

PHP 2

D'après openclassrooms

Alexandre Benoit

TS10

La semaine dernière, on a vu :

- Les bases du PHP
- Comment faire des algorithmes avec PHP
- Comment transmettre des données avec des URL
- On n'a pas pu tester à cause d'un problème technique

La semaine dernière, on a vu :

- Les bases du PHP
- Comment faire des algorithmes avec PHP
- Comment transmettre des données avec des URL
- On n'a pas pu tester à cause d'un problème technique

Aujourd'hui, on va voir :

- Comment envoyer des données de façon plus efficace
- Les variables de sessions et un peu les cookies
- Tester avec du code avec un petit TP

I Les formulaires

On crée un formulaire avec le code suivant :

```
<form method="post" action="cible.php">
```

```
<p>
```

```
    On inserera ici les elements de notre formulaire.
```

```
</p>
```

```
</form>
```

On crée un formulaire avec le code suivant :

```
<form method="post" action="cible.php">  
  
<p>  
    On inserera ici les elements de notre formulaire.  
</p>  
  
</form>
```

Ce formulaire enverra des informations à la page `cible.php` (on aurait pu l'appeler autrement).

On crée un formulaire avec le code suivant :

```
<form method="post" action="cible.php">  
  
<p>  
    On inserera ici les elements de notre formulaire.  
</p>  
  
</form>
```

Ce formulaire enverra des informations à la page `cible.php` (on aurait pu l'appeler autrement).

Si on avait utiliser la méthode `get`, on aurait envoyé les informations par la barre d'adresse.

Les éléments du formulaire : input

La balise INPUT est la balise essentielle des formulaires, car elle permet de créer un bon nombre d'éléments **interactifs**. La syntaxe de cette balise est la suivante :

```
<input type="text" name="prenom" value="toto" />
```


Les éléments du formulaire : input

La balise INPUT est la balise essentielle des formulaires, car elle permet de créer un bon nombre d'éléments **interactifs**. La syntaxe de cette balise est la suivante :

```
<input type="text" name="prenom" value="toto" />
```

On peut aussi utiliser d'autre type comme :

- submit

Les éléments du formulaire : input

La balise INPUT est la balise essentielle des formulaires, car elle permet de créer un bon nombre d'éléments **interactifs**. La syntaxe de cette balise est la suivante :

```
<input type="text" name="prenom" value="toto" />
```

On peut aussi utiliser d'autre type comme :

- submit
- checkbox

Les éléments du formulaire : input

La balise INPUT est la balise essentielle des formulaires, car elle permet de créer un bon nombre d'éléments **interactifs**. La syntaxe de cette balise est la suivante :

```
<input type="text" name="prenom" value="toto" />
```

On peut aussi utiliser d'autre type comme :

- submit
- checkbox
- radio

Les éléments du formulaire : input

La balise INPUT est la balise essentielle des formulaires, car elle permet de créer un bon nombre d'éléments **interactifs**. La syntaxe de cette balise est la suivante :

```
<input type="text" name="prenom" value="toto" />
```

On peut aussi utiliser d'autre type comme :

- submit
- checkbox
- radio
- hidden

Select permet par exemple d'avoir des menus déroulant. Par exemple :

```
<select name="choix">
  <option value="choix1">Choix 1</option>
  <option value="choix2">Choix 2</option>
  <option value="choix3">Choix 3</option>
  <option value="choix4">Choix 4</option>
</select>
```

Select permet par exemple d'avoir des menus déroulant. Par exemple :

```
<select name="choix">
  <option value="choix1">Choix 1</option>
  <option value="choix2">Choix 2</option>
  <option value="choix3">Choix 3</option>
  <option value="choix4">Choix 4</option>
</select>
```

On peut aussi rajouter l'option **multiple** qui permet de faire plusieurs choix.

La balise TEXTAREA permet de définir une zone de saisie plus vaste par rapport à la simple ligne de saisie que propose la balise INPUT. Par exemple :

```
<textarea name="message" rows="8" cols="45">
  Votre message ici.
</textarea>
```

II Les sessions

Les sessions constituent un moyen de conserver des variables sur toutes les pages d'un site.

Les sessions constituent un moyen de conserver des variables sur toutes les pages d'un site.

Pour les utiliser, on doit connaître deux fonctions.

- `session_start()` : démarre le système de sessions. Si le visiteur vient d'arriver sur le site, alors un numéro de session est généré pour lui. Vous devez appeler cette fonction au tout début de chacune des pages où vous avez besoin des variables de session.

Les sessions constituent un moyen de conserver des variables sur toutes les pages d'un site.

Pour les utiliser, on doit connaître deux fonctions.

- `session_start()` : démarre le système de sessions. Si le visiteur vient d'arriver sur le site, alors un numéro de session est généré pour lui. Vous devez appeler cette fonction au tout début de chacune des pages où vous avez besoin des variables de session.
- `session_destroy()` : ferme la session du visiteur. Cette fonction est automatiquement appelée lorsque le visiteur ne charge plus de page de votre site pendant plusieurs minutes (c'est le timeout), mais vous pouvez aussi créer une page « Déconnexion » si le visiteur souhaite se déconnecter manuellement.

Voici un exemple d'utilisation :

```
<?php
// On démarre la session AVANT d'écrire du code HTML
session_start();
?>
<!DOCTYPE html>
<html>
  <head>
    <!-- En-tête de la page -->
    <meta charset="utf-8" />
    <title>Titre</title>
  </head>
  <body>
    <?php $_SESSION['prenom'] = 'toto'; ?>
    <p> <?php echo $_SESSION['prenom']; ?> </p>
```

III TP : Mot de passe

On va écrire quelques page web accessible par un mot de passe.

- On demandera dans un premier temps à l'utilisateur d'entrer un mot de passe sur une première page.
- Le mot de passe sera envoyé vers une autre page qui nous dira succès ou échec.
- De cette autre page, on pourra accéder à d'autres pages qu'à partir de ce mot de passe. Si on essaie d'accéder à ces pages sans mot de passe, on aura une erreur.

IV Les cookies

Un **cookie** est un petit fichier que l'on enregistre sur l'ordinateur du visiteur.

Un **cookie** est un petit fichier que l'on enregistre sur l'ordinateur du visiteur. Ce fichier contient du texte et permet de « retenir » des informations sur le visiteur. Par exemple, on inscrit dans un cookie le pseudo du visiteur, comme ça la prochaine fois qu'il viendra sur le site, on pourra lire son pseudo en allant regarder ce que son cookie contient.

Un **cookie** est un petit fichier que l'on enregistre sur l'ordinateur du visiteur. Ce fichier contient du texte et permet de « retenir » des informations sur le visiteur. Par exemple, on inscrit dans un cookie le pseudo du visiteur, comme ça la prochaine fois qu'il viendra sur le site, on pourra lire son pseudo en allant regarder ce que son cookie contient. Pour écrire un **cookie**, il suffit d'entrer ceci

```
<?php setcookie( 'pseudo' , 'Homer' , time() +  
365*24*3600); ?>
```