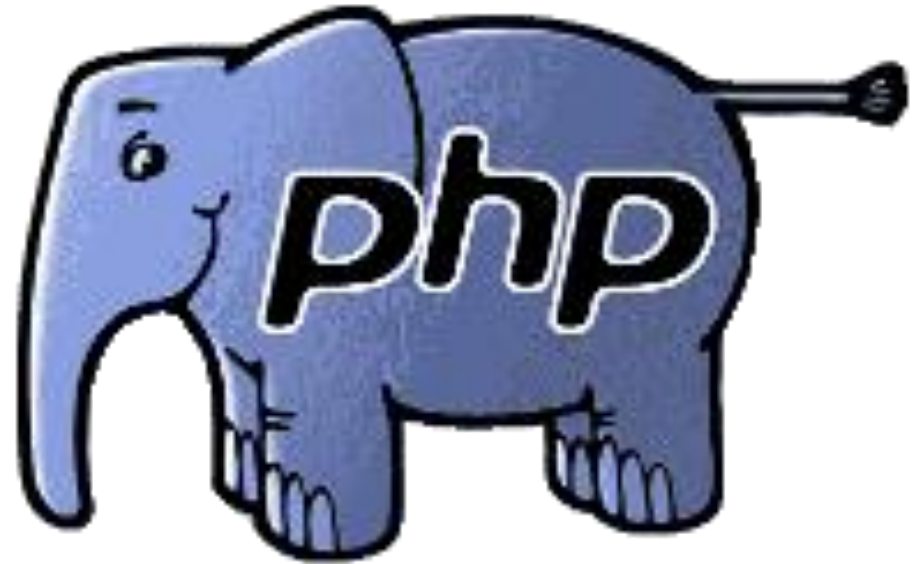


PHP : Variable GET Session Cookie



Connaissances pré-requises

- Formulaires HTML - PHP
- PHP : Les bases

Variable GET via l'URL

- [Concept et exemple](#)

Session

- [Concept](#)
- [Exemple](#)

Cookie

- [Utilité](#)
- [Création](#)
- [Limites](#)
- [Exemple](#)

PHP - Fonctionnalités utilisées dans ce document

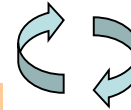
* Manipulation de **variables** (locale, globale, constantes).

\$a

* **Opérations** avec les variables, caractères, dates.

+ - !

* Les **boucles** (if, else, ternaire, switch, for, while).



* La factorisation et les **fonctions**.

f()

* Les **tableaux** de variables.



* Le passage de **paramètres** (URL) à une autre page web, formulaires (get, post).

* Manipulation des **répertoires et fichiers** sur le serveur.

* Fabrication de code



* Gestion des **sessions** et **cookies** (sur le poste utilisateur).

* PHP et Base de donnée.



* **Paramétrage** , **Sécurité** (formulaires).



* Programmation Orientée Objets (**POO**)

Passage de variable GET via l'URL

Le passage de variable GET via l'URL permet de transférer des informations de petit volume d'une page à une autre.

Ces informations sont stockées dans un tableau de variables globales **\$_GET**. (Le même tableau que celui utilisé par les formulaires de type GET).

Page de départ

```
index.php
1 <!DOCTYPE HTML><html lang="fr"><!-- fichier.cnam/nfa017/20190403/index.php.PBO -->
2 <head><meta charset="utf-8" />
3 <title>Passage de variable via l'URL</title>
4 </head>
5 <html><body>
6 <h1>Passage de variable via l'URL</h1>
7
8
9 <?php $variable = 'tutu'; /* contenu de la variable à transférer à l'autre page */?>
10
11
12 <a href="reception-donnee.php?donnee=<?php echo $variable; ?>">tranfert donnée</a>
13 <!-- lien intégrant la variable en paramètre dans l'url -->
14
15
16 </body></html>
17
```

Page d'arrivée

```
index.php reception-donnee.php
1 <!DOCTYPE HTML><html lang="fr">
2 <!-- fichier.cnam/nfa017/20190403/reception-donnee.php.PBO -->
3 <head><meta charset="utf-8" />
4 <title>reception-donnee.php</title>
5 </head>
6 <html><body>
7 <h1>reception-donnee.php</h1>
8
9 <p>donnee reçue : <?php echo $_GET['donnee']; ?></p>
10 <!-- La donnée reçue est lisible dans l'url -->
11
12 </body></html>
13
```

Session - Le concept

Les Sessions permettent de garder des variables disponibles en mémoire **coté client** tant que le navigateur n'est pas fermé (ou que la variable n'est pas supprimée). Ces variables sont stockées dans un tableau de variables globales **\$_SESSION**. Les variables de ce tableau sont accessibles dans toutes les pages PHP commençant par l'instruction **session_start()**;

Partie PHP : Exemple

```
<?php  
session_start();
```

```
$_SESSION['recette'] = 'Compote';
```

```
echo $_SESSION['recette'] ;
```

```
session_write_close() ;
```

```
unset($_SESSION);
```

```
?>
```

Explications

Permet **l'accès** aux éventuelles variables de session.

Initialisation d'une variable de session.

Affichage de la valeur de la variable de session créée.

Ferme la session en cours.

Suppression de l'ensemble des variables de session en mémoire.

Remarque : Plus d'informations sur <https://www.julp.fr/articles/17-php-cookies-et-sessions.html>



P.Bouquet

maj du 05/04/2019

Session - Exemple

Les Sessions - Exemple de code

```
index.php x session.php x session.inc.php x
1 <?php /* fichier cnam/nfa017/session.php -- 20160320-PBO */
2 session_start();
3 include("inc/header.inc.php"); headerinclude("session");
4
5 /* variables ===== */
6 $_SESSION['recette_nom'] = "Soupe de légume";
7
8 /* affichage ===== */ ?>
9 <p>J'envoie en session "Soupe de légume" => recette_nom</p>
10
11 <p><a href="index.php">retour vers index.php</a></p>
12
13 </body></html>
14
```

```
index.php x session.php x session.inc.php x
1 <!-- fichier cnam/nfa017/20170503/index.php -- 20160320-PBO -->
2
3 <strong>Session : </strong><?php
4
5 if(isset($_SESSION['recette_nom'])) {echo $_SESSION['recette_nom'];}
6
7 session_unset(); /* suppression des variables de session */ ?>
8
```

Remarque : Plus d'informations sur <https://www.julp.fr/articles/17-php-cookies-et-sessions.html>



P.Bouquet

maj du 05/04/2019

Les Cookies permettent de stocker un fichier sur l'ordinateur de l'internaute (**coté client**) d'une petite taille, pour une durée définie. Les principales utilisations sont :

- Sauvegarde du **design** préféré d'un site pour un internaute
- Afficher les **nouveaux messages** depuis la dernière visite de l'internaute
- Afficher les **messages non-lus** par le visiteur dans un forum
- Fragmentation d'un **formulaire sur plusieurs pages**. Les valeurs envoyées sur la première page sont sérialisées et envoyées par cookie à la seconde qui les désérialisera
- Compter les **nouvelles visites**
- Reconnaître les **visiteurs ayant déjà voté** à un sondage

Cookies - Création

Utiliser des Cookies grâce à PHP

Le contenu de ces fichier est accessible par un tableau de variables globales **\$_COOKIE** dans toutes les pages PHP.

Partie PHP : Exemple

```
<?php  
setcookie(
```

```
"recette",  
"gateau",  
time()+60
```

```
);
```

```
echo $_COOKIE ['recette'] ;
```

```
?>
```

Explications

Instruction PHP pour fabriquer le fichier (cookie)

Nom du fichier
contenu du fichier
heure où cookie n'est plus disponible :
 heure actuelle + 60 secondes
fin de l'instruction PHP

Affichage de la valeur du contenu du cookie créé.

Remarque : Plus d'informations sur <https://www.julp.fr/articles/17-php-cookies-et-sessions.html>



[P.Bouquet](#)

maj du 05/04/2019

Attention aux conditions d'utilisation des Cookies

- il ne doit pas y avoir de point dans le **nom du cookie** (plante avec certains navigateurs)
- l'instruction setcookie doit être placée **avant toute balise** <html>, <head>, <p>, ... et avant toute instruction echo() , print(), ...
- les cookies ne sont **disponibles qu'au chargement de la prochaine page** : on ne peut pas créer et lire un cookie dans une page php sans recharger une page (celle-ci ou une autre)
- les navigateurs n'acceptent en general que **20 cookies par domaine** et rejettent les autres
- le **poids max** d'un cookies est de 4 ko
- La **règlementation européenne** demande :
 - * une **acceptation de l'internaute** avant de déposer un cookie sur son appareil, même pour un cookie de statistique de visite
 - * une **durée de stockage du cookie limitée à 13 mois** maximum à partir du 1e dépôt sans prolongement automatique de la durée lors des nouvelles visites.

Remarque : Plus d'informations tech sur <https://www.julp.fr/articles/17-php-cookies-et-sessions.html>
Plus d'informations sur la réglementation sur : <https://www.cnil.fr/fr/cookies-traceurs-que-dit-la-loi>

Cookies - Exemple

```
5
6  /* variables ===== */
7  $cookienom = "recette_nom"; // nom du cookie
8  $cookiedata = "miam"; // contenu du cookie
9  $cookieexpire = 20; // durée de vie du cookies (en sec)
10
11 /* envoi cookie ===== */
12 setcookie($cookienom,$cookiedata,time()+$cookieexpire); // création
13
14 /* fin du HTML ===== */ ?>
15 <p>nom du cookie envoyé : "<?php echo $cookienom; ?>"<br/>
16 qui contient "<?php echo $cookiedata; ?>"<br/>
17 pour une durée de <?php echo $cookieexpire; ?> secondes.</p>
18
19 <a href="index.php">Retour Index</a>
20
21 </body>
22 </html>
23
```

```
index.php x cookie.inc.php x cookie.php x
1 <!-- fichier cnam/nfa017/inc/cookie.inc.php -- 20170503-PBO -->
2
3 <strong>Cookie : </strong><?php
4 if(isset($_COOKIE['recette_nom'])) {echo $_COOKIE['recette_nom'];} ?>
5
```