Sessions et cookies en PHP

L'ÉCOLE ATYPIQUE

Sessions et cookies en PHP

Théorie

- Les sessions et cookies sont des méthodes pour stocker des informations utilisateur et maintenir l'état entre les requêtes HTTP.
- Les sessions stockent les données côté serveur, tandis que les cookies les stockent côté client.
- session_start() est utilisé pour démarrer une session PHP, et les informations de session sont stockées dans la superglobale $_SESSION.
- Les cookies sont créés avec setcookie() et peuvent être récupérés en utilisant la superglobale $_COOKIE.

L'ÉCOLE ATYPIQUE

Sessions

Concept

- **Définition**: Une session en PHP permet de conserver des données entre plusieurs requêtes HTTP. Elle est stockée côté serveur.

- **Utilisation:** Idéale pour des données sensibles comme les informations d'authentification ou les préférences utilisateur temporaires.

Sessions

Mise en oeuvre

1. **Démarrage d'une Session** : Utilisez session_start() au début de votre script. Cela initie une nouvelle session ou reprend une existante.

2. **Stockage de Données :** Vous pouvez stocker des données dans la session en utilisant la superglobale $_SESSION, par exemple : $_SESSION['username'] = 'emmanuel';.

3. **Durée de Vie:** Par défaut, une session dure jusqu'à ce que le navigateur soit fermé. Cependant, cette durée peut être configurée dans php.ini.

Sessions

Sécurité

- **Régénération d'ID de Session :** Utilisez session_regenerate_id() lors d'une authentification pour prévenir les attaques de fixation de session.

- **Destruction de Session :** session_destroy() efface toutes les données de session, à utiliser lors de la déconnexion.

L'ÉCOLE ATYPIQUE

Cookies

Concept

- **Définition :** Les cookies sont des données stockées côté client, généralement utilisées pour le suivi des préférences utilisateur.

- **Utilisation :** Convenables pour des données non sensibles et de longue durée, comme la personnalisation de l'interface utilisateur.

L'ÉCOLE ATYPIQUE

Cookies

Mise en oeuvre

1. **Création de Cookies :** Utilisez setcookie(name, value, expire, path, domain, secure, httponly);. Par exemple, setcookie("user", "karine", time() + 3600, "/");.

   Le paramètre time() est facultatif.

   Si ce paramètre n'est pas renseigné le cookie sera supprimé à la fermeture du navigateur. Sinon le cookie est conservé le temps inscrit dans setcookie...., time() + 3600, …). Le temps est exprimé en secondes.

2. **Accès aux Cookies :** Accédez-y via la superglobale $_COOKIE, par exemple echo $_COOKIE["user"];.

L'ÉCOLE ATYPIQUE

Cookies

Sécurité

- **Durée de Vie :** Un cookie a une date d'expiration. Après cette date, il est supprimé automatiquement.

- **Attributs de Sécurité :**
  a. **Secure :**

L'attribut `Secure` garantit que le cookie n'est envoyé qu'avec des requêtes encryptées via HTTPS.

Usage: `setcookie("name", "value", [ 'secure' => true ])`.

Importance: Il prévient l'interception des cookies par des attaquants lors de la transmission entre le client et le serveur. C'est crucial pour préserver la confidentialité et l'intégrité des données dans les cookies, surtout si elles sont sensibles.

b. **HttpOnly:**

L'attribut `HttpOnly` empêche l'accès aux cookies par des scripts côté client, comme JavaScript.

Usage: `setcookie("name", "value", [ 'httponly' => true ])`.

Importance: Il réduit les risques d'attaques de type Cross-Site Scripting (XSS), où un attaquant pourrait tenter d'accéder aux cookies pour voler des informations d'authentification ou d'autres données sensibles.

Cookies

Conformité

- **Conformité RGPD :** Assurez-vous d'obtenir le consentement de l'utilisateur avant de définir des cookies, surtout pour le tracking.

L'ÉCOLE ATYPIQUE

Cookies ou sessions ?

Sessions

Sessions

Quand les utiliser ?

- **Données sensibles**: Idéal pour stocker des informations sensibles, comme les détails d'authentification, car elles sont stockées côté serveur.
- **Données temporaires**: Utilisez les sessions pour des données qui ne doivent être conservées que le temps d'une session utilisateur active.

Avantages

- **Sécurité**: Plus sécurisées que les cookies car les données sont stockées sur le serveur.
- **Pas de limitation de taille**: Contrairement aux cookies, les sessions PHP ne sont pas limitées en taille.

Préoccupations

- **Charge serveur**: Les sessions peuvent augmenter la charge sur le serveur, car les données sont stockées et gérées côté serveur.
- **Dépendance du navigateur**: Les sessions dépendent des cookies (pour l'ID de session) ou d'autres mécanismes pour identifier les utilisateurs.

L'ÉCOLE ATYPIQUE

Cookies ou sessions ?

Cookies

Cookies

Quand les utiliser ?

- **Données non sensibles**: Pour stocker des préférences utilisateur ou des données non sensibles.
- **Persistance sur le long terme**: Les cookies sont utiles pour retenir des informations entre différents usages, même après la fermeture du navigateur.
- **Indépendance du serveur**: Utilisez des cookies si vous souhaitez éviter une charge serveur supplémentaire due à la gestion des sessions.

Avantages

- **Durabilité**: Ils persistent même après la fermeture du navigateur et peuvent être configurés pour une longue durée.
- **Facilité d'accès côté client**: Les cookies peuvent être lus et écrits facilement via JavaScript, ce qui est utile pour des cas comme la personnalisation côté client.

Préoccupations

- **Sécurité et vie privée**: Les cookies sont moins sécurisés et peuvent être vulnérables à des attaques si les mesures de sécurité ne sont pas correctement implémentées.
- **Limitation de taille**: Les cookies sont limités en taille (environ 4KB par cookie).

L'ÉCOLE ATYPIQUE

Sessions et cookies en PHP

Exemples

Utilisation d'une session :

```php
<?php
session_start();
$_SESSION["username"] = "UtilisateurWebImpulse";
echo "Nom d'utilisateur enregistré dans la session : " .
$_SESSION["username"];
?>
```

Création et récupération des cookies :

```php
<?php
// Création d'un cookie
setcookie("user", "UtilisateurWebImpulse", time() + 3600, "/"); // expire après 1 heure

// Récupération d'un cookie
if(isset($_COOKIE["user"])) {
   echo "Utilisateur : " . $_COOKIE["user"];
} else {
   echo "Cookie 'user' non défini.";
}
?>
```

L'ÉCOLE ATYPIQUE

Sessions et cookies en PHP

Exercices

1. **Système de login :**

   Créez un formulaire de login simple. Utilisez les sessions pour conserver les informations de l'utilisateur à travers les pages.

2. **Préférences utilisateur :**

   Développez une page web qui permet à l'utilisateur de choisir un thème de couleur. Utilisez des cookies pour sauvegarder ce choix et l'appliquer lors des visites ultérieures.