

Développement PHP

Partie 3 : Transmettre des données

DENIS LOEUILLET – IFA - 2017

Partie 3 : Transmettre des données

- Variables superglobales
- Transmettre des données avec l'URL
- Transmettre des données avec les formulaires
- Sessions & Cookies
- Lire et écrire dans un fichier

Partie 3 : Transmettre des données

Sessions et Cookies

1. Les sessions
2. Les cookies

Partie 3 : Transmettre des données

Sessions et Cookies

1. Les sessions

- Les sessions constituent un moyen de conserver des variables sur toutes les pages de votre site.
- GET et POST permettent de transmettre des informations d'une page à une autre
 - Difficile de conserver ces informations sur toutes les pages car dès qu'on change de page les informations sont effacées
 - C'est pour cela qu'ont été inventées les sessions
 - ✓ Pour conserver des informations sur toutes les pages d'un site

Partie 3 : Transmettre des données

Sessions et Cookies : les sessions

- Fonctionnement des sessions
 - 3 étapes à connaître
 - ✓ 1^{ère} étape :
 - ❖ Un visiteur arrive sur votre site.
 - ❖ On demande à créer une session pour lui.
 - ❖ PHP génère alors un numéro unique.
 - ❖ Ce numéro est souvent très gros et écrit en hexadécimal,
 - par exemple : a02bbffc6198e6e0cc2715047bc3766f.
 - Ce numéro sert d'identifiant et est appelé « ID de session » (ou PHPSESSID).
 - PHP transmet automatiquement cet ID de page en page en utilisant généralement un cookie.

Partie 3 : Transmettre des données

Sessions et Cookies : les sessions

- Fonctionnement des sessions
 - 3 étapes à connaître
 - ✓ 2ème étape :
 - ❖ Une fois la session générée, on peut créer une infinité de variables de session pour nos besoins.
 - Par exemple, on peut créer :
 - une variable `$_SESSION['nom']` qui contient le nom du visiteur,
 - une variable `$_SESSION['prenom']` qui contient le prénom,
 - etc.
 - Le serveur conserve ces variables même lorsque la page PHP a fini d'être générée.
 - Cela veut dire que, quelle que soit la page de votre site, vous pourrez récupérer par exemple le nom et le prénom du visiteur via la superglobale `$_SESSION` !

Partie 3 : Transmettre des données

Sessions et Cookies : les sessions

- Fonctionnement des sessions
 - 3 étapes à connaître
 - ✓ 3ème étape :
 - ❖ Lorsque le visiteur se déconnecte de votre site :
 - la session est fermée et PHP « oublie » alors toutes les variables de session que vous avez créées.
 - Il est en fait difficile de savoir précisément quand un visiteur quitte votre site.
 - En effet, lorsqu'il ferme son navigateur ou va sur un autre site, le vôtre n'en est pas informé.
 - Soit le visiteur clique sur un bouton « Déconnexion » (que vous aurez créé) avant de s'en aller,
 - Soit on attend quelques minutes d'inactivité pour le déconnecter automatiquement :
 - on parle alors de timeout.
 - Le plus souvent, le visiteur est déconnecté par un timeout.

Partie 3 : Transmettre des données

Sessions et Cookies : les sessions

- Fonctionnement des sessions
 - 2 fonctions à connaître
 - ✓ `session_start()`
 - ✓ `session_destroy()`
 - `session_start ()` :
<http://php.net/manual/fr/function.session-start.php>
 - ✓ Crée une session ou restaure celle trouvée sur le serveur
 - ✓ Si le visiteur vient d'arriver sur le site, alors un numéro de session est généré pour lui.
 - ✓ Vous devez appeler cette fonction au tout début de chacune des pages où vous avez besoin des variables de session.

Partie 3 : Transmettre des données

Sessions et Cookies : les sessions

- Fonctionnement des sessions
 - 2 fonctions à connaître
 - ✓ `session_start()`
 - ✓ `session_destroy()`
 - `session_destroy ()` :
<http://php.net/manual/fr/function.session-destroy.php>
 - ✓ ferme la session du visiteur.
 - ✓ Cette fonction est automatiquement appelée lorsque le visiteur ne charge plus de page de votre site pendant plusieurs minutes (c'est le timeout),
 - ✓ Vous pouvez aussi créer une page « Déconnexion » si le visiteur souhaite se déconnecter manuellement.

Partie 3 : Transmettre des données

Sessions et Cookies : les sessions

- Fonctionnement des sessions
 - Remarque :
 - ✓ il faut appeler `session_start()` sur chacune de vos pages AVANT d'écrire le moindre code HTML
 - ✓ avant même la balise `<!DOCTYPE>` .
 - ✓ Si vous oubliez de lancer `session_start()` , vous ne pourrez pas accéder aux variables superglobales `$_SESSION` .

Partie 3 : Transmettre des données

Sessions et Cookies : les sessions

- Fonctionnement des sessions
 - Exemple d'utilisation des sessions
 - ✓ session_start.php
 - ✓ session_recup.php
 - ❖ On a créé trois variables de session qui contiennent ici le nom, le prénom et l'âge du visiteur.
 - ❖ On a fait un lien vers une autre page.
 - Ce lien est tout simples et ne transmet aucune information.
 - On ne m'occupe de rien : ni de transmettre le nom, le prénom ou l'âge du visiteur, ni de transmettre l'ID de session. PHP gère tout pour nous.
 - ❖ Maintenant, sur toutes les pages de mon site (si session_start() a été lancé), je peux utiliser les variables \$_SESSION['prenom'], \$_SESSION['nom'] et \$_SESSION['age'] !
 - Remarques :
 - ✓ On peut créer les variables de session n'importe où dans le code. La seule chose qui importe, c'est que le session_start() soit fait au tout début de la page.

Partie 3 : Transmettre des données

Sessions et Cookies : les sessions

- Fonctionnement des sessions

- Résumé :

- ✓ on peut créer des variables de session comme on crée des variables classiques :
 - ❖ à condition de les écrire dans l'array `$_SESSION`
 - ❖ et d'avoir lancé le système de sessions avec `session_start()` .
 - ✓ Ces variables sont ainsi conservées de page en page pendant toute la durée de la présence de votre visiteur.

Partie 3 : Transmettre des données

Sessions et Cookies : les sessions

- L'utilité des sessions en pratique
 - Mémoriser l'identifiant de connexion d'un visiteur
 - Puisqu'on retient son login et que la variable de session n'est créée que s'il a réussi à s'authentifier :
 - ✓ on peut l'utiliser pour restreindre certaines pages de notre site à certains visiteurs uniquement.
 - ✓ Cela permet de créer toute une zone d'administration sécurisée :
 - ❖ si la variable de session login existe, on affiche le contenu, sinon on affiche une erreur.
 - On se sert beaucoup des sessions sur les sites de vente en ligne :
 - Cela permet de gérer un « panier » :
 - ✓ on retient les produits que commande le client quelle que soit la page où il est.
 - ✓ Lorsqu'il valide sa commande, on récupère ces informations et... on le fait payer.

Partie 3 : Transmettre des données

Sessions et Cookies

2. Les cookies

- Travailler avec des cookies revient à peu près à la même chose qu'avec des sessions, à quelques différences près
 - on va voir ce qu'est exactement un cookie
 - ensuite, nous verrons comment écrire un cookie : c'est facile à faire, si on respecte quelques règles ;
 - enfin, nous verrons comment récupérer le contenu d'un cookie : ce sera le plus simple.

Partie 3 : Transmettre des données

Sessions et Cookies : les cookies

- Qu'est-ce qu'un cookie ?
 - Un cookie, c'est un petit fichier que l'on enregistre sur l'ordinateur du visiteur.
 - Ce fichier contient du texte et permet de « retenir » des informations sur le visiteur.
 - ✓ Par exemple, vous inscrivez dans un cookie le pseudo du visiteur, comme ça la prochaine fois qu'il viendra sur votre site, vous pourrez lire son pseudo en allant regarder ce que son cookie contient.
 - Les cookies ne sont pas dangereux :
 - ✓ ce ne sont pas des virus
 - ✓ ce sont des petits fichiers texte qui permettent de retenir des informations
 - ✓ au pire, un site marchand peut retenir que vous aimez les appareils photos numériques et vous afficher uniquement des pubs pour des appareils photos, mais c'est tout.
 - Chaque cookie stocke généralement une information à la fois.
 - ✓ Si vous voulez stocker le pseudonyme du visiteur et sa date de naissance, il est donc recommandé de créer deux cookies.

Partie 3 : Transmettre des données

Sessions et Cookies : les cookies

- Qu'est-ce qu'un cookie ?
 - Où sont stockés les cookies ?
 - ✓ dépend de votre navigateur web.
 - ✓ généralement on ne touche pas directement à ces fichiers, mais on peut afficher à l'intérieur du navigateur la liste des cookies qui sont stockés. On peut choisir de les supprimer à tout moment.
 - Les cookies sont classés par site web.
 - ✓ Chaque site web peut écrire, plusieurs cookies.
 - ✓ Chacun d'eux a un nom et une valeur
 - ✓ tout cookie a une date d'expiration. Après cette date, ils sont automatiquement supprimés par le navigateur.
 - Les cookies sont donc des informations temporaires que l'on stocke sur l'ordinateur des visiteurs.
 - La taille est limitée à quelques kilo-octets : vous ne pouvez pas stocker beaucoup d'informations à la fois, mais c'est en général suffisant.

Partie 3 : Transmettre des données

Sessions et Cookies : les cookies

- Écrire un cookie
 - Comme une variable, un cookie a un nom et une valeur.
 - Pour écrire un cookie, on utilise la fonction PHP « setcookie » (qui signifie « Placer un cookie » en anglais).
<http://php.net/manual/fr/function.setcookie.php>
 - ✓ On lui donne en général trois paramètres, dans l'ordre suivant :
 1. le nom du cookie (ex. : pseudo) ;
 2. la valeur du cookie (ex. : M@teo21) ;
 3. la date d'expiration du cookie, sous forme de timestamp (ex. : 1090521508).
 - timestamp : nombre de secondes écoulées depuis le 1er janvier 1970.
 - Le timestamp est une valeur qui augmente de 1 toutes les secondes.
 - Pour obtenir le timestamp actuel, on fait appel à la fonction `time()` .
 - Pour définir une date d'expiration du cookie, il faut ajouter au « moment actuel » le nombre de secondes au bout duquel il doit expirer.
 - Si vous voulez supprimer le cookie dans un an, il vous faudra donc écrire :
 - `time() + 365*24*3600` .
 - Cela veut dire : timestamp actuel + nombre de secondes dans une année.
 - Cela aura pour effet de supprimer votre cookie dans exactement un an.

Partie 3 : Transmettre des données

Sessions et Cookies : les cookies

- Écrire un cookie

- Il est recommandé d'activer l'option « httpOnly » sur le cookie :

- ✓ Lorsque ce paramètre vaut TRUE, le cookie ne sera accessible que par le protocole HTTP.
- ✓ Cela signifie que le cookie ne sera pas accessible via des langages de scripts, comme Javascript. (permet de limiter les attaques via XSS)

```
1 <?php setcookie('pseudo', 'M@teo21', time() + 365*24*3600, null, null, false, true); ?>
```

- ✓ Le dernier paramètre true permet d'activer le mode « httpOnly » sur le cookie, et donc de le rendre en quelque sorte plus sécurisé.

Partie 3 : Transmettre des données

Sessions et Cookies : les cookies

- Créer le cookie avant d'écrire du HTML
 - Comme pour « session_start », « setcookie » ne marche QUE si vous l'appellez avant tout code HTML (donc avant la balise <!DOCTYPE>)
 - Pour écrire 2 cookies il faut appeler 2 fois « setcookie »

Partie 3 : Transmettre des données

Sessions et Cookies : les cookies

- Afficher un cookie
 - Avant de commencer à travailler sur une page, PHP lit les cookies du client pour récupérer toutes les informations qu'ils contiennent.
 - Ces informations sont placées dans la superglobale `$_COOKIE`, sous forme d'array.
 - pour lire un cookie : `$_COOKIE['pseudo']`.

```
1 <p>
2     Hé ! Je me souviens de toi !<br />
3     Tu t'appelles <?php echo $_COOKIE['pseudo']; ?> et tu viens de <?php echo
   $_COOKIE['pays']; ?> c'est bien ça ?
4 </p>
```

- Si le cookie n'existe pas, la superglobale n'existe pas : utiliser `isset`
- Les cookies viennent du visiteur. Comme toute information qui vient du visiteur, elle n'est pas sûre. N'importe quel visiteur peut créer des cookies.
 - ✓ il peut les avoir modifiés, donc soyez prudents et n'ayez pas une confiance aveugle en leur contenu !

Partie 3 : Transmettre des données

Sessions et Cookies : les cookies

- Modifier un cookie existant
 - Appeler « setcookie » en gardant le même nom, ce qui écrasera l'ancien
 - Le temps d'expiration sera alors remis à zéro

Partie 3 : Transmettre des données

Sessions et Cookies : les cookies

- Résumé
 - Les variables superglobales sont des variables automatiquement créées par PHP. Elles se présentent sous la forme d'arrays contenant différents types d'informations.
 - Dans les chapitres précédents, nous avons découvert deux superglobales essentielles :
 - ✓ `$_GET` (qui contient les données issues de l'URL)
 - ✓ `$_POST` (qui contient les données issues d'un formulaire).
 - La superglobale `$_SESSION` :
 - ✓ permet de stocker des informations accessibles quel que soit le contexte pendant toute la durée de visite d'un internaute sur votre site.
 - ✓ Il faut au préalable activer les sessions en appelant la fonction `session_start()`.
 - La superglobale `$_COOKIE` :
 - ✓ représente le contenu de tous les cookies stockés par votre site sur l'ordinateur du visiteur.
 - ✓ Les cookies sont de petits fichiers que l'on peut écrire sur la machine du visiteur.
 - ✓ On crée un cookie avec la fonction `setcookie()`.