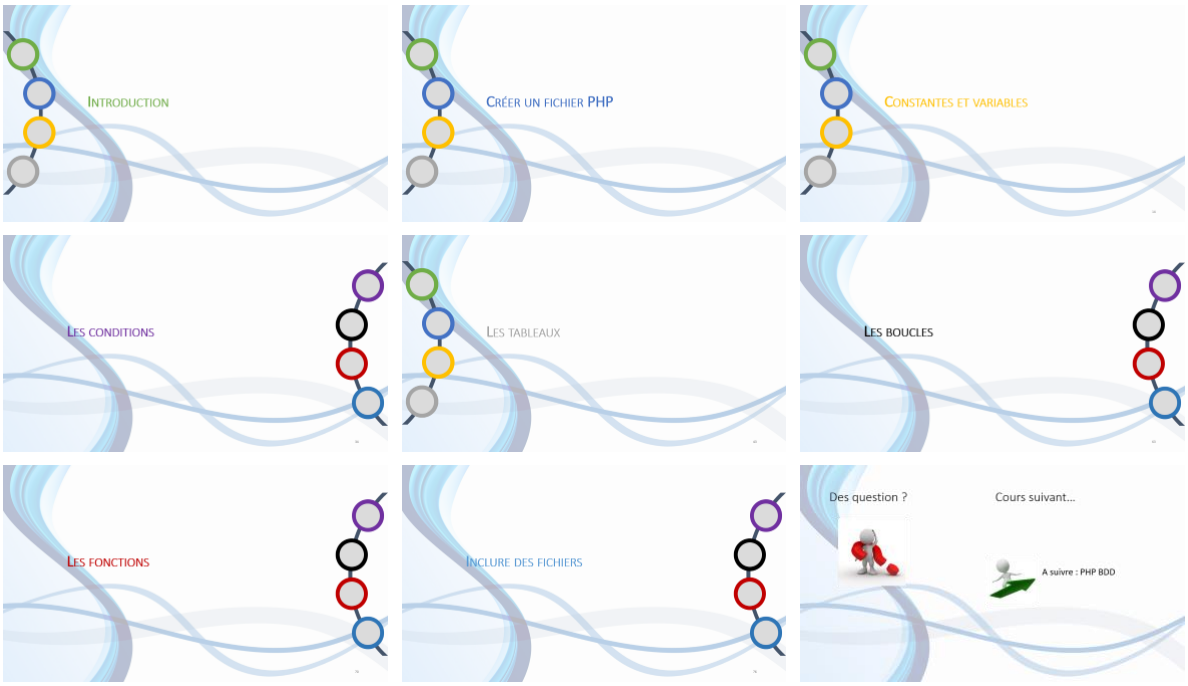
Développement Web Avancé

Bases PHP

Christel Ehrhart

1



2

3

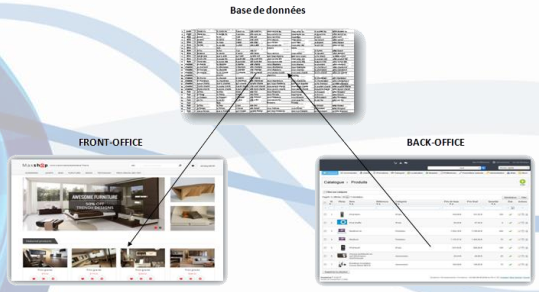
Introduction

- 1994
  - PHP = "Personal Home Page"
  - Développeur Canadodanois Rasmus Lerdorf pour stocker la trace des visiteurs consultant son CV en ligne
- PHP = "Hypertext Preprocessor"
- Utilisé par près de 83% des sites web
- Gratuit, OpenSource et distribué sous une license autorisant la modification et la redistribution
- Version et support : http://php.net/supported-versions.php

4

Sites dynamiques

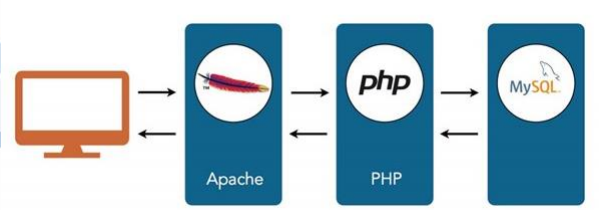
- Site dont les informations proviennent d'une base de données
- A ne pas confondre avec un site de mouvement ou d'animations
- Possède 2 interfaces :
  - un premier site FRONT (pour l'affichage du site normal)
  - un deuxième site BACK (pour la gestion et assurer les réglages du premier site)



5

Exécution

- D'un point de vue exécution, PHP a besoin d'un serveur Web (Apache ou Nginx) pour fonctionner
- Toutes les pages demandées par un client seront construites par le serveur Web, en fonction des paramètres transmis, avant d'être retournées au client



6

6

Fonctionnement

- Étape 1 : L'utilisateur tape l'adresse du site web qu'il souhaite afficher
- Étape 2 : La demande est formalisée par le navigateur
- Étape 3 : La requête est reçue par le serveur web configuré
- Étape 4 : Le serveur web exécute le code du site web demandé
- Étape 5 : Le code du site web génère la page web demandée avec appel à une base de données si besoin
- Étape 6 : La page générée est renvoyée au navigateur par le serveur web
- Étape 7 : Le résultat de la page est reçu par le navigateur

7

7

A retenir

- PHP est un langage exécuté côté serveur, ce ne sera donc pas une exécution par l'ordinateur ou le navigateur de l'internaute.
- PHP est un langage interprété, ses instructions sont traitées séquentiellement par le serveur (pas de compilation)
- PHP est open source, tout le monde peut l'utiliser et même vendre une de ces créations à l'aide du langage PHP.

8

Préparer son environnement de travail

- Choisir son environnement de travail (local/distant) et l'installer
- Choisir son IDE :
  - Notepad++ (https://notepad-plus-plus.org/fr/)
  - Sublime Text (https://www.sublimetext.com)
  - Netbeans (https://netbeans.org/features/index.html)
  - Eclipse (https://eclipse.org/)
  - …
- Utiliser une plateforme de versioning (Git, Bitbucket, …)
- Choisir son navigateur (Chrome, Firefox, …)

9

9

CRÉER UN FICHIER PHP

10

Fichier PHP

- Possède l'extension php
- Le fichier est un fichier modifiable avec un simple éditeur de texte
- Structure du fichier php :

```
<?php
        instruction;
        // Commentaire sur une ligne
        /* Commentaire sur
                plusieurs lignes */
```

- Un fichier composé uniquement de php ne contient pas d'instruction de fin
- Dans un fichier composé de HTML et de php, le bloc php se termine par ?>

11

Instructions d'affichage

- Pour afficher des informations sur la page, il faut faire appel à des instructions d'affichage :
  - echo ou print : permet d'afficher un texte ou le contenu d'une variable. N'affiche pas les tableaux.
  - var_dump : permet d'afficher tous les éléments d'un texte ou d'une variable
  - print_r : permet d'afficher un texte ou le contenu d'une variable

```
<?php
        echo("J'affiche un texte");
        print_r("J'affiche un texte ou un tableau");
        print_r($arrTab);
        var_dump("J'affiche en plus de mes collègues, des informations complémentaires");
```

12

Mixer PHP et HTML

```
<!Doctype html>
<html lang='fr'>
        <head>
                <meta charset="utf-8">
                <title>Ma Page</title>
        </head>
        <body>
                <h1>Mon premier Titre
!</h1>
                <p>Et mon premier
paragraphe de texte qui l'accompagne
!!</p>
        </body>
</html>
```

```
<!Doctype html>
<html lang='fr'>
        <head>
                <meta charset="utf-8">
                <title><?php echo("Ma page"); ?></title>
        </head>
        <body>
                <h1><?php echo("Mon premier Titre !"); ?></h1>
                <p><?php echo("Et mon premier paragraphe de
texte qui l'accompagne !!"); ?></p>
        </body>
</html>
```

Attention : il faut éviter au maximum les entrées sorties répétitives, privilégiez :

```
<?php     echo("<h1> Mon premier Titre ! </h1>");
          echo("<p>Et mon premier paragraphe de texte qui l'accompagne !! </p>");
?>
```

13

CONSTANTES ET VARIABLES

14

14

15

Les Constantes

- La valeur ne pourra être changée lors de l'exécution d'un programme
- La fonction define() permet de définir une constante avec en paramètres le nom de la variable et sa valeur :

define("*NOM_DE_LA_CONSTANTE*", *Valeur*);

```
<?php
        define('TVA', 20);
        echo TVA;
    ?>
```

- Les constantes ne sont pas précédées du signe $
- Par convention, on écrit les constantes en majuscules

- Remarque : on peut aussi utiliser const *NON_DE_LA_CONSTANTE = Valeur;* particulièrement dans les classes (POO)

16

16

Les constantes magiques

- Il existe des constantes prédéfinies par PHP, on les appelle les constantes magiques

```
<?php
    echo __FILE__ . "<br>"; // Affiche le chemin complet vers le fichier actuel
    echo __LINE__ . "<br>"; // Affiche le numéro de la ligne
```

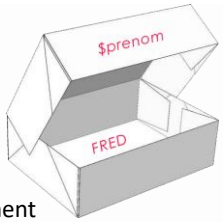
⇒https://secure.php.net/manual/fr/language.constants.predefined.php

17

LES VARIABLES

18

Les variables



- Une variable est un objet ou une information stockée en mémoire temporairement
- En PHP, la variable existe tant que la page est en cours de génération
- Après l'exécution du programme, toutes les variables sont supprimées de la mémoire car elles ne servent plus à rien
- Une variable est caractérisée par son nom
- Le nom de variable est sensible à la casse (majuscule / minuscule)
- Un nom de variable valide doit commencer par une lettre ou un souligné (_)
- La déclaration d'une variable se fait par l'écriture du symbole '$' devant la variable à manipuler

19

19

Règles de nommage

- Construire les noms de variables avec leur type :
  $strCustName, $intNb, $floatAmount, $arrList
- Nom de variables en anglais
- Commencer par une minuscule, puis chaque mot commence par une majuscule

20

20

Affectation d'une valeur à une variable en PHP

- A une variable, on assigne une valeur
- L'instruction d'affectation permet d'affecter une valeur à une variable
- Il suffit de mettre un signe égal après la déclaration de la variable et de mettre la valeur associée

```
$floatPrice = 10; //La variable prix a comme valeur 10
```

21

21

Les Types de variables en PHP

- Selon la valeur qu'elle va contenir, une variable a un type
- Contrairement à de nombreux autres langages, il n'est pas obligatoire en PHP de préciser les types
- Toutefois, il existe plusieurs types de variables en PHP

Type de variable	description	exemple
les booléens	deux valeurs constantes possibles : TRUE ou FALSE	$boolValid = true;
les réels	Ce sont les nombres décimaux connus aussi sous le vocable de " double ", " float " ou " nombre réels ". Un prix peut par exemple être un réel.	$floatPrice = 2.542;
les entiers	un nombre appartenant à l'ensemble des entiers Z: Z = {...,-1,0, 1, 2, ...}	$intAge = 10;
les chaînes de caractères	Ce sont les variables de type texte, ou string Il faut les mettre entre guillemet Les chaînes de caractères sont des séquences de caractères.	$strName = "Dupont";
les tableaux	Liste de plusieurs éléments	$arrStudent = array("Larah", "Sebastian", "Kevin", "Audile", "Rodrigue");

22

22

23

Concaténation lors de l'affichage

- La concaténation permet de mettre bout à bout deux chaînes
- La concaténation en PHP se fait avec le .

- Exemple de concaténations :

```
<?php
        $strTxt         = "Mon nom est";
        $strName        = "Dupont";

        // concaténation de textes et de variables
        echo "<p>".$strTxt." ".$strName."</p>";
        // sans concaténation, uniquement valable avec les "
        echo "<p>$strTxt $strName</p>";
        // la , au lieu du . fonctionne pour echo
        echo "<p>", $strTxt, " ", $strName, "</p>";
?>
```

24

24

Concaténation lors de l'affectation

- On peut concaténer directement au moment de l'affectation en utilisant l'opérateur .=

```
<?php
        // Affectation de la valeur "Bruno" sur la variable : $strPrenom1
        $strPrenom1 = "Bruno";
        // Affectation de la valeur "Claire" sur la variable : $strPrenom1.
        $strPrenom1 = "Claire";
        // Affiche : Claire
        echo $strPrenom1 . "<br>";

        // Affectation de la valeur "Nicolas" sur la variable : $strPrenom2
        $ strPrenom2 = "Nicolas";
        // Affectation de la valeur " Marie" sur la variable : $ strPrenom2.
        // Ajout SANS remplacement de la valeur précédente
        $ strPrenom2 .= "Marie";
        // Affiche : NicolasMarie
        echo $ strPrenom2. "<br>";
```

25

Travaux Pratiques

Exercices sur les variables

26

27

Longueur d'une chaîne et élagage

- trim() : fonction permettant d'enlever les espaces au début et à la fin de la chaîne
- strlen() : fonction permettant de connaitre la longueur du texte
- trim() peut retirer un autre caractère que l'espace en précisant le caractère en deuxième paramètre

```
<?php

        $strLib = " on étudie le php et etc.... ";

        echo strlen($strLib)."<br />"; //renvoie 30

        $strLib = trim($strLib);
        echo $strLib."<br />";
        echo strlen($strLib)."<br />"; //renvoie 28

        $strLib = trim($strLib,".");
        echo $strLib."<br />";
        echo strlen($strLib)."<br />"; //renvoie 24

?>
```

28

28

Recherche d'une sous-chaîne en PHP

- strpos() retourne un nombre qui indique la position de la première occurrence du caractère recherché
- strstr() retourne le reste de la chaîne à partir de la chaîne repérée
- Elles ont 2 deux arguments : la chaîne dans laquelle on effectue la recherche et le caractère à rechercher

```
<?php
        $strAddress = "nom@domaine.fr";
        $strArobase = "@";
        echo strpos($strAddress, $strArobase) . "<br />"; //retourne la position :3

        //retourne la chaîne après avec la chaîne recherchée: @domaine.fr
        echo strstr($strAddress, $strArobase) . "<br />";
?>
```

- stripos() et stristr() pour l'analyse sans prendre en compte la casse
- strrpos() trouve la position de la dernière occurrence d'un caractère dans une chaîne

29

29

Récupération d'une sous-chaîne

- substr() permet de récupérer une partie d'une chaîne de caractères
- syntaxe : substr(chaîne, nombre_de_départ, nombre_de_caractères)

```
<?php
        $strTxt = "Dans ce texte, essayons d'afficher juste les 10 premiers caractères. On coupe donc le texte.";
        echo substr($strTxt, 0, 10); // affiche " Dans ce te "
        echo("<br />");
        echo substr($strTxt, -1); // retourne le dernier caractère
?>
```

30

30

Remplacer un motif dans une chaîne

- str_replace() permet de remplacer un texte dans une chaîne de caractères
- syntaxe : str_replace(*valeur_cherche*, *valeur_de_remplacement*, *texte*)

```
<?php
        $strTxt ="Cher personne, <br />Nous vous remercions d'avoir bien voulus nous rejoindre. Merci personne de votre participation.C'est très aimable de votre part.<br />A bientôt, personne";
        echo str_replace( "personne", "Marcel", $strTxt);
?>
```

31

31

Changement de casse

- strtolower() permet de mettre une chaîne de caractères tout en minuscules
- strtoupper() permet de mettre une chaîne de caractères tout en majuscules

```
<?php
        $strName    = "Vernes";
        $strMail    = "JULES@domaine.fr";
        //Mettre en majuscule
        echo $strName." devient ".strtoupper($strName)."<br />";
        //Mettre en minuscule
        echo $strMail." devient ".strtolower($strMail)."<br />";
?>
```

32

32

Premier caractère en majuscule

- ucfirst() convertit en majuscules le premier caractère de la chaîne
- lcfirst() convertit en majuscules le premier caractère de chaque mot

```
<?php
        $strLib = "une phrase débute toujours par une majuscule.";
        echo $strLib."<br />";
        echo ucfirst($strLib)."<br />";

        $strAcronym = "fond monétaire international";
        echo $strAcronym."<br />";
        echo ucwords($strAcronym);
?>
```

33

33

LES CONDITIONS

34

34

Les opérateurs de comparaison

- Egal :
  - $a == $b retourne true si la valeur de $a est égale à celle de $b
- Différent :
  - $a != $b retourne true si la valeur de $a est différente de celle de $b
  - $a <> $b retourne true aussi si la valeur de $a est différente de celle de $b
- Inférieure :
  - $a < $b retourne true aussi si la valeur de $a est strictement inférieure à celle de $b
- Inférieure ou égal :
  - $a < = $b retourne true aussi si la valeur de $a est inférieure ou égale à celle de $b
- Supérieure :
  - $a > $b retourne true aussi si la valeur de $a est strictement supérieure à celle de $b
- Supérieure ou égal :
  - $a >= $b retourne true aussi si la valeur de $a est supérieure ou égale à celle de $b

35

35

La structure conditionnelle

- La syntaxe est :

```
<?php
        if(condition){
                // Bloc d'instructions
        }
?>
```

- On met la ou les conditions entre parenthèses, et le bloc d'instructions entre deux accolades

- Exemple :

```
<?php
        $intSpeed = 100;

        if($intSpeed > 50){
                echo "excès de vitesse !";
        }
?>
```

36

36

La clause else

- Clause optionnelle du if
- La syntaxe est :

```
if (condition) {
        // instruction au cas où la condition serait réalisée;
}else{
        // instruction au cas où la condition ne serait pas réalisée;
}
```

- Exemple :

```
<?php
        $intAge = 22;
        if($intAge >= 18){
                echo "Vous êtes majeur.";
        }else{
                echo "Vous êtes mineur.";
        }
?>
```

37

37

Imbrication if

- On peut imbriquer les if les uns dans les autres
- Lorsqu'un if imbriqué aura fini d'être exécuté, il retournera à l'étape logique suivante du rang hiérarchique supérieur
- Exemple :

```
<?php
        $intNbr = -4;
        if($intNbr == 0){
                echo "le nombre est égal à zéro";
        }else{
                if($intNbr > 0) {
                        echo "le nombre est positif";
                } else {
                        echo "le nombre est négatif";
                }
        }
?>
```

38

38

Clause elseif

- Le nombre d'elseif est illimité
- Le else à la fin est obligatoire et il est exécuté lorsqu'aucune des conditions en dessus ne sont pas exécutées
- Exemple :

```
<?php
        $intNbr = -4;
        if($intNbr == 0){
                echo "le nombre est égal à zéro";
        }elseif($intNbr > 0){
                echo "le nombre est positif";
        }else{
                echo "le nombre est négatif";
        }
?>
```

39

39

Syntaxe de l'instruction conditionnelle Switch

- switch() est une alternative à la structure if() / elseif() / else ou bien aux imbrications de blocs if()
- Sa syntaxe repose sur l'utilisation de 3 mots clés : switch, case et default.
- Cette instruction conditionnelle permet de tester toutes les valeurs possibles que peut prendre une variable.

40

40

Exemple d'utilisation de l'instruction switch

```
<?php
        $strVeget = "rien";
        switch($strVeget){
                case 'salade':      echo 'Vous avez acheté de la salade !';
                                    break;
                case 'Carotte':     echo 'Vous avez acheté de la Carotte !';
                                    break;
                case 'poivrons':    echo 'Vous avez acheté des poivrons!';
                                    break;
                case 'aubergines':          echo'Vous avez acheté des aubergines!';
                                    break;
                default :           echo 'Vous avez acheté un autre légume' ;
                                    break;
        }
?>
```

41

41

Travaux Pratiques

Exercices sur les conditions

42

43

Les tableaux à index numériques

- Déclarer le tableau

```
<?php
    $arrStudent = array("Larah", "Sebastian", "Kevin", "Audile", "Rodrigue");
?>
```

- Afficher le tableau

```
<?php
        echo $arrStudent[0]; // Affiche Larah

        // Affiche tous les éléments du tableau
        for($i = 0;$i <= count($arrStudent); $i++){
            echo $arrStudent[$i] . "<br />";
        }
?>
```

44

44

Les tableaux associatifs – déclarer le tableau

- Déclarer le tableau :

```
$arrList= array( cle1=>valeur1, cle2=>valeur2, ... );
```

- Exemple:

```
<?php
        $arrPerson =    array(
                        "firstname"  => "Jessy",
                        "name"       => "Brown",
                        "tel"        => "000011111"
                    );
?>
```

45

45

Les tableaux associatifs – afficher le tableau

```
<?php
        echo $arrPerson['name']; //affichage de l'élément nom

        // Affichage de tous les éléments du tableau
        foreach ($arrPerson as $key => $value) {
                echo "Clé: ".$key.", Valeur : ".$value."<br / >";
        }
?>
```

46

46

Les tableaux multidimensionnels – déclarer

```
<?php
        $arrPerson = array(
                        1 => array(   'firstname' => 'Jessy',
                                      'name' => 'Brown',
                                      'tel' => '00001111'),
                        2 => array(   'firstname' => 'Sharon',
                                      'name' => 'Dain',
                                      'tel' => '00221111'),
                        3 => array(   'firstname' => 'Marta',
                                      'name' => 'Blanca',
                                      'tel' => '003311111')
                        );
?>
```

47

47

Les tableaux multidimensionnels - afficher

```
<?php
        echo $arrPerson[1]['firstname']; //Jessy

        foreach($arrPerson as $cle1 => $valeur1){
                echo "personne n°:" . $cle1 . "<br />";
                foreach ($valeur1 as $cle2=>$valeur2)  {
                        echo "Clé : ".$cle2 .", Valeur: " . $valeur2 . "<br />\n";
                }
        }
?>
```

48

48

Les fonctions dédiées aux tableaux

$arrStudent = array ("Alissa", "Marianne", "Mickael", "Shania", "Odile", "Stefanie", "Marianne");

- count() : permet de connaitre la taille d'un tableau
  $intNb =count($arrStudent);
- in_array() : indique si une valeur appartient à un tableau ou non
  $boolInArray = in_array("Marianne", $arrStudent);
- array_search() : permet de connaitre la clé correspondante de l'élément recherché
  $intKeyIndex = array_search("Marianne", $arrStudent);
- array_count_value() : Nombre d'occurrences d'un élément
  $arrOccurence = array_count_values($arrStudent);
- array_unique() : enlève les doublons
  $arrStudent = array_unique($arrStudent);

49

49

SUPERGLOBALES

50

Variables superglobales

- Ecrites en majuscules
- Commencent par un _
- Sont des tableaux (array)
- Automatiquement créées par PHP à chaque fois qu'une page est chargée

Pour afficher le contenu :

```
<?php
        echo("<pre>");
        print_r($_POST);
        echo("</pre>");
?>
```

51

51

Les variables prédéfinies

- $GLOBALS
  - Contient une référence sur chaque variable qui est disponible dans l'environnement d'exécution global. Les clés de ce tableau sont les noms des variables globales.
- $_GET, $_POST et $_COOKIE
  - Ce sont les tableaux des variables fournies par le protocole HTTP en méthode GET et POST, et dans les cookies
- $_FILES
  - C'est le variables fournies par le protocole HTTP, suite à un téléchargement de fichier
- $_ENV
  - Les variables fournies par l'environnement
- $_REQUEST
  - Les variables fournies au script par n'importe quel mécanisme d'entrée et qui ne doit recevoir une confiance limitée
- $_SESSION
  - Les variables qui sont actuellement enregistrées dans la session attachée au script

52

52

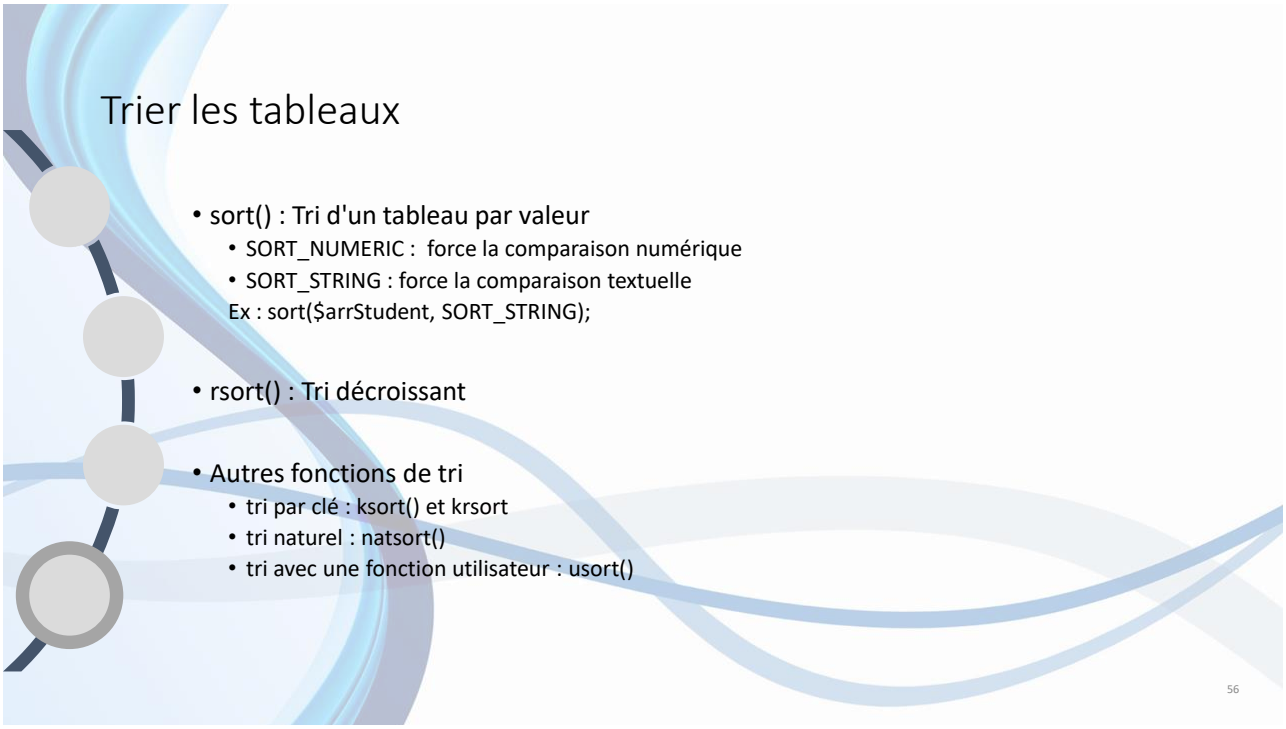
Les variables d'environnement

- $_SERVER est un tableau des variables fournies par le serveur web
  - $_SERVER['REQUEST_METHOD'] : La méthode d'appel (POST ou GET)
  - $_SERVER['SERVER_NAME'] : Nom du serveur
  - $_SERVER['SERVER_ADMIN'] : L'email de l'administrateur du serveur
  - $_SERVER['SERVER_ADDR'] : L'Adresse IP du serveur
  - $_SERVER['QUERY_STRING'] : Les paramètres indiquées à votre script
  - $_SERVER['REMOTE_PORT'] : Port HTTP de la requête
  - $_SERVER['REMOTE_ADDR'] : Adresse IP de l'internaute
  - $_SERVER['REQUEST_URI'] : Chemin du script
  - $_SERVER['PATH_TRANSLATED'] : Chemin physique (complet) du script
  - $_SERVER['HTTP_USER_AGENT'] : User agent du navigateur du client
  - $_SERVER['HTTP_REFERER'] : L'URL de la page d'où provient l'internaute
  - $_SERVER['HTTP_HOST'] : Le nom de domaine où est exécuté le script
  - $_SERVER['HTTP_ACCEPT_LANGUAGE'] : Langue acceptée par le navigateur de l'internaute
  - $_SERVER['DOCUMENT_ROOT'] : Adresse de la racine du serveur

Travaux Pratiques

Récupérer les informations dans un formulaire

55



56

Créer un tableau en fonction d'une chaîne

- Utilisation de la fonction explode("séparateur", tableau)

```
<?php
        $strFruitList = "pomme,orange,kiwi,poire";
        $arrFruits    = explode(",", $strFruitList);

        echo "Le tableau des fruits:";
        echo "<pre>";
        print_r($arrFruits);
        echo "</pre>";
?>
```

57

57

Créer une liste en fonction d'un tableau

- Utilisation de la fonction implode(tableau, "séparateur")

```
<?php
        $arrFruits = array("pomme", "orange", "poire");
        $strFruitList = implode($arrFruits, ",");
        echo $strFruitList;
?>
```

58

58

Division d'un tableau en plusieurs

- La fonction array_chunk() sépare un tableau en plusieurs dont la taille est déterminée par le deuxième argument de la fonction.

```
<?php
	$arrStudent = array ("Alissa", "Mickael", "Shania", "Odile", "Stefanie", "Marianne", "Sophie",
"Marco");

	//Découpage du tableau
	$arrBin = array_chunk($arrStudent, 2);

	echo "Le tableaux des binômes:";
	echo "<pre>";
	print_r($arrBin);
	echo "</pre>";

	echo "Le premier binôme:";
	echo "<pre>";
	print_r($arrBin[0]);
	echo "</pre>";
?>
```

59

59

Fusion de tableaux

- array_merge() fusionne les tableaux

```
<?php
	// suite du code précédent

	//regroupement des deux premiers binômes
	$arrStudentGroup = array_merge($arrBin[0], $arrBin[1]);

	echo "Le regroupement des deux binôme:";
	echo "<pre>";
	print_r($arrStudentGroup);
	echo "</pre>";
?>
```

60

60

Différence entre tableaux

- array_diff() permet de faire la différence entre deux tableaux

```
<?php
        $arrNatation = array ("Alissa", "Mickael", "Shania", "Odile", "Stefanie","Anaia", "Marianne",
"Sophie","Marco","Zora");
        $arrDanse      = array ("Natacha", "Mickael", "Shania", "Michèle", "Stefanie", "Adrien",
"Patrick","Marco");

        $arrDiff = array_diff($arrNatation, $arrDanse );

        echo "Ces étudiants ne font pas de la danse mais seulement de la natation:";
        echo "<pre>";
        print_r($arrDiff);
        echo "</pre>";
?>
```

61

61

Intersection entre tableaux

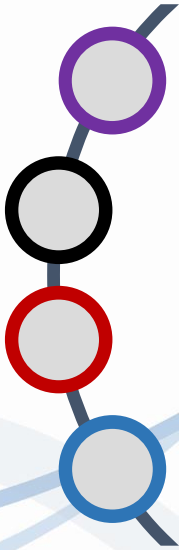
- array_intersect() permet de faire l'intersection entre deux tableaux

```
<?php
    $arrNatation = array ("Alissa", "Mickael", "Shania", "Odile", "Stefanie","Anaia", "Marianne",
    "Sophie","Marco","Zora");
    $arrDanse = array ("Natacha", "Mickael", "Shania", "Michèle", "Stefanie", "Adrien", "Patrick","Marco");

        $arrIntersect = array_intersect($arrNatation, $arrDanse );

        echo "Ces étudiants font à la fois de la danse et de la natation:";
        echo "<pre>";
        print_r($arrIntersect);
        echo "</pre>";
?>
```

62

62

LES BOUCLES



63

63

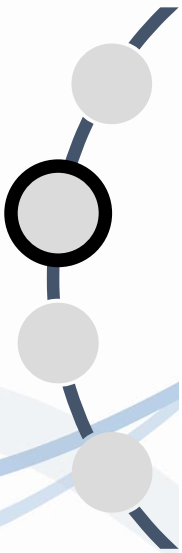
Les opérateurs d'incrémentation et de décrémentation

- Incrémenter une valeur signifie qu'on augmente la valeur d'un certain ordre
- Pour dire que la valeur d'une variable $i a augmenté d'un point, on écrit :

```
$i = $i+1 ;        $i += 1 ;        $i++ ;
```

- Décrémenter une valeur veut dire qu'on diminue la valeur de $i d'un certain nombre de points
- La syntaxe est la même que celle de l'incrémentation, sauf qu'à la place du signe + (plus) on met un signe – (moins).

- Il est tout à fait possible de faire : $i += 2; pour dire que la valeur de $i augmentera de deux points.



64

64

Boucle while

- Syntaxe de while :

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

- La boucle while() signifie que l'on va répéter un bloc d'instructions tant que la condition passée en paramètre reste vraie (TRUE)
- Lorsque celle-ci deviendra fausse (FALSE), le programme sortira de la boucle

- On affiche tous les nombres pairs qui sont inférieurs à 20

```
<?php
        $i = 0;
        while($i < 20) {
                echo $i."<br />";
                $i += 2 ;
        }
?>
```

- Attention a ne pas oublier d'incrémenter la variable $i, sinon la boucle est infinie

65

65

Boucle for

- Il faut connaître par avance la condition d'arrêt (la valeur qui rendra la condition fausse et stoppera la boucle)

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

- init : valeur de départ
- condition : condition d'arrêt de la boucle
- incrémentation (ou décrémentation) : pour mettre à jour le compteur de la boucle

- Exemple :

```
<?php
        // Boucle générant la table de multiplication du 8
        for($i=0; $i<=10; $i++){
                echo "8 x " . $i . " = " . (8*$i) . "<br/>";
        }
?>
```

66

66

La boucle do-while en PHP

- L'instruction do{ ... } while() est une alternative à l'instruction while()
- Elle permet de tester la condition après la première itération et exécution du premier bloc d'instructions

- Syntaxe :
- <?php
- do{
  // bloc d'instructions;
  } while(condition);
- ?>

- Exemple :

```
<?php
    // Déclaration et initialisation du compteur
    $i = 1;
    // Boucle générant la table de multiplication du 8
    do{
        echo "8 x ". $i . " = " . (8*$i) . "<br/>";
        // Incrémentation du compteur
        $i++;
    } while($i <= 10);
?>
```

67

67

Break et Continue

- break
  - permet de sortir de la boucle courante
  - permet aussi de gérer des événements plus exceptionnels, comme des erreurs: en cas d'erreur, on quitte la boucle et on affiche un message.

- continue
  - permet de sauter les instructions de l'itération courante, afin de passer directement à l'itération suivante

```
<?php
    $i = 0;
    while($i < 20){
        if (! ($i % 2) ) {
            continue;
        }
        echo $i . "<br />";
        $i+=1 ;
    }
?>
```

68

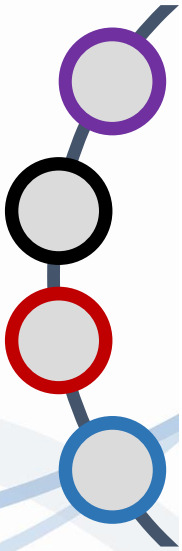
68

Travaux Pratiques

Exercices sur les boucles

69

LES FONCTIONS



70

Définition de la fonction

```
function Nom_de_la_fonction(){
        Instructions
}
```

- Une fonction peut ne pas retourner une valeur. Dans ce cas, on parle généralement de procédure.

- Exercice
  - Créer une fonction qui s'appelle hello et qui affiche bonjour tout le monde !

```
<?php
        function hello()  {
                echo "Bonjour tout le monde !";
        }
        hello();
?>
```

71

71

Fonction avec paramètres

- L'ordre dans la disposition des paramètres dans la définition de la fonction a son importance

- Créer une fonction qui s'appelle division, qui effectue la division de deux nombres passés en paramètre et retourne le résultat

```
<?php
    function division($floatNb1, $floatNb2) {
        if ($floatNb2 == 0){
                echo "Division par 0 impossible";
                return false;
        }else{
                $floatResult=$floatNb1 /$floatNb2;
                return $floatResult;
        }
    }
    $floatResultDiv = division(100,50);
    echo $floatResultDiv;

?>
```

72

72

Valeurs par défaut des paramètres d'une fonction

- Des valeurs par défaut peuvent être assignés aux paramètres

```
<?php
function division($floatNb1=10, $floatNb2=2) {
        if ($floatNb2 == 0){
                echo "Division par 0 impossible";
                return false;
        }else{
                $floatResult=$floatNb1 /$floatNb2;
                return $floatResult;
        }
}
?>
```

- Quels sont les résultats des instructions suivantes ?

```
<?php
        echo division();

        echo division(100);

        echo division(100,50);
?>
```

73

73

Travaux Pratiques

Créer les fonctions suivantes

- Dans un fichier appelé price.php
  - Une fonction diff qui renvoie le résultat de la différences de deux prix
  - Une fonction calcTTC qui renvoie le calcul entre prix HT et une valeur de TVA
  - Une fonction calcDiscount qui renvoie le calcul du prix de la remise en fonction d'un prix de base et d'une valeur de remise
- Tester les fonctions en les appelant

74

74

Fonction par référence

- Par défaut, les arguments sont passés à la fonction par valeur (changer la valeur d'un argument dans la fonction ne change pas sa valeur à l'extérieur de la fonction)
- Pour passer un paramètre en référence, le faire précéder de '&' dans la déclaration de la fonction

```
<?php

    function addSomeExtra(&$strTxt) {
        $strTxt .= ', et un peu plus.';
    }

    $strTxt  = 'Ceci est une chaîne';
    addSomeExtra($strTxt);
    echo  $strTxt;

?>
```

75

75

INCLURE DES FICHIERS

76

76

Inclure un fichier avec la fonction include

- La fonction include() permet d'inclure un fichier PHP dans un autre
- La fonction en question se remplace elle-même par le contenu du fichier spécifié

77

77

Travaux Pratiques

Créer le fichier bill.php

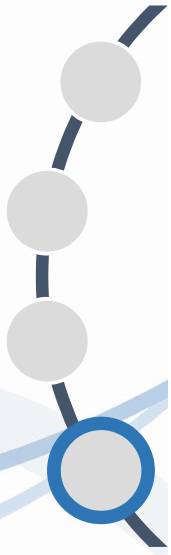
- Créer le fichier bill.php qui doit afficher les informations suivantes, en incluant le fichier price.php :
  - Prix HT:100€
  - Prix TTC:120€
  - Remise 10%:12€
  - Prix à payer:108€

78

78

Inclure des fichiers avec la fonction require

- L'instruction require fonctionne de la même façon qu'include
- Si le fichier n'existe pas :
  - Include déclenche un warning et continue l'exécution du script
  - Require déclenche une erreur fatale et le script est interrompu

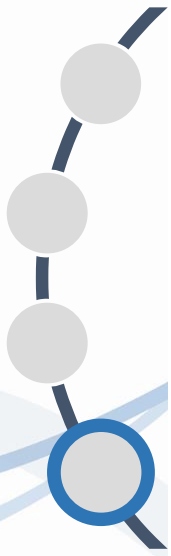


79

79

require_once et include_once

- Deux autres fonctions require_once et include_once sont les même que require et include mais elles s'assurent que le fichier qu'on inclue ne l'a pas déjà été
- Il est conseillé de les utiliser lorsque vous voulez être sûrs que le fichier ne sera inclus qu'une seule fois
- Un fichier rempli de fonctions, par exemple, ne peut être inclus qu'une seule fois (par exécution). Les fonctions étant déjà déclarées, elle ne peuvent l'être une seconde fois, et cela génèrerait une erreur.



80

80

Des question ?

Cours suivant…

A suivre : PHP BDD

81