

TRANSMISSION DE DONNÉES DE PAGE EN PAGE

Envoyer des paramètres dans l'URL

2

- Écrire les paramètres dans l'url
- Syntaxe :
 - ▣ `page.php?param1=val eur1¶m2=val eur2¶m3=val eur3...`
- Exemple :
 - ▣ ``
- **Récupérer les paramètres dans la page destinataire à travers le tableau \$_GET**
- Exemple :
 - ▣ `echo $_GET['prenom'] ;`

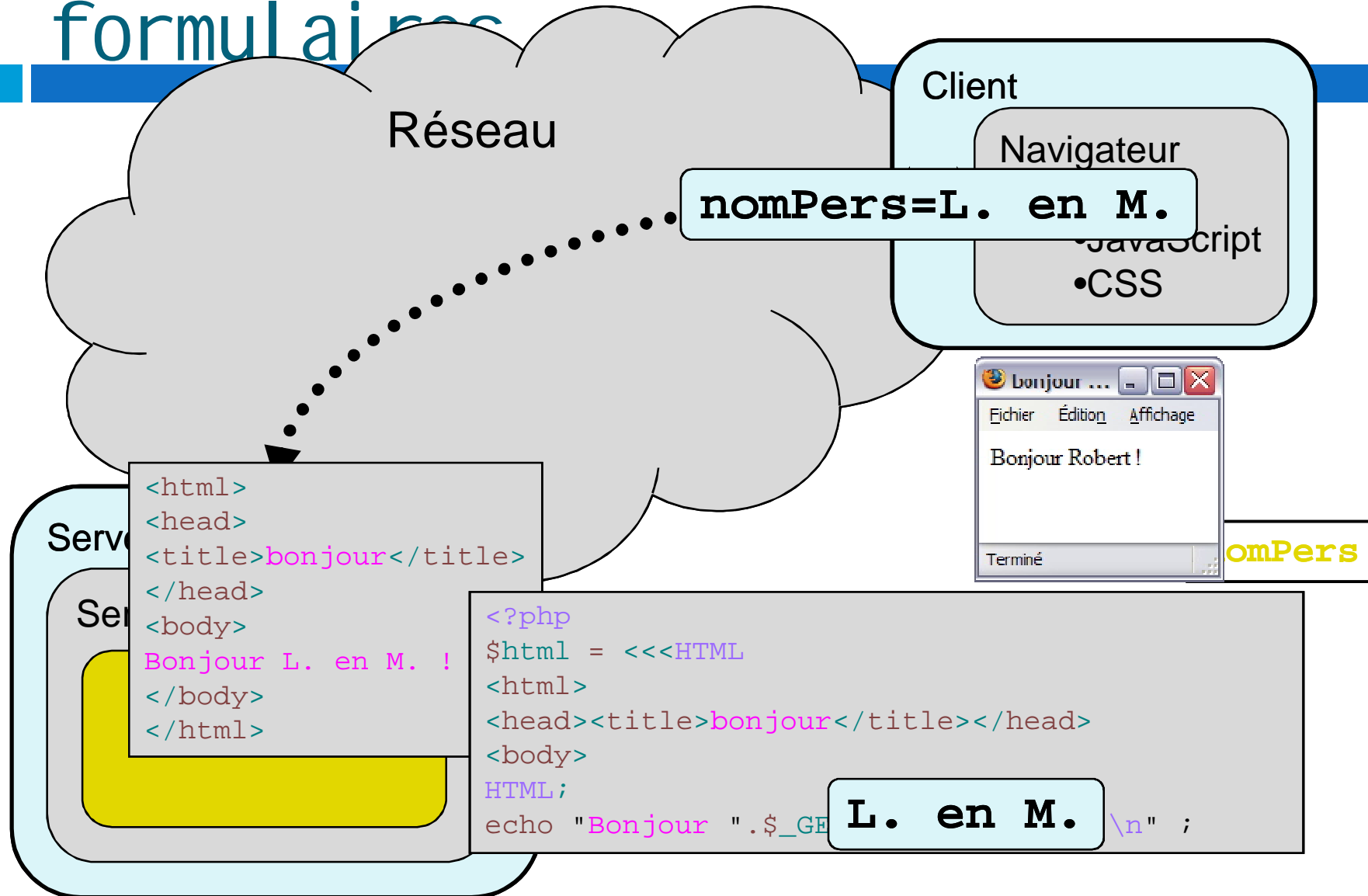
propriétés

3

- Test d'existence du paramètre :
 - `isset($_GET['prenom']`
- Conversion de type:
 - `$_GET['repeter'] =(int) $_GET['repeter']`

Traitement des données de formulaires

4



Traitements des données de formulaires

5

- **1^{ère} étape** : Le champ **ACTION** du formulaire désigne une page PHP du serveur.
- **2^{ème} étape** : ajouter l'attribut (method) avec le contenu POST ou GET
- les données sont contenues dans un tableau associatif **\$_POST** ou **\$_GET**
- Exemple: `<form method="post" action="page.php">`

POST Vs GET

6

- GET : Les données transiteront par l'URL (peu utilisée car on ne peut pas envoyer beaucoup d'informations dans l'URL)
- POST : Les données ne transiteront pas par l'URL (envoyer autant de données que l'on veut)

Exemple – Formulaire HTML

7

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
  <title>formulaire</title>
</head>
<body>
  <form action="valide1.php" method="get">
    Nom: <input type="text" name="nomPers">
    <input type="submit" value="Envoyer">
  </form>
</body>
</html>
```

Exemple – Traitement en PHP

8

```
<html>
<head>
  <title>Validation</title>
</head>
<body>
<?php
if (isset($_GET['nomPers']))
{
    if (!empty($_GET['nomPers']))
    {
        echo "Vous avez saisi '"
            . $_GET['nomPers'] . "'\n" ;
    }
    else
        echo "Aucune valeur saisie\n";
}
else
    echo "Utilisation incorrecte\n" ;
?>
```

`$_GET['nomPers']`
est-il défini ?

`$_GET['nomPers']`
est-il vide ?

Retour des sélecteurs

9

Sélecteur : renvoi la valeur du choix sélectionné

Sélecteur à choix simple

```
<select name="sel">
  <option>Fraise
  <option>Pomme
  <option>Poire
</select>
```

- Appel du fichier à travers
result.php?sel=Pomme

Sélecteur à choix multiple

```
<select name="sel[]" multiple>
  <option>Fraise
  <option>Pomme
  <option>Poire
</select>
```

- Appel du fichier à travers
result.php?sel[]=Pomme&sel[]=Poire
- Traitement dans le fichier
result.php:
foreach(\$_GET['sel'] as \$ele)

Retour des CHECKBOXs

10

□ Transmission du paramètre lorsqu'il est coché

□ Exemple :

```
<input type="checkbox" name="ch1" value="Fraise">Fraise<br>
```

```
<input type="checkbox" name="ch2" value="Pomme" >Pomme <br>
```

```
<input type="checkbox" name="ch3" value="Pomme" >Pomme <br>
```

☐ Fraise
☐ Poire
☐ Pomme

entrer

☒ Fraise
☐ Poire
☐ Pomme

entrer

☒ Fraise
☐ Poire
☒ Pomme

entrer

Appel du fichier :
result.php

Appel du fichier :
result.php?ch1=Fraise

Appel du fichier :
result.php?ch1=Fraise
&ch3=Pomme

Retour des CHECKBOXs

11

□ Transmission d'un ensemble de paramètres

□ Exemple :

```
<input type="checkbox" name="ch[]" value="Fraise">Fraise<br>
```

```
<input type="checkbox" name="ch[]" value="Pomme" >Pomme <br>
```

```
<input type="checkbox" name="ch[]" value="Pomme" >Pomme <br>
```

☐ Fraise
☐ Poire
☐ Pomme

entrer

☒ Fraise
☐ Poire
☐ Pomme

entrer

☒ Fraise
☐ Poire
☒ Pomme

entrer

Appel du fichier :
result.php

Appel du fichier :
result.php?ch[]=Fraise

Appel du fichier :
result.php?ch[]=Fraise&
ch[]=Pomme

radioBox

12

□ Retourne la valeur de l'élément coché

□ Exemple :

```
<input type="radio" name="nb" value="1" checked="checked"/><label>1 personne</label>
```

```
<input type="radio" name="nb" value="2"/><label>2 personnes</label>
```

```
<input type="radio" name="nb" value="3"/><label>3 personnes</label>
```

☐ 1 personne ☐ 2 personnes ☒ 3 personnes

➔ resultat.php?nb=3

Exercice :

13

On propose de créer une application pour le choix d'un appartement à louer dans une résidence.

Donner le code d'une page html contenant un radiobox permettant de choisir le type entre (studio, S+1, S+2, S+3, duplex) et des checkboxes pour choisir (garage, piscine, chauffage central, ascenseur).

Donner le code d'une page php, permettant de ré-afficher les choix de l'utilisateur et de retourner à la page html.

Envoi de fichiers

14

- Indiquer l'envoi de fichier à travers le formulaire:

```
<form method="post" action="prix.php" enctype="multipart/form-data">
```

...

```
<input type="file" name="monfichier" />
```

- Lors de l'exécution du fichier php, le fichier est stocké dans un dossier temporaire
- Une variable `$_FILES['name']` est créée pour chaque fichier envoyé
-

Informations sur le fichier

15

- `$_FILES['monfichier']['name']`: nom du fichier lors de son envoi
`$_FILES['monfichier']['size']` : taille en octets, taille maximale en PHP est e 8 Mo
- `$_FILES['monfichier']['type']`
- `$_FILES['monfichier']['error']` : vaut zéro s'il n'y a pas un problème d'envoi, sinon contient le motif de l'erreur.
- `$_FILES['monfichier']['tmp_name']`: l'emplacement temporaire du fichier

Récupération du fichier

16

- Le fichier est accepté en utilisant la commande « `move_uploaded_file()` » avec deux paramètres :
 - le nom temporaire du fichier
`($_FILES['monfichier']['tmp_name'])`
 - le chemin du stockage définitive du fichier
- Exemple :
`move_uploaded_file($_FILES['fichier']['tmp_name'], "C:\essay.jpg") ;`

Exercices :

17

Donner le code nécessaire permettant de héberger un photo de type jpg, jpeg ou gif avec une taille inférieure à 2Mo.

Accès au fichier

18

- Principe :
- Configurer le fichier pour être accessible en lecture ou/et écriture
- Les étapes :
 - ▣ Ouvrir le fichier
 - ▣ Effectuer le traitement nécessaire
 - ▣ Fermer le fichier

□ Ouverture du fichier :

□ `$monfichier=fopen(cheminFichier, type-accès);`

Mode	Explication
r	Ouvre le fichier en lecture seule. Cela signifie que vous pourrez seulement lire le fichier.
r+	Ouvre le fichier en lecture et écriture. Vous pourrez non seulement lire le fichier, mais aussi y écrire (on l'utilisera assez souvent en pratique).
a	Ouvre le fichier en écriture seule. Mais il y a un avantage : si le fichier n'existe pas, il est automatiquement créé.
a+	Ouvre le fichier en lecture et écriture. Si le fichier & n'existe pas, il est créé automatiquement. Attention : le répertoire doit avoir un CHMOD à 777 dans ce cas ! À noter que si le fichier existe déjà, le texte sera rajouté à la fin.

□ Fermer un fichier :

■ `fclose($monfichier) ;`

Lecture/écriture

20

- Lecture de la première ligne :
 - `$ligne = fgets($monfichier) ;`
- Écriture avec la commande :
 - `fputs($monfichier, contenu);`
- Propriétés :
 - Lors de l'ouverture du fichier, un curseur est pointé au début du fichier.
 - À chaque `fgets`, le curseur pointe la ligne suivante
 - L'écriture aura lieu à l'emplacement du curseur
 - N.B. : utiliser `"\r\n"` le retour à la ligne

Exercices

21

Exercice 1 :

Soit un fichier « `isitec.txt` » contenant les informations administratives de notre institut (le nom, l'adresse, le téléphone, ...). Afficher ce contenu dans la page « `contact.php` ».

Exercice 2 :

Créer 3 pages dans la page « `accueil.html` » ayant un lien vers la 2^{ème} page pour l'ajout d'une information de l'institut et un lien vers la 3^{ème} page pour l'affichage des informations.