**GESTION DES SESSIONS**

***Fonctionnement session\_start:***

on créé une page header.php dans laquelle il y a une fonction session\_start() et qu’on va inclure à l’aide de *include ou require .*

le session\_start génère un identifiant de session unique qui va être envoyé au navigateur sous forme de cookie c'est **PHPSESSID**.

***Voici comment recup l’id de session :***

<?php

//On démarre une nouvelle session

**session\_start();**

/\*On utilise session\_id() pour récupérer l'id de session s'il existe.

\*Si l'id de session n'existe pas, session\_id() rnevoie une chaine

\*de caractères vide\*/

**$id\_session = session\_id();**

?>

<!DOCTYPE html>

<html>

<head>

<title>Cours PHP & MySQL</title>

<meta charset="utf-8">

<link rel="stylesheet" href="cours.css">

</head>

<body>

<h1>Titre principal</h1>

<?php

if($id\_session){

echo 'ID de session (récupéré via session\_id()) : <br>'

.**$id\_session**. '<br>';

}

echo '<br><br>';

if(isset($\_COOKIE['PHPSESSID'])){

echo 'ID de session (récupéré via $\_COOKIE) : <br>'

.**$\_COOKIE['PHPSESSID']**;

}

?>

<p>Un paragraphe</p>

</body>

</html>

***Utiliser les infos de session :***

utiliser la variable superglobale $\_SESSION qui est un **tableau associatif** dans lequel on peut définir ce qu on veut mettre dans les infos de session et qu’on « trimbale » de page en page.

<?php

//On démarre une nouvelle session

session\_start();

//On définit des variables de session

$\_SESSION['prenom'] = 'Marie';

$\_SESSION['age'] = 49;

?>

***On l utilise de cette façon :***

<?php

echo 'Bonjour ' .$\_SESSION['prenom']. ',

tu as ' .$\_SESSION['age']. ' ans';

?>

***Detruire une session :***

utiliser les fonctions session\_destroy() *qui détruit toutes les données associées à la session* courante et session\_unset() q*ui détruit toutes les variables d’une session.*

La fonction session\_destroy() va supprimer le fichier de session dans lequel sont stockées toutes les informations de session. Cependant, **cette fonction ne détruit pas les variables globales associées à la session (c’est-à-dire le contenu du tableau $\_SESSION) ni le cookie de session**.

le cookie PHPSESSID pourra être effacé en utilisant setcookie() en définissant une date d’expiration passée pour le cookie.

Il va cependant être très rare d’avoir besoin de détruire les données associées à une session et donc d’appeler session\_destroy(). On préférera généralement modifier le tableau $\_SESSION manuellement pour supprimer des données en particulier.

utiliser la fonction session\_unset() (sans lui passer d’argument) pour détruire toutes les variables de la session courante. Cette fonction va également nous permettre de détruire une variable de session en particulier en lui passant sa valeur de la manière suivante : unset($\_SESSION['nom-de-la-variable-de-session-a-detruire']).

Ci dessous malgré le destroy, le prénom apparaît encore alors que avec le unset en plus, il détruit les données lié à [age]

<?php

/\*Si la variable de session age est définie, on echo sa valeur

\*puis on la détruit avec unset()\*/

if(isset($\_SESSION['age'])){

echo 'Tu as ' .$\_SESSION['age']. ' ans<br>';

**unset($\_SESSION['age']);**

}

/\*On détruit les données de session\*/

**session\_destroy();**

//On tente d'afficher les valeurs des variables age et prenom

echo 'Contenu de $\_SESSION[\'age\'] : ' .$\_SESSION['age']. '<br>';

echo 'Contenu de $\_SESSION[\'prenom\'] : ' .$\_SESSION['prenom'];

?>

session\_destroy() ne va pas détruire les variables globales de session. Cependant, comme les informations de session sont détruites, les variables de session *ne vont plus être accessibles que dans le script courant*.

<?php

// On démarre la session

session\_start ();

// On détruit les variables de notre session

[session\_unset](https://www.lephpfacile.com/manuel-php/function.session-unset.php) ();

// On détruit notre session

[session\_destroy](https://www.lephpfacile.com/manuel-php/function.session-destroy.php) ();

// On redirige le visiteur vers la page d'accueil

[header](https://www.lephpfacile.com/manuel-php/function.header.php) ('location: index.htm');

?>

Quelques fonctions liés à la session

|  |  |
| --- | --- |
| Fonction | Signification |
| session\_start | Démarre une session |
| session\_register | Enregistre une variable de session |
| session\_unregister | Efface une variable de session |
| session\_is\_registered | Vérifie si une variable est déclarée pour la session en cours |
| session\_id | Retourne l'id de la session en cours |
| session\_name | Retourne le nom de la session en cours |
| session\_unset | Detruit toutes les variables de la session en cours |
| session\_destroy | Destruit la session en cours |

## session\_regenerate\_id()?

Comme le nom de la fonction l'indique, il s'agit d'une fonction qui remplacera l'ID de session actuel par un nouveau et conservera les informations de session en cours.

Il est utilisé après session\_start

## Qu'est ce que ça fait?

Il aide principalement à prévenir les attaques de fixation de session. Les attaques de fixation de session sont lorsqu'un utilisateur malveillant tente d'exploiter la vulnérabilité d'un système pour fixer (définir) l'ID de session (SID) d'un autre utilisateur.