

Chapitre 1(p2): Concepts de base du langage PHP

ING 3

S. NOUFEL

Plan

1. Les Fonctions
2. Les Superglobals
3. Les Formulaires

Les Fonctions

- Description d'une fonctionnalité dépendant éventuellement de paramètres et retournant éventuellement un résultat

Syntaxe pour la déclaration d'une fonction

```
Function nomFonction(param1, param2,  
param3){
```

[Retrun \$resultat;] // optionnel

```
}
```

Exemple

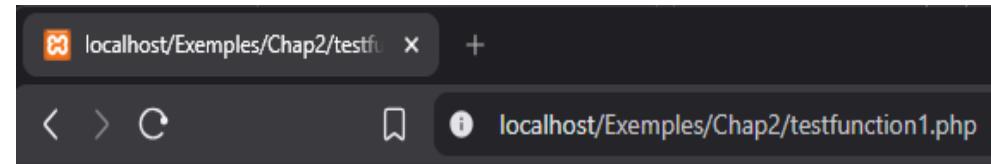
```
<?php

function testFunction1() {
    echo "This is test function 1.<br>";
}

function calculerTTC($prixHT, $tva=1.20){
    $total = $prixHT * $tva;
    return $total;
}

// Appel de la fonction
testFunction1();

$prixHT = 100;
$prixTTC = calculerTTC($prixHT);
echo "\nLe prix TTC est : " . $prixTTC;
?>
```



This is test function 1.
Le prix TTC est : 120

La Portée des Variables (Global)

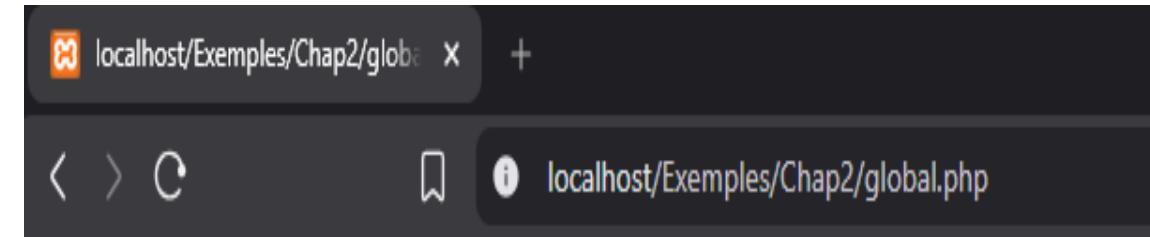
```
<?php  
$taxe = 20; // Globale  
  
function calcul($montant) {  
    return $montant + $taxe;  
}  
  
$resultat = calcul(100);  
echo "Le montant calculé est : " . $resultat;
```



Warning: Undefined variable \$taxe in
C:\xampp\htdocs\Exemples\Chap2\global.php on line 5
Le montant calculé est : 100

La Portée des Variables (Global)

```
1 <?php
2 // $taxe = 20; // Globale
3
4 function calcul($montant, $taxe) {
5     return $montant + $taxe;
6 }
7
8
9 $resultat = calcul(100, $taxe=20);
10 echo "Le montant calculé est : " . $resultat;
```



Le montant calculé est : 120

Passage par valeur vs par adresse

Le passage par valeur: J'envoie un PDF par email. Si vous le modifiez, mon original ne change pas. (Copie)

Le passage par adresse: J'envoie un lien Google Docs. Si vous le modifiez, mon original change aussi. (Pointeur)

Passage par valeur vs par adresse

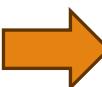
```
<?php

function ajout_valeur($nbr){
    $nbr += 10;
    echo "Valeur à l'intérieur de la fonction : " . $nbr . "<br>";
}

function ajout_reference(&$nbr){
    $nbr += 10;
    echo "Valeur à l'intérieur de la fonction (référence) : " . $nbr . "<br>";
}

$nombre = 5;
echo "Avant appel par valeur : " . $nombre . "<br>";
ajout_valeur($nombre);
echo "Après appel par valeur : " . $nombre . "<br><br>";
$nombre_ref = 5;
echo "Avant appel par référence : " . $nombre_ref . "<br>";
ajout_reference($nombre_ref);
echo "Après appel par référence : " . $nombre_ref . "<br>";
```

Avant appel par valeur : 5
Valeur à l'intérieur de la fonction : 15
Après appel par valeur : 5



Avant appel par référence : 5
Valeur à l'intérieur de la fonction (référence) : 15
Après appel par référence : 15

Exercice

Vous développez un algorithme de trading haute fréquence. Vous disposez d'un portefeuille d'actions.

```
$portefeuille = [  
    "Apple" => 150,  
    "Tesla" => 800,  
    "Bitcoin" => 40000  
];
```

1. Ecrivez une fonction *simulerKrach(\$tab)*:
 - Elle accepte le tableau par **Valeur**.
 - Elle divise tous les prix par 2.
 - Elle affiche le nouveau tableau.
2. Ecrivez une fonction *appliquerHausse(&\$tab)*:
 - Elle accepte le tableau par **Référence**
 - Elle augmente tous les prix de 10%
3. Le Test:
 - Affichez le portefeuille.
 - Appelez *simulerKrach(\$portefeuille)*.
 - *Affichez le portefeuille*
 - *Appelez appliquerHausse(\$portefeuille)*.
 - Affichez le portefeuille

Interaction avec l'utilisateur

Il existe deux types d'interaction avec l'utilisateur

- Passage des paramètres via URL (hyperlien `$_GET`)
- Passage des paramètres via des formulaires (`$_POST`)

`$_GET`: Les paramètres d'URL

URL: profile.php?nom=Saad&age=27



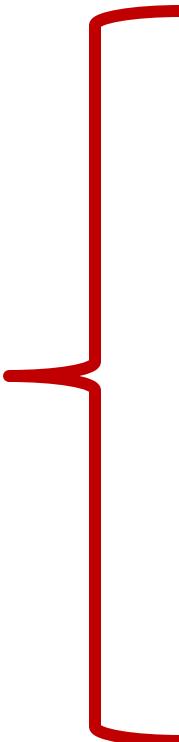
- Le ? Veut dire attention les données commencent
- Le & sépare les variables



- Tout le monde peut modifier l'URL! Ne faites jamais confiance à ces données pour la sécurité.

Exemple

```
Exemples > Chap2 > get.php > ...  
1 <?php  
2 // On récupère la variable 'name' du URL  
3 $user_name = $_GET['name'];  
4  
5 echo "Hello " . $user_name;  
6 ?>
```



localhost/Exemples/Chap2/get.php

Warning: Undefined array key "name" in C:\xampp\htdocs\Exemples\Chap2\get.php on line 3

Hello

localhost/Exemples/Chap2/get.php

localhost/Exemples/Chap2/get.php?name=Saad

Hello Saad

Interaction avec l'utilisateur

Le passage des paramètres via URL se fait de la manière suivante

```
<a href="resultat.php?etudiant=Saad"> etudiant 1</a><br>
```

Page index.html

```
<a href= "./resultat.php?etudiant=saad">etudiant 1</a> <br/>
<a href= "./resultat.php?etudiant=Durant">etudiant 2</a> <br/>
<a href= "./resultat.php?etudiant=Dupont">etudiant 3</a> <br/>
```

Page resultat.php

```
echo "le nom de l'étudiant est = " . $_GET['etudiant'];
```

Exercice

Vous devez créer un script capable de calculer l'IMC(indice masse corporelle) d'un patient instantanément en recevant ses données via l'URL.

Objectif: Créer un script *imc.php* qui:

1. Récupère le **poids** et la **taille** passés dans l'URL.
2. Utilise une fonction *calculerIMC(\$p, \$t)* pour faire le calcul (Formule: *Poids/Taille²*)
3. Utilise une fonction *interpréter(\$imc)* pour afficher l'état de santé
 - Si $imc < 18,5 \Rightarrow$ "Maireur"
 - Si $18,5 \leq imc < 25 \Rightarrow$ "Normale"
 - Si $25 \leq imc < 30 \Rightarrow$ "Surpoids"
 - Si $imc \geq 30 \Rightarrow$ "Obésité"
4. Affiche le résultat final

HINT:

`http://localhost/imc.php?poids=80&taille=1.80`

`$_POST`: Pour les Données Sensibles

- Contrairement à `$_GET`, les données ne voyagent pas dans l'URL.
- Elles sont cachées dans le **CORPS** de la requête HTTP.

Pourquoi l'utiliser?

- **Sécurité visuelle:** Les mots de passe ne s'affichent pas dans l'historique ou la barre d'adresse.
- **Volume:** Pas de limite de taille (on peut envoyer des fichiers, ...)

La Syntaxe (Le Duo HTML/PHP)

L'attribut *name* dans le HTML devient la **CLE** dans le tableau PHP,
L'attribut *method="POST"* est obligatoire.

```
<form method="POST" action="traitement.php">
    <input type="text" name="login" placeholder="Votre email">

    <input type="password" name="pass" placeholder="Mot de passe">

    <button type="submit">Se connecter</button>
</form>
```

Utilisation des formulaires

Pour utiliser un formulaire en PHP, deux étapes sont nécessaires:

- La première est l'écriture de la page Web de création de formulaire. Il s'agit de la page PHP ou HTML qui comprend de code HTML permettant d'afficher le formulaire.
- La seconde étape est la création du script de traitement de formulaire. Il s'agit de la page PHP qui est mentionnée dans la propriété action du formulaire.

Le script du traitement du formulaire a deux tâches principales :

- Stocker les données de champs de formulaire dans des variables
- Traiter ces variables (informations)

Utilisation des formulaires

Remarques

- Les données envoyées par la méthode **Post** seront récupérées dans la variable **`$_POST`**
- Les données envoyées par la méthode **Get** seront récupérées dans la variable **`$_GET`**.

Lire les valeurs d'une liste

- Une liste est créée avec deux balises HTML `<select>` et `<option>`.
- La balise `<select>` possède une propriété importante, Name, qui indique le nom de la liste.
- La propriété **Value** de la balise `<option>` indique la valeur de l'option sélectionnée de la liste qui sera envoyée pour être traitée par une page PHP.

Lire les valeurs d'une liste

Exemples > Chap2 > 🐘 recuperation.php > ...

```
1  <form method="Post" action="recuperation.php">
2  Indiquez le jour de la semaine :<select name="jours_sem">
3  <option value="Dim">Dimanche </option>
4  <option value="Lun">Lundi </option>
5  <option value="Mar">Mardi </option>
6  <option value="Mer">Mercredi </option>
7  <option value="Jeu">Jeudi </option>
8  <option value="Ven">Vendredi </option>
9  <option value="Sam">Samedi </option>
10 </select>
11 <input type=submit value="valider"><input type=reset value=Annuler>
12 </form>
```

Lire les valeurs d'une liste

```
<?php  
$jour = $_POST['jours_sem'];  
echo "Nous sommes le $jour";  
?>
```

Lire les boutons d'option

- Les boutons d'options sont utilisés lorsque vous voulez donner à l'utilisateur la possibilité de sélectionner une et une seule option dans un groupe de boutons.
- Les boutons d'un même groupe doivent avoir le même nom (propriété Name) et des valeurs différentes (propriété Value).
- La valeur transmise à la page PHP, et qui indiquera le choix de l'utilisateur, est la propriété Value du groupe des boutons d'option.

Lire les boutons d'option

Exemples > Chap2 > 📁 boutons.php > 🗂️ form

```
1  <form method="Get" action="boutons.php">
2  Quelle tranche d'âge avez-vous ? <br>
3  <input type=radio name=age value="Moins de 18 ans" checked> Moins de 18 ans<br>
4  <input type=radio name=age value="entre 19 et 25 ans"> Entre 19 et 25 ans<br>
5  <input type=radio name=age value="Plus que 26 ans"> Plus que 26 ans<br>
6  <input type=submit value="valider"><input type=reset  value=Annuler>
7
8  <?php
9  $vage=$_GET["age"];
10 echo "Vous avez choisi la tranche d'âge : $vage";
11 ?>
```

Les valeurs de cases à cocher

- Les cases à cocher sont les contrôles les plus difficiles à gérer dans un formulaire, et ceci est dû au fait qu'un utilisateur peut ne cocher aucune case, comme il peut en cocher plusieurs.
- Un groupe de cases à cocher est créé en attribuant la valeur **checkbox** à la propriété **type** et la même valeur pour la propriété **Name sous forme de tableau**. Seule la valeur de **Value** permet de différencier entre les cases à cocher d'un même groupe.
- Dans un script de traitement de formulaire, les valeurs des cases qui sont cochées sont stockées sous forme de tableau.

Les valeurs de cases à cocher

Exemples > Chap2 > 🐘 case.php > ...

```
1  <form method="Post" action="case.php">
2  Choisir les cours qui vous intéresse : <br>
3  <input type=checkbox name=cours[] value=HTML>HTML
4  <input type=checkbox name=cours[] value=Javascript>Javascript
5  <input type=checkbox name=cours[] value=PHP>PHP
6  <input type=checkbox name=cours[] value=Java>Java
7  <input type=checkbox name=cours[] value=Python>Python
8  <input type=submit>
9  <input type=reset>
10 </form>
11
```

Les valeurs de cases à cocher

```
12 <?php
13 $cours_sel = isset($_POST['cours']) ? $_POST['cours'] : [];
14 $n = count($cours_sel);
15 if($n > 0) {
16     echo "Vous avez choisi les cours suivants : <br>";
17     for($i=0; $i < $n; $i++) {
18         echo($cours_sel[$i] . "<br>");
19     }
20 } else {
21     echo("Vous n'avez sélectionné aucun cours.");
22 }
```

Exercice

Créez un script *login.php* qui:

1. Affiche un formulaire (Email + Mot de passe) si l'utilisateur n'a rien envoyé.
2. Si le formulaire est soumis:
 - Récupère l'email et le mot de passe.
 - Vérifie si l'Email = "admin@uir.ac.ma" et mot de passe = "1234".
 - Si oui => Affiche "Bienvenue Admin!"
 - Si non => Affiche "Accès Refusé" en Rouge.

Challenge

Vous développez un module de devis pour une agence de location de voitures. Le client doit pouvoir configurer sa location et obtenir un prix immédiat.

Cahier des Charges:

1. Le Formulaire (HTML):

- Nom du client (Input Text)
- Modèle de voiture (Radio):
 - Citadine (300 DH/jour)
 - Berline (500 DH/jour)
 - 4x4 (800 DH/jour)
- Durée (Input Number): Nombre de jours.
- Options Supplémentaires (Checkboxes – Attention au tableau []):
 - GPS (+50 DH)
 - Siège Enfant (+30 DH)
 - Assurance Tous Risques (+100 DH)

2. La Logique (PHP):

- Créez une fonction *calculerDevis(\$module, \$duree, \$options)*
- Cette fonction doit retourner le prix total TTC.

3. Le Traitement:

- Récupérez les données via `$_POST`
- Gérez le cas où **aucune option** n'est cochée (tableau vide).
- Affichez: "Bonjour [Nom], le montant total pour [X] jours est de: [Total] DH",