

## Plan

- Chapitre 1 : Introduction
- Chapitre 2 : Notions de base
- Chapitre 3 : La gestion des fichiers avec PHP
- **Chapitre 4 : La gestion des formulaires**
- Chapitre 5: Les cookies et les sessions
- Chapitre 6 : POO avec PHP
- Chapitre 7 : Interfaçage avec une base de données
- Chapitre 8 : Frameworks

## La gestion des formulaires

### 👉 Formulaire HTML

- Retourne des informations saisies par un utilisateur vers une application serveur
- La création d'un formulaire nécessite la connaissance de quelques balises HTML indispensables :

- Structure : un formulaire commence toujours par la balise

`<form>` et se termine par la balise `</form>`

- Champ de saisie de text en ligne :

`<input type = "text" name ="nom_du_champ" value="chaîne">`

## La gestion des formulaires

### 👉 Formulaire HTML

- Boutons d'envoi et d'effacement :

`<input type="submit" value = "Envoyer">`

`<input type = "reset" name ="efface" value = "Effacer">`

- Case à cocher et bouton radio :

`<input type = "checkbox" name ="case1" value="valeur_case">`

`<input type = "radio" name ="radio1" value ="valeur_radio">`

## La gestion des formulaires

### 👉 Envoi de données

- Le formulaire doit spécifier le nom du fichier PHP et la méthode d'envoi

`<form action=« nomfichier.php" method= « POST/GET">`

- Le formulaire envoie les données au programme PHP décrit par le paramètre « action » de la balise « form », est invoqué lors du clic sur le bouton « submit »

## La gestion des formulaires

### 👉 Envoi de données

#### ■ Méthodes d'envoi get et post:

- La méthode GET place les informations d'un formulaire directement à la suite de l'adresse URL de la page appelée.

Exemple : <http://www.site.com/cible.php?champ=valeur&champ2=valeur>

- Inconvénients de la méthode GET:

- Rendre visibles les données dans la barre d'adresse du navigateur.
- De plus, la longueur totale est limitée à 255 caractères, ce qui rend impossible la transmission d'un volume de données important

- La méthode POST regroupe les informations dans l'entête d'une requête HTTP

- Assure une confidentialité efficace des données

## La gestion des formulaires

### 👉 Récupération des données

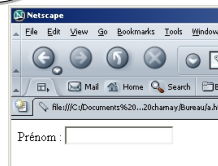
- Le programme PHP peut récupérer les valeurs saisies dans les différents champs du formulaire pour les traiter
- C'est le nom, au sens HTML, défini par le paramètre « name », qui va être utilisé pour créer la variable PHP

## La gestion des formulaires

### 👉 Récupération des données

```
<html>
<form
  name=xyz method=GET
  action=test.php>
Préacute;nom :
<input name=prenom>
</form>
</html>
```

```
<?
echo "Bonjour" . $ GET['prenom'] ;
?>
```



Cette méthode n'est possible que si `register_globals = on` dans le fichier de configuration de l'interpréteur PHP (`php.ini`)

## La gestion des formulaires

### 👉 Exemple

Fichier HTML

```
Texte
Option_1 ▲
Option_2 ▲
Option_3 ▼
Texte par défaut

 Case 1
 Case 2
 Case 3
 radio 1
 radio 2
 radio 3
Annuler Soumettre
```

## La gestion des formulaires

### Exemple

Fichier HTML

```
<form method="GET" action="page_cible.php">
<input type="text" name="Champ_saisie" value="Texte" > <BR>
<select name="Liste_Choix" size="3">
<option value="Option_1">Option_1</option>
<option value="Option_2">Option_2</option>
<option value="Option_3">Option_3</option>
</select> <BR>
<textarea name="Zone_Texte" cols="30" rows="5"> Texte par défaut
</textarea> <BR>
<input type="checkbox" name="Case_Cocher[]" value="Case_1"> Case 1<br>
<input type="checkbox" name="Case_Cocher[]" value="Case_2"> Case 2<br>
<input type="checkbox" name="Case_Cocher[]" value="Case_3"> Case 3<br>
<input type="radio" name="Case_Radio" value="Case radio 1"> radio 1<br>
<input type="radio" name="Case_Radio" value="Case radio 2"> radio 2<br>
<input type="radio" name="Case_Radio" value="Case radio 3"> radio 3<br>
<input type="reset" name="Annulation" value="Annuler">
<input type="submit" name="Soumission" value="Soumettre">
</form>
```

## La gestion des formulaires

### Exemple

Fichier page\_cible.php

```
<?php
$resultat = $_GET["Champ_saisie"] . "<br>";
$resultat .= $_GET["Liste_Choix"] . "<br>";
$resultat .= $_GET["Zone_Texte"] . "<br>";
for ($i = 0; $i < count($_GET["Case_Cocher"]); $i++)
{
    $resultat .= $_GET["Case_Cocher"][$i] . "<br>";
}
$resultat .= $_GET["Case_Radio"] . "<br>";
echo $resultat;
?>
```

## La gestion des formulaires

### Récupération des données

- La plupart des éléments d'un formulaire n'acceptent qu'une seule et unique valeur

```
echo $_GET["Champ_saisie"] ;
```

- Pour des cases à cocher et les listes à choix multiples, plusieurs valeurs peuvent être sélectionnées entraînant l'affectation d'un tableau de valeurs aux variables correspondantes.

```
for ($i = 0; $i < count($_GET["Case_Cocher"]); $i++)
{
    $resultat .= $_GET["Case_Cocher"][$i] . "<br>";
}
```

## La gestion des formulaires

### Exercices

- Écrire un formulaire qui demande le nom et l'âge de l'utilisateur. Le bouton submit de ce formulaire provoquera l'affichage d'une page qui contient cette phrase : « Bonjour ..., vous avez ... ans... »
- Écrire une page HTML qui affiche le formulaire suivant et envoi les données à une autre page PHP appelée calcul.php qui doit afficher le résultat de l'opération.

Valeur 1

Valeur 2

Addition

Soustraction

Multiplication

## Plan

- Chapitre 1 : Introduction
- Chapitre 2 : Notions de base
- Chapitre 3 : La gestion des fichiers avec PHP
- Chapitre 4 : La gestion des formulaires
- **Chapitre 5: Les cookies et les sessions**
- Chapitre 6 : POO avec PHP
- Chapitre 7 : Interfaçage avec une base de données
- Chapitre 8 : Frameworks