



## Partie 2: transmettre les données

# Partie 2 : transmettre les données



1. Transmettre les données par l'URL
2. Transmettre les données avec un formulaire
3. Les types de champs de formulaire
4. Variables de session
5. Les Cookies

# Partie 2 / Transmettre un paramètre d'URL

<https://www.google.fr/search?q=php>

? = indique qu'à partir de ce caractère l'URL contient des variables

q = nom de la variable

php = Valeur de la variable

Plusieurs paramètres dans l'URL :

<https://www.google.fr/search?q=php&ie=utf-8>

& = Sépare chaque groupe de variable = valeur

# Partie 2 / Lire un paramètre d'URL



Pour afficher la valeur de la variable `q` dans l'url  
`https://www.google.fr/search?q=php`

```
echo $_GET['q'];
```

# Partie 2 / Transmettre des données par un formulaire

A screenshot of a 'Sign In' dialog box. The dialog has a title bar with 'Sign In' and a close button (X). It contains two input fields: the first has a user icon and the text 'icondeposit|'; the second has a lock icon and a masked password '\*\*\*\*\*'. Below the fields is a link 'Forgot Username or Password?'. At the bottom is a green 'Log In' button.

Sign In X

icondeposit|

\*\*\*\*\*

[Forgot Username or Password?](#)

Log In

# Partie 2 / Transmettre des données par un formulaire



```
1 <html>
2 <head>
3   <title></title>
4 </head>
5 <body>
6   <form method="get" action="traitement.php">
7     <p> Mes éléments de formulaire</p>
8   </form>
9 </body>
10 </html>
```

attribut **method** : **get** ou **post**

**get** : les données transitent en clair par l'URL. Limité à 256 caractères.

**post** : les données ne transitent pas en clair dans l'url. La quantité de données n'est pas limitée

attribut **action** :

Il indique la page de destination, celle qui recevra les données

## Partie 2 / Transmettre des données par un formulaire



```
<form method="get" action="traitement.php">
  <p>Merci de vous identifier</p>
  <input type="text" name="identifiant" value="" />
  <input type="password" name="password" value="" />
  <input type="submit" name="envoyer" value="connexion"/>
</form>
```

l'attribut **type** détermine que type de champ sera produit :

- balise `input type="text"` : champ de saisie texte
- balise `input type="password"` : champ de saisie texte, le texte saisi est masqué
- balise `input type="submit"` : crée un bouton pour envoyer le formulaire

l'attribut **name** donne le nom de la variable qui sera réceptionné

l'attribut **value** donne une valeur par défaut à notre champ de formulaire

## Partie 2 / Lire les données envoyées par un formulaire



```
<body>  
    <p>Bonjour <?php echo $_POST['identifiant']?></p>  
</body>
```

Pour lire le contenu d'une variable POST en PHP il suffit de l'appeler ainsi :

```
$_POST['nomDeMonChamp'];
```



# Partie 2 / Les autres types de champ



## Liste déroulante

```
<select name="listederoulante">
  <option value="valeur0">Texte Affiché</option>
  <option value="valeur1">Texte Affiché</option>
  <option value="valeur2" selected="selected">Texte Affiché</option>
</select>
```

## Case à cocher



```
<input type="checkbox" name="caseAChochoer" value="valeurEnvoyee" checked="checked"/>
```

# Partie 2 / Les autres types de champ



## Boutons d'option

Etes vous un homme ou une femme

- ☐ Femme  
☒ Homme

```
<p>Etes vous un homme ou une femme</p>  
<input type="radio" name="genre" value="femme" />Femme <br/>  
<input type="radio" name="genre" value="homme" checked="checked" />Homme <br/>
```

## Zones de texte

```
<textarea name="adresse">  
    Tout ce qui est la dedans sera transmis même si le texte est tres long  
</textarea>
```

## Champ caché

```
<input type="hidden" value="valeurTransMise" />
```

## Partie 2 / TP : Page protégée par un mot de passe



Je souhaite limiter l'accès à mon cv qu'a un groupe restreint de personnes.

Pour cela je souhaite que l'internaute tape le mot de passe chocolat pour lire le contenu de ma page.

1. Créer un formulaire de saisie du mot de passe
2. Si mot de passe ok, j'affiche la page
3. Si mot de passe pas ok je propose un lien vers la page précédente pour re-essayer.

BONUS : pour ceux qui se sentent à l'aise faire l'ensemble sur une seule page.

## Partie 2 / TP : Consulter les données d'un tableau



Je dispose d'une base de données des élèves sous forme d'un tableau, je souhaite :

afficher la liste de tous les élèves nom / prénom

au clic sur un nom : afficher toutes les informations de l'élève sélectionné : (nom / prénom / email / date de naissance).

BONUS : la liste comporte une case à cocher devant chaque élève.

Lorsque que l'on clique sur le bouton rechercher, on affiche le détail de tous les élèves cochés.

# Partie 2 : les variables de session



Les sessions servent à conserver des variables entre toutes les pages d'un site.

- **Écriture** d'une variable de session : `$_SESSION['nom'] = 'TOTO';`
- **Lecture** d'une variable de session : `echo $_SESSION['nom'];`

Dans cet exemple la variable `$_SESSION['nom']` affichera TOTO sur toutes les pages du site.

2 Fonction pour utiliser les sessions :

- **session\_start()** : doit être utilisé en tête de page, dans chacune des page ou l'on voudra lire ou écrire une variable de session.
- **session\_destroy** : sert à détruire les variables de session. exemple une page de déconnexion.

# Partie 2 : les variables de session



SYSTEME  
DE  
SESSION

`$_SESSION['username']`

`$_SESSION['motdepasse']`

Username

username

Password

••••••••••

Log In

login.php

Bonjour Xavier Salaberria  
Vous êtes authentifié

nimportequellepage.php

# Partie 2 / TP : Page protégée par un mot de passe



Je viens de découvrir les sessions PHP, je souhaite créer un système d'authentification sur mon site :

1. Ma page d'identification demandera un identifiant et un mot de passe
2. Si l'identification est ok, je peux accéder à toutes les page du site et afficher mon login
3. Ma page d'identification comportera un lien permettant de me déconnecter

BONUS : pour ceux qui se sentent à l'aise :

Si je consulte suis la page d'identification alors que je suis déjà authentifié, je serai redirigé vers la page par défaut de mon site

# Partie 2 : les cookies



Les cookies fonctionnent un peu comme les sessions, ils gardent une information entre plusieurs pages. L'information est stockée dans le navigateur et non sur le serveur. La validité du cookie est programmable.

**Écriture d'un cookie** : `setcookie('nomcookie','valeur',1433103868);`

- `nomcookie` : nom de la variable
- `valeur` : valeur de la variable
- `1433103868` : expiration timestamp

**Timestamp** = nombre de secondes écoulées depuis le 1er janvier 1970, le 31/5/2015 à 22:24:28 la valeur du timestamp était : 1433103868

**Pour écrire un cookie qui expire dans 24h** : `setcookie('nomcookie','valeur',time()+(3600*24));`



# Partie 2 : les cookies



## Sécuriser son cookie avec le mode httpOnly :

Sans entrer dans les détails, il existe certaines failles de sécurité liés a l'utilisation des cookies. Pour les éviter il faut utiliser le mode httpOnly, cela interdit la lecture d'un cookie par du code javascript.

## Écriture d'un cookie avec le mode httpOnly:

```
setcookie('nomcookie','valeur',time()+(3600*24),null, null, false, true);
```

**true** : active le mode httpOnly

# Partie 2 : les cookies



## Lire un cookie :

```
$_COOKIE['nomcookie'];
```

## Modifier un cookie :

```
setcookie('nomcookie','nouvellevaleur',time()+3600*24,null, null, false, true);
```

## Supprimer un cookie :

```
setcookie('nomcookie','nouvellevaleur',time(),null, null, false, true);
```

# Partie 2 / TP : Page protégée par un mot de passe



Je viens de découvrir les sessions PHP, je souhaite créer un système d'authentification sur mon site :

1. Ma page d'identification demandera un identifiant et un mot de passe et une case à cocher se souvenir de moi. Permettant d'enregistrer mon cookie pendant 30 Jours
2. Si l'identification est ok, je peux accéder à toutes les pages du site et afficher mon login
3. Ma page d'identification comportera un lien permettant de me déconnecter