_fonction(param1,&param2,…,paramN)
- Valeur par défaut
  - function nom_fonction(param1,param2,…,paramN =val)
  - Doivent être à la fin de la liste des paramétres
  - Peuvent être omis lors de l'appel.
- variables globale: les variables globales sont statiques et existent pendant tout le script sauf dans les fonctions.
- Il faut précéder cette variable par **global** pour qu'elle soit prise en compte en tant que globale dans une fonction.
- Pour mémoriser la variable locale d'un appel à un autre il faut préciser qu'elle est statique en la précèdent du mot **static**.

fun_num_args() :nombre d'arguments fourni lors de l'appel

func_get_arg(n) : fourni l' l'argument numéro n de l'appel

(numérotation à partir de 0)

func_get_args() : retourne un tableau de tout les arguments

EXEMPLE3

```
<?php
   function decrement($n)
{      if ($n!=0){ echo $n,"<br>";
                     decrement($n-1);
                  }
 // Écrivez votre code ici
}
echo decrement(10);
?>
```

EXERCICE1

Écrivez un programme pour afficher le triangle d'etoile suivant en utilisant une boucle for.

Exemple:

```
*
**
***
****
*****
******
*******
********
*********
**********
```

SOLUTION1

```
<?php
for($ligne = 1; $ligne <= 10; $ligne++)
{
for ($etoile = 1; $etoile <= $ligne; $etoile++)
{
echo "*";
}
echo "<br>";
}
?>
```

- Exercice:

Ecrire une fonction qui calcule le factoriel d'un nombre

EXERCICE 2

```php
<?php
function fact($n){
                    if($n <= 1){return 1;}
                    else{return $n * fact($n - 1);}
                }
$n = 3;
$f = fact($n);
echo "La factorielle de $n est $f";
?>
```

SOLUTION2

EXERCICE3

Écrire deux fonctions qui :

1. Transfome une chaine de caractétre en chaine en majuscule.
2. Puis applique cette fonction sur un tableau
3. Ecrire une deuxiéme fonction qui affiche es éléments d'un tableau
4. Afficher le tableau de chaine transformé en majuscule.

```
<?php
 function maj(&$case)
            {$case=strtoupper($case);
            }
$case='bonjour’;
maj($case);
echo$case,"<br>";
$produits=array('poulet','pizza','lasagne’);
array_walk($produits,'maj’);
affich($produits);
 function affich($tab)
             {$k=count($tab);
            for($i=0;$i<$k;$i++)
            {echo $tab[$i],"<br>";}
             }
?>
```

EXPRESSION RATIONNELLES (REGULAR EXPRESSION)

- est une chaine de caractére pour un modèle de chaines de caractères.
- Utile pour effectuer des recherches et traitements dans les chaines de caractères.
- Regex POSIX & PCRE(Perl Compatible Regular Expression): des extensions de php pour utiliser ces expressions.
- depuis la version 5.3,PCRE est la seule présente et ne peut être désactiver.

-a lamême syntaxe que le language PERL
- est plus souple,puissante et plus rapide.

STRUTURE

- les délimiteurs:
  - souvent on utilise :/ ,( ) ,{ } , [ ],< >,#...
  - si l'un de ces caractéres fait partie de la chaine d'origine il doient être précédé par l'antislash \
  - exemple: /http:\/ \/ /
  - Pour exprimer l'antislash même dans un modéle de chaîne il faut le précéder par un autre antislash: \\
- les carctéres spéciaux: l'antislash suivie d'une lettre permet de les exprimer

CARACTÈRES SPÉCIAUX

symbole	signification
\w	Alphabet ou chiffre
\W	caractère ≠Alphabet ou chiffre
\d	Chiffre
\D	Caractère ≠chiffres
\s	Caractères d'espacement
\S	Tout sauf un caractère d'espacement
\b	Une correspondance au début de la chaine pour \b**ch** et à la fin pour **ch**\b
\n	nouvelle ligne
\r	retour chariot
\f	saut de page
\t	tabulation

Utilisés pour donner le modèle à rechercher

INDICATEUR D'OCCURRENCE

Symbole	signification	expression	exemple
*	Zero occurrence ou plus	a*	« », «a», «aa», «aaa»,..
+	Au moins une occurrence	a+	«a», «aa», «aaa»,..
?	Zero ou une occurrence	a?	« », «a»
{n}	Exactement **n** occurrences	a{3}	«aaa»
{n,}	Au moins **n** occurrences	a{2,}	«aa», «aaa»,..
{n,m}	Au moins **n** occurrences et au plus **m**	a{1,3}	«a», «aa», «aaa»

CLASSES DE CARACTÈRES

- Les crochets permettent de définir les classes de symboles(liste de symboles).
- Les classes prédéfinies s'écrivent entre « [: » et « :] »
- Les intervalles de caractères sont exprimé par –

symbole	signification	expression	exemple
[:alpha:]	[AZ-az]	[[:alpha:]]	Bonjou
[:lower:]	[a-z]	[[:lower:]]+	onjou
[:upper:]	[A-Z]	[[:upper:]]+	B
[:digit:]	[0-9]	[[:digit:]]+	5
[:alnum:]	[AZ-az-0-9]	[[:alnum:]]+	Bonjou3
[:xdigit:]	[0-9a-fA-F]	[[:xdigit:]]	D
[:punct:]	ponctuation	[[:punct:]]	.
[:blank:]	Espace	([[:AZ-az:]]+[[:blank:]])+	Il est
[:space:]	Caractère d'espacement		

CLASSE DE CARACTÈRES

Symbole	signification	expression	exemple
.	remplace un caractère quelconque	.u	Ou
		i.*	il est tard
[liste_car]	Tout caractère de la liste	[ruBjno]+	Bonjour
[^liste_car]	Tout sauf un caractère de la liste	h[^s]+	heur
-	intervalle	[A-Za-z]*	Bonjour
		[0-9]	9

LA CORRESPONDANCE

- Est cherchée
  - par défaut dans toute la chaine
  - peut intervenir dans le début, fin ou début et fin de la chaine.
  - Exemple la correspondance entre les expressions et la chaîne 'Bonsoir, il est 5 heures'

symbole	signification	Exemple d'expression	Résultat de correspondance
^	Correspondance en début	^i.*	Aucune
		^B.{6}	Bonsoir
$	Correspondance en fin	.{5}s$	heures
		B.{6}$	Aucune
[:<:]	Le mot commence par	[[:<:]]e.+	est 5 heures
[:>:]	Le mot se termine par	.*t[[:>:]]	Bonsoir, il est

ENCORE DES OPTIONS

- Une sous-occurrence d'une expression est une partie de l'expression comprise entre parenthèse, exemple: $exp='([0-9]{2}):([0-9]{2})' ,$exp comporte deux sous-occurrences.

symbole	signification
\b	Une correspondance au début de la chaine pour \b**ch** et à la fin pour **ch**\b

Option de recherche:sont placés apres le délimiteur de fermeture pour preciser le comportement de la recherche.

symbole	signification
i	recherche insensible à la case
m	recherche multilignes
x	permet d'ignorer les caractéres d'espacement

FONCTIONS PRÉDÉFINIES

Ajout de i après délimiteur de chaine pour une correspondance non sensible à la casse

Function	Description
Boolean preg_match(string exp, string ch, array t)	Retourne 1 s'il y a une correspondence de l'expression dans la chaine,0 si non.la correspondance est stockée dans la case 0 de t.
preg_match_all(string exp, string ch, array t)	Retourne le nombre de correspondence de l'expression dans la chaine, si non 0.
preg_replace(string exp, string ch, array t)	Returns a new string where matched patterns have been replaced with another string

```
$str = "Visit W3Schools";
$pattern = "/w3schools/i";
echo preg_match($pattern, $str);
```

1

EXEMPLES

```
<?php
$str = "Visit W3Schools";
$pattern = "/w3schools/i";
if (preg_match($pattern, $str,$tab)) echo $tab[0];
?>
```

W3Schools

```
<?php
$txt = "Hello World";
$pattern = "/\bHel/";
echo preg_match_all($pattern, $txt);
?>
```

1

EXERCICE1

modele="/^[a-z]([a-z0-9])+(\.|-)?([a-z0-9]+)@([a-z0-9]{2,})\.([a-z]{2,4})$/«

- **On** cherche à valider une ad… …utilisateur.
- Les extensions des d… …quatre caractères.
- Les adresses valid… :
  - nom@domai…
  - prenom.nom@domaine.com
  - prenom-nom@domaine.info

- **Commence** par une lettre
- Suivi par **au moins** une lettre et/ou chiffre
  - **Optionnellement** suivi par un point ou -
- Puis par au **moins** une lettre et/ou chiffre
- Suivi obligatoirement par @
- Puis au moins deux caractères alphabet et/ou chiffre
- **Se termine** par 2 à 4 lettres (l'extension)

```php
<?php
//Création de la fonction de validation
function validmail($ch)
{$modele="/^[a-z]([a-z0-9])+(\.|-)?([a-z0-9]+)@([a-z0-9]{2,})\.([a-z]{2,4})$/";
$ch=strtolower($ch);
if (preg_match($modele,$ch)){echo "$ch est valide <br />"; return TRUE;}
else{echo "$ch est invalide <br />";return FALSE;}
}
//Utilisation de la fonction de validation
$mail="Jean5.dupont@laposte2.uk";
$mail2="5pierre.dupapi@plusloin.info";
$mail3="engels-jean@funphp.com";
validmail($mail);
validmail($mail2);
validmail($mail3);
?>
```

jean5.dupont@laposte2.uk est valide
5pierre.dupapi@plusloin.info est invalide
engels-jean@funphp.com est valide