

# Langages de l'internet

Web dynamique avancé : quelques outils

Bruno Mascret

**CPE Lyon**

Techniques et langages de l'Internet

# Plan

- 1 PHP avancé
  - Les variables super globales
  - Les sessions
  - Les formulaires

# Introduction

Auteurs

Cours : **Bruno Mascret**

TPs : **Bruno Mascret**

Intervenants : **Bruno Mascret, Françoise Perrin, Mohamed Sallami, Charles Perrin, Vivien Guillet, Timothé Bordiga, Rémi Marengo, Jonas Pauthier**

# Plan

- 1 PHP avancé
  - Les variables super globales
  - Les sessions
  - Les formulaires

# PHP avancé

## Les variables super globales

- Accessibles PARTOUT...
- ...ce qui ne veut pas dire qu'il soit recommandé de les utiliser partout !!!

Principales variables superglobales :

- `$_SERVER` : valeurs données par le serveur (très nombreuses !).
- `$_ENV` : variables d'environnement données par le serveur. ex : "USER", nom de l'utilisateur
- `$_SESSION` : variables pour un client qui sont stockées sur le serveur le temps de sa présence sur le site.
- `$_COOKIE` : variables enregistrés sur l'ordinateur du client. Pas de durée de fin de vie.
- `$_GET` : données envoyées en paramètres dans l'URL.
- `$_POST` : données de la page encapsulées dans la requête (mais visible avec les bons outils quand même !)
- `$_FILES` : liste des fichiers envoyés par la page (majoritairement via formulaire).

# PHP avancé

## Les variables super globales

### Utilisation

```
<?php
```

```
$_COOKIE[preference] = array();  
$_COOKIE['preference'][couleur] = 'vert';
```

```
$_SESSION['nom'] = 'Mascret';  
$_SESSION['mail'] = $_POST['mail'];
```

```
if (isset($_GET['action'])) {  
    if ($_GET['action']=="menu") {  
        affiche_Menu();  
    }  
    elseif ($_GET['action']=="cmd") {  
        afficheCommande();  
    }  
}  
else {  
    echo "rien_a_faire";  
}  
?>
```

# PHP avancé

## Les variables super globales

### Utilisation

<?php

```
echo "<h1>Serveur</h1>";
print_r($_SERVER);
displayTab($_SERVER);
echo "<h1>Env</h1>";
print_r($_ENV);
displayTab($_ENV);

echo "<h1>POST</h1>";
print_r($_POST);
displayTab($_POST);

echo "<h1>GET</h1>";
print_r($_GET);
displayTab($_GET);

function displayTab($t){
    echo "<ul>";
    foreach ($t as $k=>$v){
        echo "<li>". $k. ": ". $v. "</li>";
    }
    echo "</ul>";
}
```

?>

<http://localhost/tp5/lib/main.php>

# PHP avancé

## Les sessions

- Permettent de maintenir une pseudo-connexion
- Optimisation
- Calculs facilités
- Gains en temps... si utilisées !
- Utilisent explicitement la superglobale `$_SESSION` et implicitement `$_COOKIE`



# PHP avancé

## Les variables super globales

### Exemple : Déroulement d'une session

1. Ouverture de session : `session_start()` demande à PHP de générer un numéro unique (PHPSESSID) en hexadécimal (bf4c57deff89006ca44347dfe9egf899 par exemple) qui est transmis au client lors de la réponse par un cookie. A ce stade :
  - le serveur dispose chez lui du numéro de session et des données associées ;
  - le client dispose d'un cookie avec son identifiant.
2. Les scripts PHP sont alors en mesure de créer et stocker des informations sous forme de variables dans l'espace session **côté serveur** (chaînes, nombres, tableaux, etc.) ;
3. A chaque nouvelle requête client, celui-ci transfère son identifiant. **Attention : Il est indispensable d'appeler `session_start()` à chaque nouvelle requête !.** Le programme PHP est en mesure grâce à cet identifiant de retrouver les données chez lui.
4. Après un certain temps d'inactivité, ou lorsque l'utilisateur est déconnecté (`session_destroy()`), les données sont supprimées sur le serveur. Il faut alors commencer une nouvelle session.

# PHP avancé

## Les variables super globales

### Exemple : Déroulement d'une session

1. Ouverture de session : `session_start()` demande à PHP de générer un numéro unique (PHPSESSID) en hexadécimal (bf4c57deff89006ca44347dfe9egf899 par exemple) qui est transmis au client lors de la réponse par un cookie. A ce stade :
  - le serveur dispose chez lui du numéro de session et des données associées ;
  - le client dispose d'un cookie avec son identifiant.
2. Les scripts PHP sont alors en mesure de créer et stocker des informations sous forme de variables dans l'espace session **côté serveur** (chaînes, nombres, tableaux, etc.) ;
3. A chaque nouvelle requête client, celui-ci transfère son identifiant. **Attention : Il est indispensable d'appeler `session_start()` à chaque nouvelle requête !.** Le programme PHP est en mesure grâce à cet identifiant de retrouver les données chez lui.
4. Après un certain temps d'inactivité, ou lorsque l'utilisateur est déconnecté (`session_destroy()`), les données sont supprimées sur le serveur. Il faut alors commencer une nouvelle session.

# PHP avancé

## Les variables super globales

### Exemple : Déroulement d'une session

1. Ouverture de session : `session_start()` demande à PHP de générer un numéro unique (PHPSESSID) en hexadécimal (bf4c57deff89006ca44347dfe9egf899 par exemple) qui est transmis au client lors de la réponse par un cookie. A ce stade :
  - le serveur dispose chez lui du numéro de session et des données associées ;
  - le client dispose d'un cookie avec son identifiant.
2. Les scripts PHP sont alors en mesure de créer et stocker des informations sous forme de variables dans l'espace session **côté serveur** (chaînes, nombres, tableaux, etc.) ;
3. A chaque nouvelle requête client, celui-ci transfère son identifiant. **Attention : Il est indispensable d'appeler `session_start()` à chaque nouvelle requête !.** Le programme PHP est en mesure grâce à cet identifiant de retrouver les données chez lui.
4. Après un certain temps d'inactivité, ou lorsque l'utilisateur est déconnecté (`session_destroy()`), les données sont supprimées sur le serveur. Il faut alors commencer une nouvelle session.

# PHP avancé

## Les variables super globales

### Exemple : Déroulement d'une session

1. Ouverture de session : `session_start()` demande à PHP de générer un numéro unique (PHPSESSID) en hexadécimal (bf4c57deff89006ca44347dfe9egf899 par exemple) qui est transmis au client lors de la réponse par un cookie. A ce stade :
  - le serveur dispose chez lui du numéro de session et des données associées ;
  - le client dispose d'un cookie avec son identifiant.
2. Les scripts PHP sont alors en mesure de créer et stocker des informations sous forme de variables dans l'espace session **côté serveur** (chaînes, nombres, tableaux, etc.) ;
3. A chaque nouvelle requête client, celui-ci transfère son identifiant. **Attention : Il est indispensable d'appeler `session_start()` à chaque nouvelle requête !.** Le programme PHP est en mesure grâce à cet identifiant de retrouver les données chez lui.
4. Après un certain temps d'inactivité, ou lorsque l'utilisateur est déconnecté (`session_destroy()`), les données sont supprimées sur le serveur. Il faut alors commencer une nouvelle session.

# PHP avancé

## Les sessions

### Exemple simple

```
<?php
session_start ();

$_SESSION[ 'prenom' ] = 'Bruno';
$_SESSION[ 'nom' ] = 'Mascret';
$_SESSION[ 'mail' ] = $_POST[ 'mail' ];
?>
```

# PHP avancé

## Les sessions

### Exemple complet

```
<?php
    session_start();

    if (isset($_SESSION['nom'])){
        echo "Session ouverte: " . $_REQUEST['SESSION_NAME'];
    }
    else{
        if (verifLoginPass($_POST['login'], $_POST['pass'])){
            //identifiant et pass ok

            $_SESSION['prenom'] = $_POST['prenom'];
            $_SESSION['nom'] = $_POST['nom'];
            $_SESSION['mail'] = $_POST['mail'];
        }
        else{
            echo "Erreur";
        }
    }

?>
```

# PHP avancé

## Les formulaires

- Transmettent des informations dans des variables POST ou GET
- Les variables sont encapsulées dans la requête HTTP
- La **méthode**, POST ou GET, doit être précisée ;
- il est possible de vérifier les données côté client (javascript, html 5) ;
- il est **OBLIGATOIRE de vérifier les données côté serveur !**

# PHP avancé

## Les formulaires

```

<h1>Un exemple de formulaire html5</h1>
<form oninput="getTotal(this)" method="post" action="lib/main.php?variable=truc">
  <ul>
    <li>
      <label for="full_name">Nom complet&nbsp;</label>
      <input type="text" id="full_name" name="full_name" placeholder="Jane_Doe" required>
    </li>
    <li>
      <label for="email_addr">Adresse e-mail&nbsp;</label>
      <input type="email" id="email_addr" name="email_addr" required>
    </li>
    <li>
      <label for="email_addr_repeat">Confirmez l'adresse e-mail&nbsp;</label>
      <input type="email" id="email_addr_repeat" name="email_addr_repeat" required
        oninput="check(this)">
    </li>
    <li>
      <label for="arrival_dt">Date d'arrivée&nbsp;</label>
      <input type="date" id="arrival_dt" name="arrival_dt" required>
    </li>
    <li>
      <label for="nights">Nombre de nuitées (15 E par nuit)&nbsp;</label>
      <input type="number" id="nights" name="nights" value="1" min="1" max="30" required>
    </li>
  </ul>

```



# PHP avancé

## Les formulaires

```

</li>
    <label for="guests">Nombre d'invités (10 E par invité supplémentaire)&nbsp;  </label>
    <input type="number" id="guests" name="guests" value="1" min="1" max="4" required>
</li>
<li>
    <label for="total">Total&nbsp;  </label>
    <output id="total" name="total">99</output>.00 E
</li>
<li>
    <label for="promo">Code de promotion&nbsp;  </label>
    <input type="text" id="promo" name="promo" pattern="[A-Za-z0-9]{6}"
        title="Le code de promotion contient six caractères alphanumériques.">
</li>
<li>
    <input type="submit" value="Effectuer la réservation" />
</li>
</ul>
</form>

```

<http://localhost/tp5/formulaire.html>

# PHP avancé

## Les formulaires

```

/* Repris et adapté © par Bruno Mascret de http://dmouronval.developpez.com/tutoriels/html/formulaires.html /
 * Les trois auteurs et traducteur
 *
 * Jan Kleinert *
 * Traducteur : Didier Mouronval
 * Bovino
 *
 * Licence creative commons By
 * http://creativecommons.org/licenses/by/3.0/deed.fr
 */

function check(input) {
    if (input.value != document.getElementById('email_addr').value) {
        input.setCustomValidity('Les_deux_adresses_email_ne_correspondent_pas. ');
    } else {
        // le champ est valide : on réinitialise le message d'erreur
        input.setCustomValidity('');
    }
}

function getTotal(form){
    form.total.value = (form.nights.valueAsNumber * 15) +
        ((form.guests.valueAsNumber - 1) * 10)
}

```