

Cookies PHP

Dans ce chapitre, vous apprendrez à stocker une petite quantité d'informations dans le navigateur de l'utilisateur lui-même à l'aide des cookies PHP.

Qu'est-ce qu'un cookie

Un cookie est un petit fichier texte qui permet de stocker une petite quantité de données (près de 4 Ko) sur l'ordinateur de l'utilisateur. Ils sont généralement utilisés pour garder une trace d'informations telles que le nom d'utilisateur que le site peut récupérer pour personnaliser la page lors de la prochaine visite de l'utilisateur sur le site Web.

Conseil : Chaque fois que le navigateur demande une page au serveur, toutes les données du cookie sont automatiquement envoyées au serveur dans la requête.

Paramétrer un cookie en PHP

La fonction **setcookie()** est utilisée pour définir un cookie en PHP. Assurez-vous d'appeler la fonction **setcookie()** avant toute sortie générée par votre script, sinon le cookie ne sera pas défini. La syntaxe de base de cette fonction peut être donnée par :

```
setcookie(name, value, expire, path, domain, secure);
```

Les paramètres de la fonction setcookie() ont les significations suivantes :

Paramètre	La description
name	Le nom du cookie.
value	La valeur du cookie. Ne stockez pas d'informations sensibles car cette valeur est stockée sur l'ordinateur de l'utilisateur.
expires	La date d'expiration au format d'horodatage UNIX. Passé ce délai, le cookie deviendra inaccessible. La valeur par défaut est 0.
path	Spécifiez le chemin sur le serveur pour lequel le cookie sera disponible. S'il est défini sur /, le cookie sera disponible dans l'ensemble du domaine.
domain	Spécifiez le domaine pour lequel le cookie est disponible, par exemple www.example.com.
secure	Ce champ, s'il est présent, indique que le cookie ne doit être envoyé que si une connexion HTTPS sécurisée existe.

Conseil : Si le délai d'expiration du cookie est défini sur 0 ou omis, le cookie expirera à la fin de la session, c'est-à-dire à la fermeture du navigateur.

Voici un exemple qui utilise la fonction **setcookie()** pour créer un cookie nommé *username* et lui attribuer la valeur *value John Carter*. Il précise également que le cookie expirera après 30 jours (*30 days * 24 hours * 60 min * 60 sec*).

```
<?php
// Création de cookie
setcookie("username", "John Carter", time()+30*24*60*60);
?>
```

Remarque : Tous les arguments, à l'exception du nom, sont facultatifs. Vous pouvez également remplacer un argument par une chaîne vide (""), afin d'ignorer cet argument, cependant pour ignorer l'argument d'expiration, utilisez un zéro (0) à la place, car il s'agit d'un entier.

Avertissement : Ne stockez pas de données sensibles dans les cookies car elles pourraient potentiellement être manipulées par l'utilisateur malveillant. Pour stocker les données sensibles en toute sécurité, utilisez plutôt des *sessions* .

Accéder aux valeurs des cookies

La variable superglobale PHP `$_COOKIE` est utilisée pour récupérer une valeur de cookie. Il s'agit généralement d'un tableau associatif qui contient une liste de toutes les valeurs de cookies envoyées par le navigateur dans la requête en cours, classées par nom de cookie. La valeur de cookie individuelle est accessible à l'aide de la notation de tableau standard. Par exemple, pour afficher le cookie de nom d'utilisateur défini dans l'exemple précédent, vous pouvez utiliser le code suivant.

```
<?php
// Accès à la valeur d'un cookie individuel
echo $_COOKIE["username"];
?>
```

Le code PHP de l'exemple ci-dessus produit la sortie suivante.

John Carter

Il est recommandé de vérifier si un cookie est défini ou non avant d'accéder à sa valeur. Pour ce faire, vous pouvez utiliser la fonction PHP `isset()`, comme ceci :

```
<?php
// Vérifier si un cookie est défini ou non
if(isset($_COOKIE["username"])){
    echo "Bonjour " . $_COOKIE["username"];
} else{
    echo "Bienvenue à l'invité!";
}
?>
```

Suppression des cookies

Vous pouvez supprimer un cookie en appelant la même fonction **setcookie()** avec le nom du cookie et n'importe quelle valeur (telle qu'une chaîne vide), mais cette fois, vous devez définir la date d'expiration dans le passé, comme indiqué dans l'exemple ci-dessous :

```
<?php
// Supprimer le cookie
setcookie("username", "", time()-3600);
?>
```

Conseil : vous devez transmettre exactement le même chemin, domaine et autres arguments que vous avez utilisés lors de la première création du cookie afin de vous assurer que le bon cookie est supprimé.