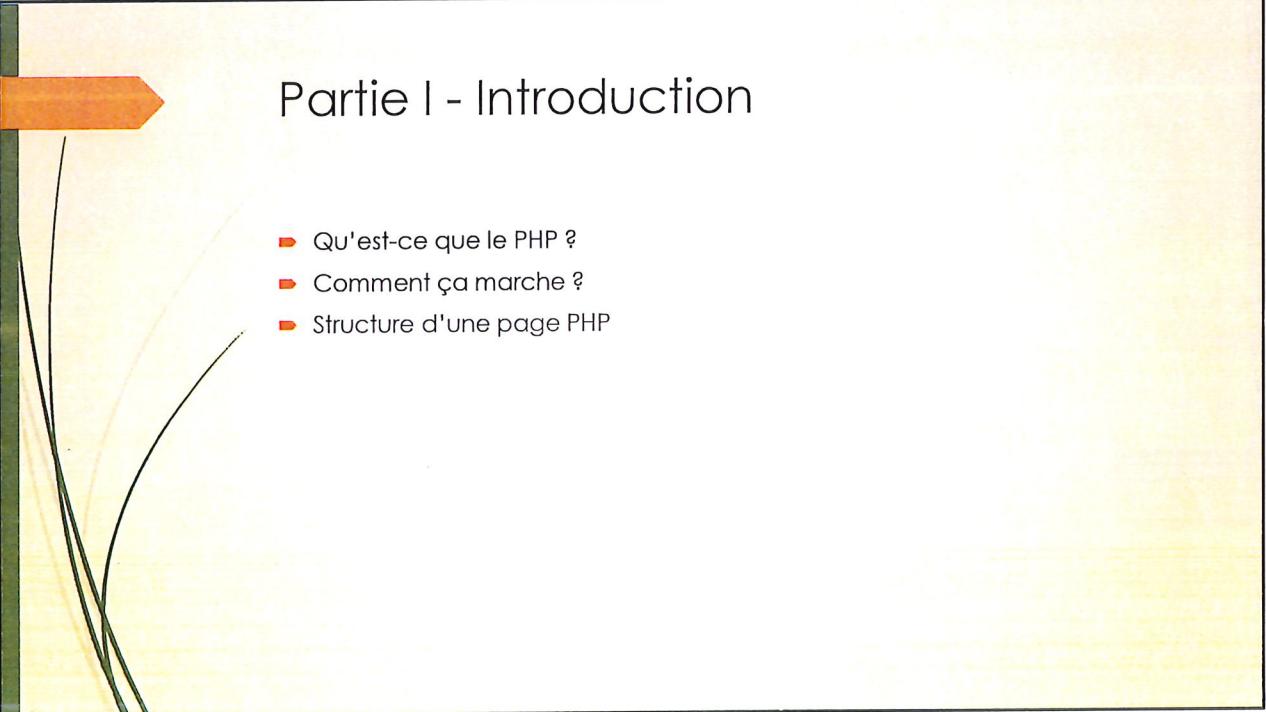


PHP

Pages web dynamiques

Philippe Schmutzler - ITS4U © 2019



Partie I - Introduction

- ▶ Qu'est-ce que le PHP ?
- ▶ Comment ça marche ?
- ▶ Structure d'une page PHP

Qu'est-ce que le PHP ?

- ▶ PHP Hypertext Preprocessor
- ▶ Langage de programmation
- ▶ Génère des pages Web
- ▶ Site statique vs. site dynamique

Diapositive 3

TS0 PHP est un langage de script côté serveur conçu pour le développement web mais aussi utilisé comme langage de programmation généraliste.

Il permet de générer des pages Web dynamiques.

On parle de page Web dynamique car contrairement au page Web statique le contenu du site change en fonction de l'utilisateur.

Un contenu différent est généré et offre une interaction à l'utilisateur par le biais d'une programmation avancée et des bases de données en plus du HTML.

Timothée SICCHIA; 2021-12-15T10:16:28.672

Comment ça marche ?

- ▶ Architecture client-serveur
 - ▶ Client = Effectue une demande
 - ▶ Serveur = Répond à la demande
- ▶ Dans le cas du Web :
 - ▶ Le client demande à consulter une page web
 - ▶ Le serveur fournit la page web demandée

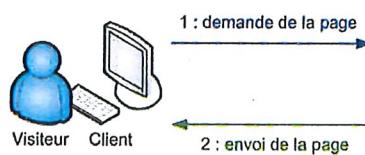
Diapositive 4

TS0 Architecture client-serveur est identique à ce qu'on peut retrouver dans les protocoles réseaux.
Un client, l'utilisateur qui souhaite accéder au site, émet une demande puis le serveur la reçoit pour ensuite accepter, ou non, de fournir la page web demandée.

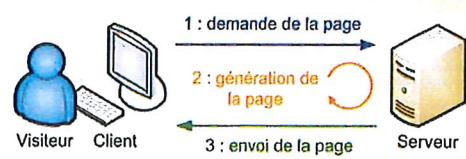
Timothée SICCHIA; 2021-12-15T10:17:44.715

Comment ça marche ?

Site statique



Site dynamique



Diapositive 5

TS0 Schéma pour mieux comprendre le fonctionnement d'une communication client-serveur mais également saisir la différence qui oppose le site statique et dynamique.
Timothée SICCHIA; 2021-12-15T10:18:24.681

Question ?

- ▶ Pour un site Web :
 - ▶ Qui est le client ?
 - ▶ Qui est le serveur ?

Diapositive 6

TS0 Reposons-nous ces questions élémentaires pour la suite.
Timothée SICCHIA; 2021-12-15T10:19:00.223

Structure d'une page PHP

- Fichier avec l'extension .php
- Débute par : <?php et termine par ?>
- Contient une série d'instructions
- Peut être placé dans du code HTML
- Exemple :

```
<?php  
    // instructions  
    echo 'Hello World';  
?>
```

Diapositive 7

TS0 Pour utiliser PHP il nous faut nommer l'extension de notre en .php

Pour espérer le faire fonctionner, on range notre code dans une balise qui prendra toute notre page de programmation en php, <?php et qui se clôture par ?>

Dans cette balise, nous noterons une série d'instructions. Par exemple dans cet exemple, le fait de demander d'écrire Hello Word.

Timothée SICCHIA; 2021-12-15T10:21:59.716

Partie II : Serveur

- ▶ WAMP Server
- ▶ Héberger des pages Web

Diapositive 8

TS0 À vous de jouer.

Afin de pleinement utiliser PHP, vous devrez vous munir d'un serveur local.

Timothée SICCHIA; 2021-12-15T10:23:01.693

WAMP Server

- ▶ WAMP Server :
 - ▶ Windows
 - ▶ Apache (serveur web)
 - ▶ MySQL (serveur de base de données)
 - ▶ PHP
- ▶ Environnement de développement prêt à l'emploi

Diapositive 9

TS0 À l'instar d'un LAMP, WAMP pour sa part est un système de distribution Microsoft contenant également Apache qui sera notre serveur Web, MySQL pour notre serveur de base de données puisque nous avons énoncé précédemment que PHP pouvait travailler avec des bases de données, et évidemment PHP.

Timothée SICCHIA; 2021-12-15T10:24:47.484

Exercice

- ▶ Installer Easy PHP
- ▶ Vérifier qu'il fonctionne
 - ▶ A partir de la VM (en local) : <http://localhost/>
 - ▶ A partir de Easy PHP : ouvrir le logiciel, faire F7 et naviguer parmi vos fichiers (ranger dans dossier www du programme)

Diapositive 10

TS0 Vérifier que votre environnement de travail soit fonctionnel.

Dans notre cas, nous allons utiliser le logiciel easyphp pour pouvoir continuer à travailler sur notre machine.

Timothée SICCHIA; 2021-12-15T12:25:18.539

TSO

Héberger des pages Web

- ▶ Répertoire www
- ▶ Apache est multi-site
- ▶ Organisation par projet
- ▶ Fichiers index.html / index.php

Diapositive 11

TS0 Pour héberger des pages Web, il faut utiliser le répertoire www.

Apache est un serveur Web multi-site, il est donc possible d'en héberger plusieurs mais il vaut mieux s'organiser selon nos différents projets par peur de ne plus s'y retrouver.

À l'instar de note fichier index.html faire de même avec index.php

Timothée SICCHIA; 2021-12-15T10:31:19.068

Exercice

- ▶ Créer un nouveau projet
- ▶ Héberger votre page web
- ▶ Vérifier qu'elle est accessible depuis l'hôte

Diapositive 12

TS0 À vous de jouer en créant un nouveau projet en réussissant à héberger votre page Web puis vérifier son accessibilité depuis l'hôte.
Timothée SICCHIA; 2021-12-15T10:47:26.046



Partie III : Les instructions

- ▶ L'instruction « echo »
- ▶ Les variables
- ▶ Les conditions
- ▶ Les boucles

L'instruction « echo »

- ▶ Instruction = commande
- ▶ L'instruction « echo » permet d'afficher du texte
- ▶ Exemples :

```
<?php echo "Ceci est du texte"; ?>
<?php echo "Ceci est du <strong>texte</strong>"; ?>
```

Diapositive 14

TS0 L'instruction echo permet d'écrire (afficher) du texte comme avec l'exemple ci-dessous.

On remarque qu'il est donc impossible d'y inclure des balises HTML puisqu'elles se situent entre les guillemets.

Timothée SICCHIA; 2021-12-15T10:48:22,074

L'instruction « echo »

```
<!DOCTYPE html>
<html>
<head>
    <title>Notre première instruction : echo</title>
    <meta charset="UTF-8" />
</head>
<body>
    <h2>Affichage de texte avec PHP</h2>
    <p>
        Cette ligne a été écrite entièrement en HTML.   

        <?php echo "Celle-ci a été écrite entièrement en PHP."; ?>
    </p>
</body>
</html>
```

Diapositive 15

TS0 L'instruction echo prend une toute autre sémantique lorsqu'elle est placée dans la balise <?php ?>
Timothée SICCHIA; 2021-12-15T12:26:07.919

Les variables

- ▶ Information stockée en mémoire, définie par un nom et une valeur
- ▶ On distingue 4 principaux types de données :
 - ▶ Les chaînes de caractères
 - ▶ Les nombres entiers
 - ▶ Les nombres décimaux
 - ▶ Les booléens

Diapositive 16

TS0 Le langage PHP travail avec des variables.

Des variables peuvent être défini comme étant des informations stockées en mémoire.

Les variables sont définies par un nom et stockent une valeur.

Plusieurs types de variables existent.

Nous avons des variables qui contiennent :

- une chaîne de caractères
- un nombre entier
- un nombre décimal
- un booléen (vrai/faux)

Timothée SICCHIA; 2021-12-15T12:28:37.924

Les variables

```
<?php  
    $name = 'Jon';  
    $age = 22;  
    $size = 1.73;  
    $alive = true;  
?>
```

Diapositive 17

TS0 Comme précédemment citées, définir quelles sont les types de variables représentées.
Timothée SICCHIA; 2021-12-15T12:29:11.051

Les variables

- Concaténation : assembler des chaînes de caractères

```
<?php  
    $name = 'Jon';  
    $age = 22;  
  
    $text = "Il s'appelle $name et a $age ans.";  
    $text = 'Il s\'appelle ' . $name . ' et a ' . $age . ' ans.';  
?>
```

Diapositive 18

TS0 Une concaténation permet de former une chaîne, liaison, en mettant bout à bout au moins deux variables dans notre cas.

Par exemple, nous avons défini une variable qui contient une chaîne de caractère mais affiche également par son intermédiaire deux autres variables, le nom et l'âge.

Pour utiliser la concaténation, il faut prêté attention à espacer les variables par des points et délimiter le texte avec des apostrophes.

Timothée SICCHIA; 2021-12-15T12:32:12.318

Les variables

► Expressions et calculs :

```
<?php  
    $result = 8 + 2;  
    $result = 12 - 2;  
    $result = 5 * 2;  
    $result = 20 / 2;  
  
    $pi = 3.14;  
    $radius = 5;  
    $area = $pi * $radius * $radius;  
?>
```

Diapositive 19

TS0 Une variable peut également stocker le résultat d'un calcul si on lui présente l'opération.

Par exemple, \$result contiendra en premier lieu le résultat de la somme de 8 + 2 soit 10, et pareil pour les autres opérations même si elles sont différentes, le résultat restera le même.

On peut également calculer l'aire d'un cercle à l'aide de variable stockant des nombres entiers ou décimaux et en les calculant ensemble.

Timothée SICCHIA; 2021-12-15T12:36:32.817

Les variables

- ▶ Tableaux numérotés / associatifs

```
<?php  
    $fruits = ['Cerise', 'Pomme', 'Banane'];  
  
    $person['name'] = 'Alice';  
    $person['age'] = 29;  
?>
```

Diapositive 20

TS0 Un tableau associatif est un tableau qui va utiliser des clefs textuelles qu'on va associer à chaque valeur.

Dans notre cas, la variable \$fruits va contenir une liste de divers fruits.

Tandis que la variable \$person sera composée par des clefs ou indices du tableau tels que name ou age renfermant leurs informations respectives.

Timothée SICCHIA; 2021-12-15T12:41:21.943

Les conditions

- ▶ Exécuter du code de façon conditionnelle
- ▶ Structure de type « if... else... »
- ▶ Evaluation d'une expression booléenne
- ▶ Opérateurs de comparaison

Opérateur	Description
<code>==</code>	Est égal à
<code>!=</code>	Est différent de
<code><</code>	Est inférieur à
<code><=</code>	Est inférieur ou égal à
<code>></code>	Est supérieur à
<code>>=</code>	Est supérieur ou égal à

Diapositive 21

TS0 Comme dans tous langages de programmation il y a des conditions.

En plus des exemples d'opérateurs de comparaison qui suivent, nous pouvons également profiter de structure de type si (if) ou sinon si (if else)

Timothée SICCHIA; 2021-12-15T12:57:10.938

Les conditions

```
<?php  
    $age = 22;  
  
    if ($age >= 18) {  
        echo 'Il est majeur';  
    }  
    else {  
        echo 'Il est mineur';  
    }  
?>
```

Diapositive 22

TS0 Voici un exemple de conditions.

Si l'âge est supérieur ou égal à 18, alors il faut afficher "Il est majeur". Au cas échant, sinon, il faudra afficher "Il est mineur".

Nous sommes sur une structure else if, sinon si. Dans ces conditions il faut prêter une attention particulière aux accolades.

Timothée SICCHIA; 2021-12-15T12:58:58.203

Les conditions

- ▶ Conditions multiples
- ▶ Opérateurs logiques :
 - ▶ AND
 - ▶ OR
 - ▶ NOT

Opérateur	Symbol
AND	&&
OR	
NOT	!

Diapositive 23

TS0 Il existe également des conditions multiples. Multiples car elles permettent de faire un choix parmi deux ou plusieurs propositions.

ET, OU, ou PAS sont des opérateurs logiques.

Timothée SICCHIA; 2021-12-15T13:00:37.791

Les conditions

```
<?php  
$temperature = 30;  
  
if ($temperature >= 0 && $temperature <= 100) {  
    echo 'liquide';  
}  
  
$solid = $temperature < 0;  
$gaseous = $temperature > 100;  
  
if ($solid || $gaseous) {  
    echo 'Solide ou gazeux';  
}  
>
```

Diapositive 24

- TS0** Il est ensuite possible de multiplier les conditions.
Dans cet exemple, la température est à 30 degrés.
On conditionne que "si la température est supérieure ou égale à 0 et qu'elle est également inférieure ou égale à 100" ce qui est le cas puisqu'elle est comprise dans cette plage... Alors écrire "Liquide".
Mais si elle est inférieur à 0 c'est solide et si elle est supérieur à 100 c'est gazeux. On utilise un "Ou" qui est une addition de ces deux possibilités pour afficher "Solide ou gazeux".
Timothée SICCHIA; 2021-12-16T07:48:05.413

Les boucles

- ▶ Répéter des instructions
- ▶ 3 types de boucles :
 - ▶ While (nombre d'itérations inconnues)
 - ▶ For (nombre d'itérations connues)
 - ▶ Foreach (parcours de tableaux)

Diapositive 25

TS0 Étudions les boucles.

Timothée SICCHIA; 2021-12-16T07:49:59.471

Les boucles

```
<?php  
    $limit = 100;  
    $nb = 1;  
    $result = $nb;  
  
    while ($result < $limit) {  
        // $nb = $nb + 1;  
        $nb++;  
        $result = $result + $nb;  
    }  
  
    echo "Il faut additionner les nombres de 1 à $nb pour dépasser $limit";  
?>
```

Diapositive 26

TS0 C'est un exemple de boucle, d'une itération qui permet d'additionner le résultat qui est la somme de lui-même + le nombre pour parvenir à la limite fixée à 100.

While "tant que \$result est inférieur à \$limit faire..."

Timothée SICCHIA; 2021-12-16T07:52:48.989

Les boucles

```
<?php  
    echo '<nav><ul>';  
  
    for ($chapter = 1; $chapter <= 5; $chapter++) {  
        echo '<li><a href="#" . $chapter . '">Chapitre ' . $chapter .  
'</a></li>';  
    }  
  
    echo '</ul></nav>';  
?>
```

Diapositive 27

TS0 Un exemple de boucles pour noter le numéro de chapitre.

On remarque dans cette structure qu'il est possible d'ajouter sa variable et condition après un "for".

On défini la variable \$chapter sur 1, et tant que \$chapter est inférieur ou égal à 5 (il devra dépasser ce chiffre) alors noter le numéro du chapitre correspondant et finir par +1 à la variable \$chapter.

Timothée SICCHIA; 2021-12-16T07:55:16.320

Les boucles

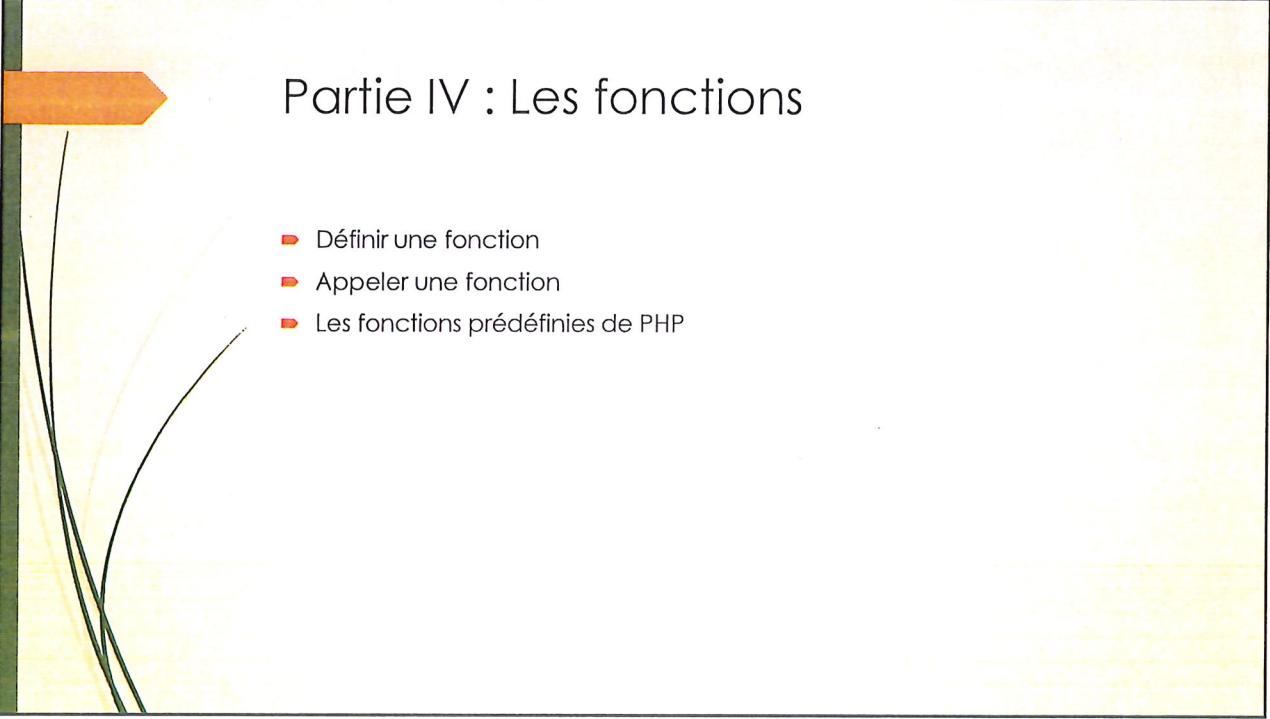
```
<?php  
    $languages['php'] = 'PHP';  
    $languages['html'] = 'HTML';  
    $languages['css'] = 'CSS';  
    $languages['js'] = 'Javascript';  
    $languages['java'] = 'Java';  
  
    foreach ($languages as $extension => $name) {  
        echo "L'extension $extension est pour le langage $name<br>";  
    }  
?>
```

Diapositive 28

TS0 Il est également possible de faire des boucles en utilisant des tableaux associatifs.

"foreach" correspond à "pour chaque" \$languages qui a comme \$extensions un \$name alors écrire :
"L'extension js est pour le langage Javascript".

Timothée SICCHIA; 2021-12-16T08:02:22.374



Partie IV : Les fonctions

- ▶ Définir une fonction
- ▶ Appeler une fonction
- ▶ Les fonctions prédéfinies de PHP

Définir une fonction

- ▶ Une fonction exécute un ensemble d'instructions
- ▶ Elle est caractérisée par :
 - ▶ un nom (qu'on utilisera pour l'appeler)
 - ▶ des paramètres (variables dont a besoin la fonction)
- ▶ Elle peut renvoyer un résultat

```
<?php
    function hello($name) {
        echo 'Bonjour ' . $name . ' !<br />';
    }
?>
```

Diapositive 30

TS0 Une fonction correspond à une série d'instructions qui ont été créées pour effectuer une tâche précise. Pour exécuter le code contenu dans une fonction, il va falloir appeler la fonction.

Une fonction est toujours caractérisée par un nom et des paramètres.

Les paramètres étant des informations qui ne se situent pas dans la définition de la fonction mais dont nous aurons besoin d'importer pour la faire fonctionner correctement.

Timothée SICCHIA; 2021-12-16T08:37:19.811

Appeler une fonction

```
<?php  
    function hello($name) {  
        echo 'Bonjour ' . $name . ' !<br />';  
    }  
  
    hello('Alice');  
    hello('Bob');  
    hello('Charlie');  
?>
```

Diapositive 31

- TS0** Par exemple, la fonction de nom hello emprunte en paramètre la variable \$name.
La variable \$name est externe à la fonction, sont résultats est défini en paramètre dans les appels de fonction.

Timothée SICCHIA; 2021-12-16T08:41:33.963

Les fonctions prédéfinies de PHP

- ▶ `strlen` : longueur d'une chaîne de caractères
- ▶ `str_replace` : remplace une chaîne par une autre
- ▶ `strtolower` : mettre en minuscules
- ▶ `strtoupper` : mettre en majuscules
- ▶ `date` : récupère la date et l'heure
- ▶ `include` : insère le contenu d'une page PHP

Diapositive 32

- TS0** Il y a des fonctions prédéfinies pour le langage PHP. C'est-à-dire des fonctions que nous n'avons pas besoin de développer pour les utiliser, il nous suffit simplement de les appeler.
Timothée SICCHIA; 2021-12-16T08:44:58.875

Les fonctions prédéfinies de PHP

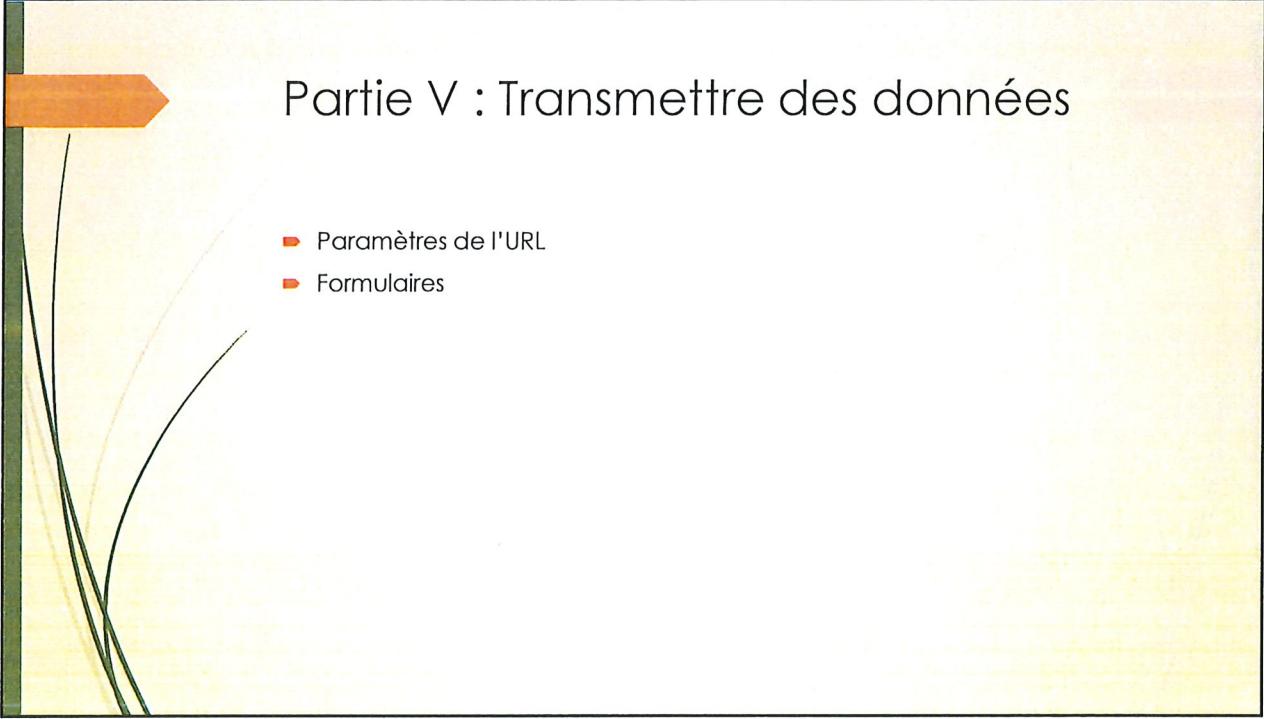
```
<?php
    $length = strlen('Longueur');
    $banana = str_replace('e', 'a', 'Banane');
    $lower = strtolower('MINUSCULE');
    $upper = strtoupper('majuscule');
    $day = date('d/m/Y');
    $hour = date('H:i:s');

    echo "Nous sommes le $day et il est $hour";
    include('menu.php');
?>
```

Diapositive 33

TS0 Nous pouvons stocker ces fonctions et leur paramètres dans nos variables pour les appeler plus facilement dans notre texte.

Timothée SICCHIA; 2021-12-16T08:47:16.419



Partie V : Transmettre des données

- ▶ Paramètres de l'URL
- ▶ Formulaires

Paramètres de l'URL

- ▶ Une URL peut avoir des paramètres
- ▶ Un paramètre est défini par un nom et une valeur
- ▶ Les paramètres sont placés après le symbole « ? »
- ▶ Les paramètres sont séparés par le symbole « & »
- ▶ Exemple : page.php?lang=fr&version=2

Diapositive 35

TS0 En PHP, nous pouvons également utiliser la variable `$_GET` pour obtenir des paramètres à partir d'une chaîne d'URL.

La structure d'une URL est spécifique.

Timothée SICCHIA; 2021-12-16T08:54:38.098

Paramètres de l'URL

- ▶ On peut lire les paramètres de l'URL dans une page PHP
- ▶ `$_GET` : tableau associatif contenant les paramètres
- ▶ Fonction `isset($variable)` : vérifie si une variable est définie

```
<?php  
    if (isset($_GET['lang'])) {  
        echo 'Langue : ' . $_GET['lang'];  
    }  
?>
```

Diapositive 36

TS0 Pour ce cas de figure, "isset" vérifie par exemple si la variable 'lang' existe.
Si elle existe, alors on affiche "Langue :" suivi de la langue utilisée.

Timothée SICCHIA; 2021-12-16T08:57:17.832

Formulaires

- ▶ Élément « **form** » :
 - ▶ Attribut « **action** » : destination du traitement du formulaire
 - ▶ Attribut « **method** » : méthode d'envoi des paramètres
- ▶ Élément « **input** » :
 - ▶ Attribut « **name** » : nom du paramètre envoyé
- ▶ Fonction **htmlspecialchars()**

```
// login.html
<form action="welcome.php" method="post">
    <label>Login :</label>
    <input type="text" name="login" />
</form>

// welcome.php
<?php
    if (isset($_POST['login'])) {
        echo 'Welcome ' . $_POST['login'];
    }
?>
```

Diapositive 37

TS0 PHP peut gérer des formulaires.

Parlons de ce que propose la balise <form>. Concernant "action", sa valeur détermine l'adresse du script qui recevra le contenu du formulaire, une fois transmis. Cet attribut est requis.

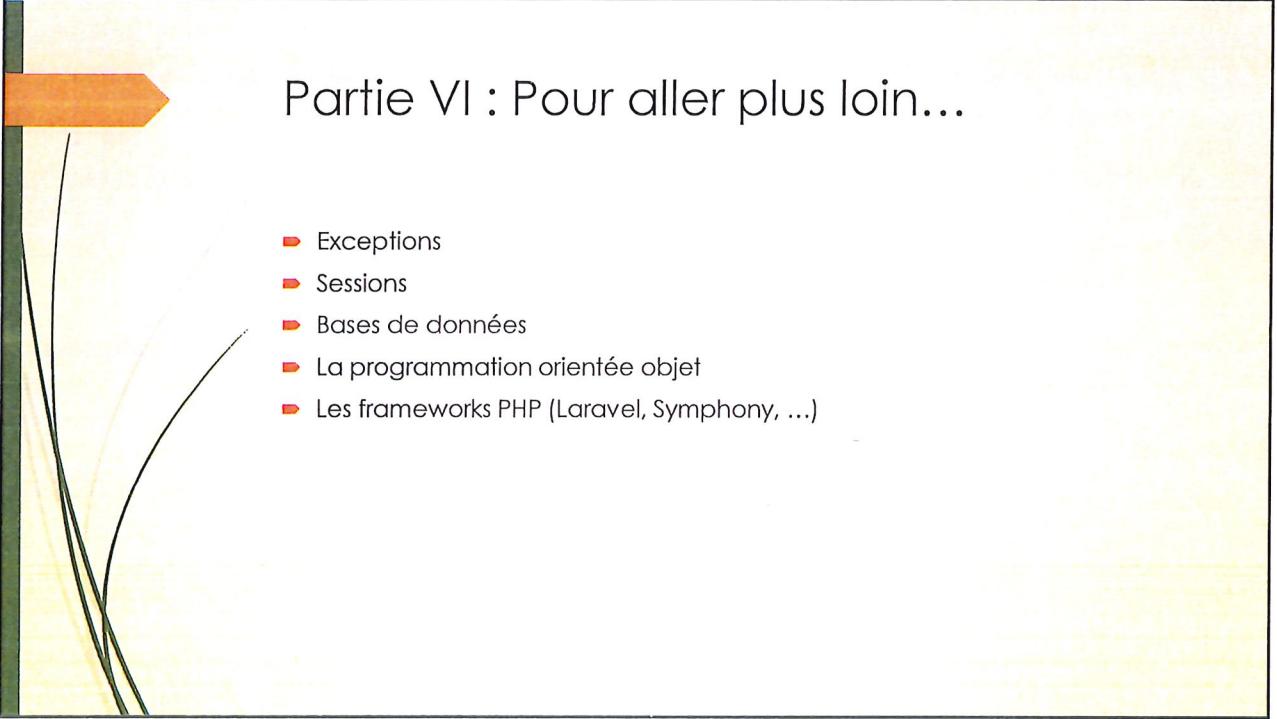
Pour "method", Les valeurs peuvent être "get" ou bien "post". Cet attribut est optionnel et vaut "get" s'il n'est pas noté.

"post" signifie le fait que les variables soient transmises de manière cachée.

Un puisque un formulaire a généralement pour vocation de permettre à l'internaute de saisir des informations on utilise la balise <input> dans laquelle on définit "name" qui est le nom du paramètre à envoyer, soit le login dans notre cas.

la fonction htmlspecialchars est très utile puisqu'elle permet de convertir les caractères spéciaux en entités HTML.

Timothée SICCHIA; 2021-12-16T09:04:14.895



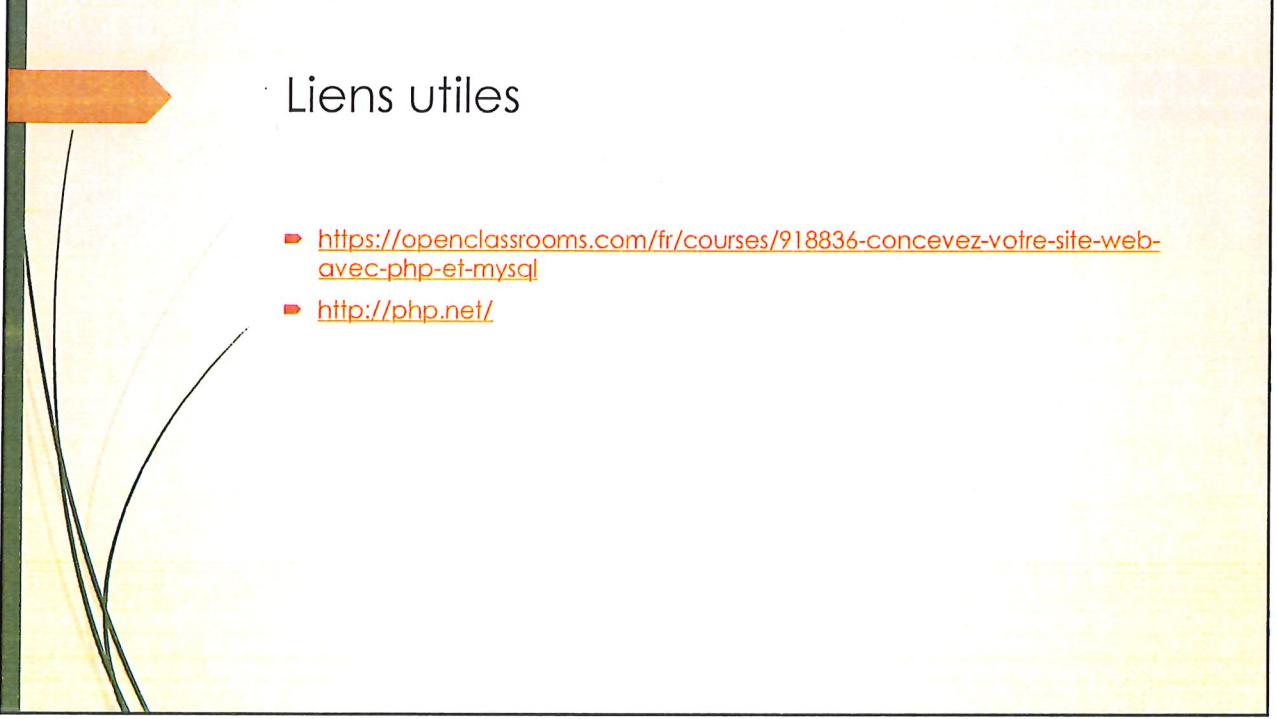
Partie VI : Pour aller plus loin...

- ▶ Exceptions
- ▶ Sessions
- ▶ Bases de données
- ▶ La programmation orientée objet
- ▶ Les frameworks PHP (Laravel, Symphony, ...)



TP

- ▶ Réaliser une calculatrice simplifiée avec :
 - ▶ 2 champs de texte pour les opérandes
 - ▶ 1 liste déroulante pour les 4 opérateurs
 - ▶ Un bouton pour calculer le résultat



Liens utiles

- ▶ <https://openclassrooms.com/fr/courses/918836-concevez-votre-site-web-avec-php-et-mysql>
- ▶ <http://php.net/>