

# COURS DE PROGRAMMATION WEB DYNAMIQUE



# PARTIE 1 :

## *Passage de paramètres à un script*

# PASSAGE DE PARAMÈTRES À UN SCRIPT

## introduction

- Lecture des données
- Site web : interaction des internautes avec le webmaster envoi de l'information par l'internaute au webmaster (propriétaire du site) :
  - recueil des avis des visiteurs, leur e-mail, leur questions ...
  - Bon de commande, sondage, jeu...
  - Besoin d'une passerelle entre le visiteur et des applications écrites en Java ou JavaScript.

====> formulaires

# PASSAGE DE PARAMÈTRES À UN SCRIPT

## Remarque

En php le passage de valeur d'un script à l'autre peut se faire :

- soit par formulaires
- soit par URL
- soit par les sessions

# LES FORMULAIRES

## Qu'est ce qu'un formulaire ?

Un formulaire est une fiche que l'utilisateur peut remplir, ces informations ainsi saisies sont traitées par le serveur htdocs.

## **Intérêt des formulaires**

Les formulaires interactifs permettent de doter les pages web d'éléments interactifs permettant le dialogue avec l'internaute, la saisie de données ...

# LES FORMULAIRES

## Exemple de formulaire

Un formulaire est une fiche que l'utilisateur peut remplir, ces informations ainsi saisies sont traitées par le serveur htdocs.

## **Intérêt des formulaires**

Les formulaires interactifs permettent de doter les pages web d'éléments interactifs permettant le dialogue avec l'internaute, la saisie de données ...

# LES FORMULAIRES

## Balise des formulaire

Un formulaire HTML est créé par la balise `<form></form>` Il peut contenir :

- Des champs de saisie de texte
- Des boutons radio
- Des cases à cocher
- Des listes de sélections ou liste déroulante
- Un bouton de soumission du formulaire et son contenu au serveur web
- Un bouton de remise à zéro pour effacer ou modifier les données saisies ...

# LES FORMULAIRES

## La balise principale

Un formulaire commence et finit par : FORM

```
<FORM ACTION=" " METHOD=" " > ... </FORM>
```

- *Les attributs METHOD et ACTION sont obligatoires*
- **ACTION** = " adresse du script qui va recevoir les données saisies dans le formulaire "

*exemple :*

```
<form action="page.php" methode=" " > ... </form>
```

Les données peuvent être envoyées aussi bien à la page contenant le formulaire qu'à une autre



# LES FORMULAIRES

## La balise principale

Un formulaire commence et finit par : FORM

```
<FORM ACTION=" " METHOD=" " > ... </FORM>
```

- *Les attributs METHOD et ACTION sont obligatoires*
- **METHOD** = " la méthode d'envoi des données"  
La transmission des données peut se faire selon une des deux méthodes d'envoi : **GET** ou **POST**

***exemple :***

```
<form action=" " method=" POST/GET" > </form>
```

# LES FORMULAIRES

## La balise principale

Un formulaire commence et finit par : FORM

```
<FORM ACTION=" " METHOD=" " > ... </FORM>
```

- *Les attributs METHOD et ACTION sont obligatoires*
  - Les données présentes dans le formulaire sont envoyées au script sous forme de paires (**nom/valeur**) :
    - C'est-à-dire sous forme : nom de l'élément du formulaire, le caractère "=", puis la valeur associée.
    - Les paires (nom/valeur) sont séparées entre elles par le caractère &.

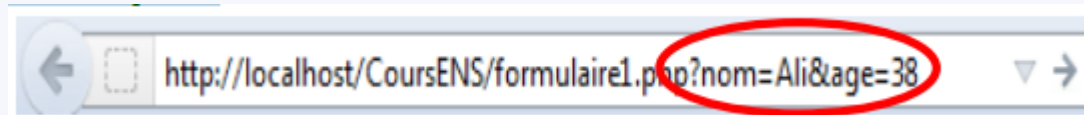
# LES FORMULAIRES

## La balise principale

► L'attribut METHOD : méthodes d'envoi

➤ **La méthode GET** passe les informations d'un formulaire par l'intermédiaire des URLs. Les données sont placées directement à la suite de l'adresse URL de la page appelée.

*exemple:*



### inconvénients:

- visibilité des données dans la barre d'adresse du navigateur donc modifiable par l'internaute.
- Longueur totale limitée à 255 caractères ==> impossibilité de transmission d'un volume de données important

# LES FORMULAIRES

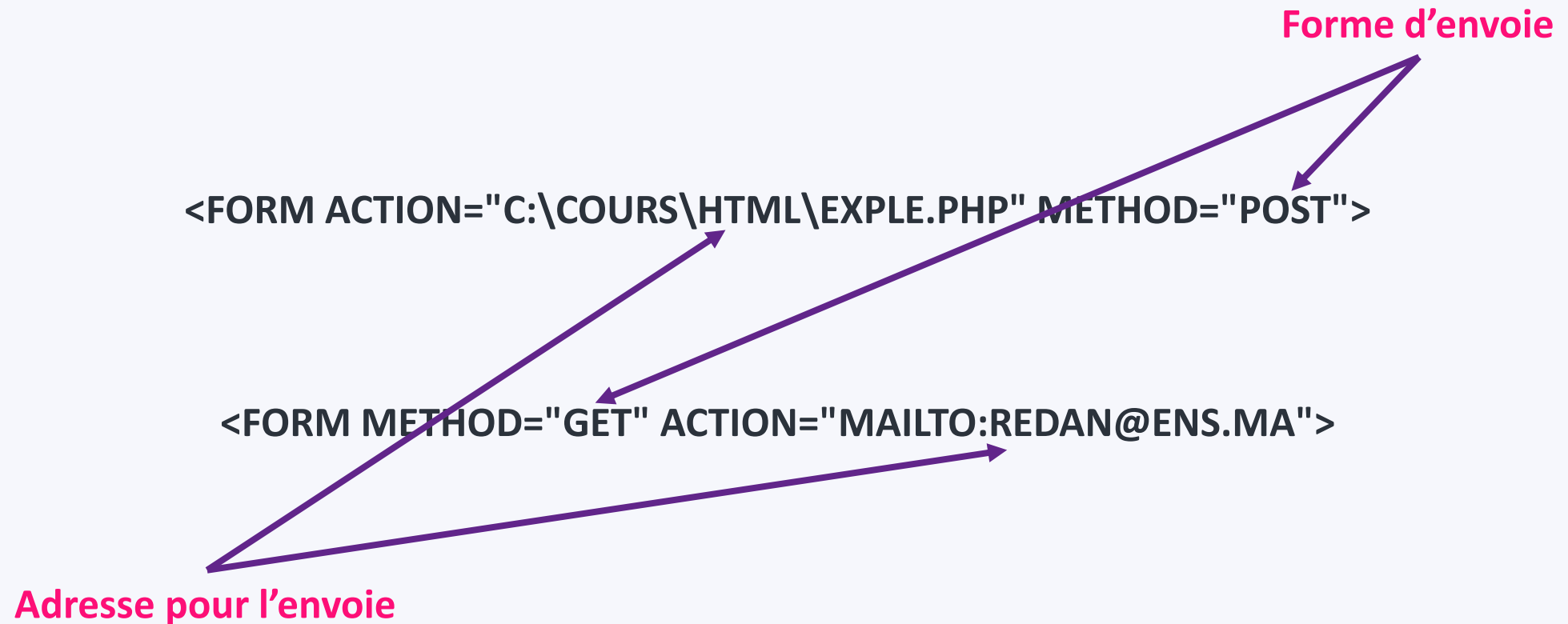
## La balise principale

► L'attribut METHOD : méthodes d'envoi

- **La méthode POST** est la plus utilisée, elle regroupe les informations dans l'entête d'une requête HTTP:
  - . Assure une confidentialité efficace des données
  - . Lorsque l'utilisateur valide et envoie les données du formulaire au serveur, les variables du formulaire seront connues comme variables globales du script php destination.

# LES FORMULAIRES

*exemples :*



# LES FORMULAIRES

## Les éléments du formulaire

A l'intérieur de la balise double "form", on peut insérer deux types d'éléments interactifs :

- ◆ Des éléments pour saisir des données
- ◆ Des éléments pour faire des choix Ces éléments sont contenus dans trois types de balises :
  - ◆ la balise **INPUT** : un ensemble de boutons et de champs de saisie à une ligne
  - ◆ la balise **TEXTAREA** : une zone de saisie multilignes
  - ◆ la balise **SELECT** : une liste à choix multiples

# LES FORMULAIRES

## Les éléments du formulaire

Les balises utilisées dans les formulaires :

`<form action= " " method= " " > ... </form>`

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

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

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

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

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

`<select name ="select" size="1">`

`<option value="R">choix1</option>`

`<option value="D"> choix2 </option>`

`</select>`

`<textarea name="" cols="" rows=""></textarea>`

[zone de saisie](#)

[bouton d'envoi](#)

[bouton effacer](#)

[liste de selection](#)

[zone de texte](#)

# PASSAGE DE PARAMÈTRES À UN SCRIPT

## Le traitement PHP:

Les informations envoyées au moyen d'un formulaire sont stockées dans les tableaux superglobaux de l'environnement PHP.

Les tableaux utilisés sont :

- **\$\_GET** : Les variables de l'URL (méthode GET)
- **\$\_POST** : Les variables envoyées par la méthode POST
- **\$\_FILES** : Les fichiers envoyés par formulaire



# PASSAGE DE PARAMÈTRES À UN SCRIPT

## Le traitement PHP – \$\_POST

La récupération des données dans le fichier **.php**, se fait dans le tableau `$_POST` :

**`$_POST["nom_variable"]`**

Pour parcourir le contenu d'un formulaire, on manipule le tableau `$_POST` comme un tableau associatif :

==> le nom de chaque champ (propriété **name** des divers contrôles) du formulaire est utilisé comme **index** de `$_POST`. Dans l'exemple suivant, deux contrôles portent un nom : "**login**" devient **`$_POST['login']`** et "password" devient **`$_POST['password']`**.