

Développement PHP

Partie 3 : Transmettre des données

DENIS LOEUILLET – IFA - 2017

Partie 3 : Transmettre des données

- Variables superglobales
- Transmettre des données avec l'URL
- Transmettre des données avec les formulaires
- Sessions & Cookies
- Lire et écrire dans un fichier

Partie 3 : Transmettre des données Avec l'URL

1. Envoyer des paramètres dans l'URL
2. Récupérer les paramètres en PHP
3. Contrôler les données reçues

Partie 3 : Transmettre des données

Avec l'URL

- URL : signification
 - Uniform Resource Locator : littéralement « localisateur uniforme de ressource »)
 - Ou adresse web
 - Invention du World Wide Web
 - Chaîne de caractères utilisée pour adresser les ressources internet :
 - ❖ document HTML,
 - ❖ image,
 - ❖ son,
 - ❖ boîte aux lettres électronique,
 - ❖ ...
 - Inventées pour pouvoir indiquer avec une notation (d'où l'adjectif « uniforme ») aux navigateurs web comment accéder à toutes les ressources d'Internet.

Partie 3 : Transmettre des données Avec l'URL

- URL : description

- Exemple : <http://www.example.com/index.php?q=req&q2=req2#signet>

- ❖ Protocole :

- ❖ http : protocole de communication, en l'occurrence ici HTTP pour accéder à un serveur web.

- ❖ : caractère de séparation obligatoire si le protocole est précisé

- ❖ Localisation complète de la ressource :

- ❖ // chaîne de caractères pour les protocoles dont la requête comprend un chemin d'accès, permettant de préciser et localiser le service avant ce chemin

- ❖ www.example.com : nom de domaine du service (emplacement du service)

- ❖ Nom complet de la ressource à demander sur le service une fois connecté

- ✓ Index.php : nom de la page web

- ❖ Données supplémentaires optionnelles, transmises au service lors de la demande à la ressource

- ✓ ? - caractère de séparation obligatoire pour indiquer que des données complémentaires suivent.

- ✓ q=req&q2=req2 - chaîne de requête, traitée par la page web sur le serveur

- ❖ Données supplémentaires optionnelles, pour l'exploitation de la ressource après son obtention par le logiciel client (non transmises dans la requête au service)

- ✓ # - caractère de séparation obligatoire pour indiquer un signet (ou une balise).

- ✓ signet - identificateur du signet ou de la balise, il s'agit d'un emplacement à l'intérieur de la page web retournée par le service, cette donnée sera traitée par le navigateur web.

Partie 3 : Transmettre des données

Avec l'URL

- URL : exemples
 - URL de Wikipédia :
 - ✓ <http://fr.wikipedia.org/>
 - URL d'une page sur Wikipédia
 - ✓ http://fr.wikipedia.org/wiki/Uniform_Resource_Locator
 - URL d'un fichier sur un site FTP
 - ✓ <ftp://ftp.rfc-editor.org/in-notes/rfc2396.txt>
 - URL d'un lien (mailto) vers une adresse courriel
 - ✓ <mailto:quidam@example.com>
 - URL d'un forum de discussion de Usenet
 - ✓ <news:fr.comp.infosystemes.www.auteurs>
 - URL d'un fichier de l'ordinateur
 - ❖ <file:///C:/Users/Public/Desktop/Microsoft%20Word.lnk>

Partie 3 : Transmettre des données Avec l'URL

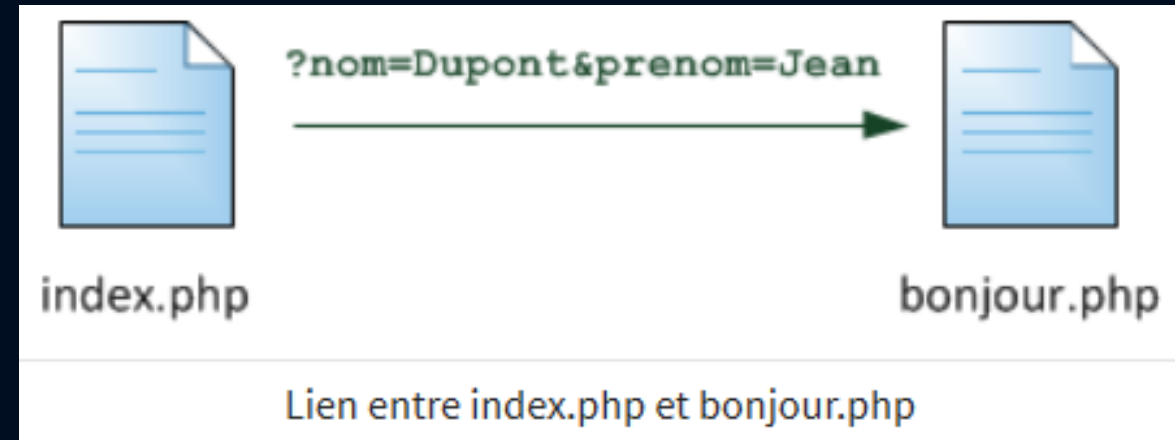
1. Envoyer des paramètres dans l'URL

- URL permet de transmettre des informations
- Former une URL pour envoyer des informations
 - Exemple : <http://www.monsite.com/bonjour.php?nom=Dupont&prenom=Jean>
 - Nom de domaine : www.monsite.com
 - Page PHP : `bonjour.php`
 - Paramètres :
 - ✓ Paramètre 1
 - Nom du paramètre : `nom`
 - Valeur du paramètre : `Dupont`
 - ✓ Paramètre 2 :
 - Nom du paramètre : `prenom`
 - Valeur du paramètre : `Jean`
 - ✓ Le point d'interrogation sépare le nom de la page PHP des paramètres.

Partie 3 : Transmettre des données

Avec l'URL : envoyer des paramètres dans l'URL

- Créer un lien avec des paramètres
 - On peut créer des lien en HTML qui transmettent des paramètres d'une page à une autre
 - Exemple :
 - ✓ Vous disposez de deux fichiers sur votre site :
 - ❖ index.php : l'accueil
 - ❖ bonjour.php
 - ✓ On veut faire un lien pour aller :
 - ❖ de index.php
 - ❖ à bonjour.php
 - ✓ Ce lien transmet des informations dans l'URL
 - ✓ Dans le fichier index.php on ajoute le code suivant :



```
1 <a href="bonjour.php?nom=Dupont&prenom=Jean">Dis-moi bonjour !</a>
```


Partie 3 : Transmettre des données

Avec l'URL : envoyer des paramètres dans l'URL

```
1 <a href="bonjour.php?nom=Dupont&prenom=Jean">Dis-moi bonjour !</a>
```

- Le « & » dans le code a été remplacé par le code HTML « & » dans le lien.
- Rien à voir avec PHP : simplement, en HTML, on demande à ce que les « & » soient écrits « & » dans le code source. Si vous ne le faites pas, le code ne passera pas la validation W3C.
- Que fait ce lien :
 - Il appelle la page « bonjour.php »
 - Il envoie les paramètres :
 - ✓ nom : Dupont
 - ✓ prenom : Jean
 - Il reste à récupérer les paramètres dans la page « bonjour.php »

Partie 3 : Transmettre des données Avec l'URL

2. Récupérer les paramètres en PHP

- On récupère les paramètres grâce à :
 - l'array « `$_GET` » : variable superglobale, elle contient les données envoyées en paramètres dans l'URL
 - `$_GET` est généré automatiquement par PHP
 - `$_GET` est un tableau associatif
 - ✓ Les clés (ou étiquettes) correspondent aux noms des paramètres envoyés dans l'URL
- Dans notre exemple, nous avons fait un lien vers :
 - `bonjour.php?nom=Dupont&prenom=Jean`
 - Nous avons accès aux variables suivantes :
 - On peut donc récupérer ces données pour :
 - ✓ les traiter
 - ✓ les afficher

Nom	Valeur
<code>\$_GET['nom']</code>	Dupont
<code>\$_GET['prenom']</code>	Jean

Partie 3 : Transmettre des données

Avec l'URL : récupérer les paramètres en PHP

- Afficher les données
 - Créez le fichier index.php et ajoutez le code suivant :

Nom	Valeur
<code>\$_GET['nom']</code>	Dupont
<code>\$_GET['prenom']</code>	Jean

```
1 <a href="bonjour.php?nom=Dupont&prenom=Jean">Dis-moi bonjour !</a>
```

- Créez le fichier bonjour.php avec le code suivant :

```
1 <p>Bonjour <?php echo $_GET['prenom'] . ' ' . $_GET['nom']; ?> !</p>
```

➤ Ce code va afficher les valeurs associées aux variables « prenom » et « nom » passées dans l'URL

Partie 3 : Transmettre des données Avec l'URL

- 3. Contrôler les données reçues
 - Il ne faut JAMAIS faire confiance aux variables qui transitent de page en page
 - Tous les internautes peuvent modifier les URLs
 - Essayez de modifier votre URL comme suit :
 - `bonjour.php?nom=Dupont&prenom=Marc`
 - Que se passe-t-il?
 - Quel est le problème?
 - on ne peut pas avoir confiance dans les données qu'on reçoit
 - Dans le cas présent ça n'a pas de conséquences graves mais ce n'est pas toujours le cas

Partie 3 : Transmettre des données

Avec l'URL : contrôler les données reçues

- Tester la présence d'un paramètre
 - L'internaute peut supprimer l'ensemble des paramètres
 - ✓ Que va-t-il se passer ?
 - ❖ on essaie d'afficher `$_GET['nom']` et `$_GET['nom']`
 - ❖ elles n'existent pas
 - ❖ PHP nous avertit qu'on essaie d'utiliser des variables qui n'existent pas : « Undefined index »
 - ❖ on peut utiliser la fonction « `isset()` » :
<http://php.net/manual/fr/function.isset.php>
 - « `isset` » teste si une variable existe
 - On peut s'en servir pour afficher un message si une ou plusieurs variables n'existent pas
 - Ce code teste si les variables `$_GET['nom']` et `$_GET['nom']` existent
 - Si oui on les utilise
 - Si non on affiche un message

```
1 <?php
2 if (isset($_GET['prenom']) AND isset($_GET['nom'])) // On a le nom et le prénom
3 {
4     echo 'Bonjour ' . $_GET['prenom'] . ' ' . $_GET['nom'] . ' !';
5 }
6 else // Il manque des paramètres, on avertit le visiteur
7 {
8     echo 'Il faut renseigner un nom et un prénom !';
9 }
10 ?>
```


Partie 3 : Transmettre des données

Avec l'URL : contrôler les données reçues

- Tester la présence d'un paramètre
 - L'internaute peut supprimer l'ensemble des paramètres
 - ✓ Effectuez à nouveau le test maintenant
 - ✓ Que va-t-il se passer ?
 - ✓ Est-ce vraiment utile ?
 - ✓ OUI, même si dans le cas présent les conséquences ne sont pas graves

Partie 3 : Transmettre des données

Avec l'URL : contrôler les données reçues

- Contrôler la valeur des paramètres
 - on va modifier notre code source pour gérer un nouveau paramètre
 - le nombre de fois que le message doit être répété
 - Les paramètres vont donc être :
 - ✓ bonjour.php?nom=Dupont&prenom=Jean&repetier=8
 - ✓ Le paramètre « repeter » définit le nombre de fois que l'on dit « bonjour » sur la page

```
1 <?php
2 if (isset($_GET['prenom']) AND isset($_GET['nom']) AND isset($_GET['repetier']))
3 {
4     for ($i = 0 ; $i < $_GET['repetier'] ; $i++)
5     {
6         echo 'Bonjour ' . $_GET['prenom'] . ' ' . $_GET['nom'] . ' !<br />';
7     }
8 }
9 else
10 {
11     echo 'Il faut renseigner un nom, un prénom et un nombre de répétitions !';
12 }
13 ?>
```

Partie 3 : Transmettre des données

Avec l'URL : contrôler les données reçues

- Contrôler la valeur des paramètres
 - On teste si le paramètre « repeter » existe lui aussi
 - Si tous les paramètres existent
 - ✓ on fait une boucle
 - ✓ La boucle incrémente un compteur \$i pour répéter le message de bienvenue le nombre de fois indiqué.

```
1 <?php
2 if (isset($_GET['prenom']) AND isset($_GET['nom']) AND isset($_GET['repeter']))
3 {
4     for ($i = 0 ; $i < $_GET['repeter'] ; $i++)
5     {
6         echo 'Bonjour ' . $_GET['prenom'] . ' ' . $_GET['nom'] . ' !<br />';
7     }
8 }
9 else
10 {
11     echo 'Il faut renseigner un nom, un prénom et un nombre de répétitions !';
12 }
13 ?>
```

Partie 3 : Transmettre des données

Avec l'URL : contrôler les données reçues

- Contrôler la valeur des paramètres
 - Que se passe-t-il si on remplace dans l'URL 8 par 1984986545665156 ?
 - La page PHP va boucler un très grand nombre de fois et le serveur ne va pas apprécier
 - PHP va certainement arrêter le processus au bout d'un certain temps
 - Il faut contrôler également le contenu des variables
 - ✓ Contrôler la valeur reçue
 - ❖ Poser par exemple des bornes à une variable entière
 - ✓ Contrôler le type de variable reçu
 - ❖ Vérifier dans notre cas que la variable est bien de type int
 - ✓ Pour vérifier qu'une variable est de type int :
 - ❖ On peut forcer la conversion de la variable en type int
 - Faire du transtypage

```
1 <?php
2 $_GET['repeter'] = (int) $_GET['repeter'];
3 ?>
```

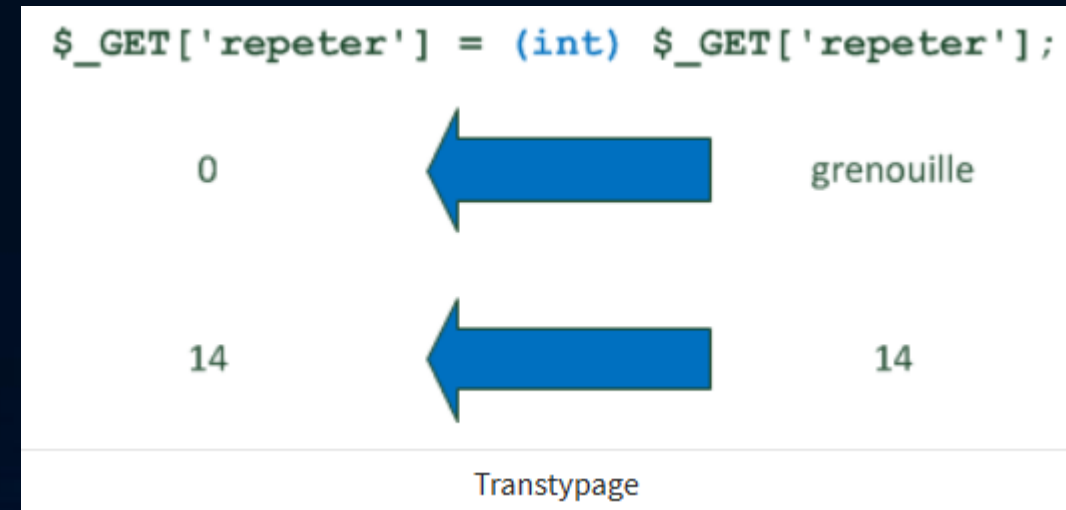
Partie 3 : Transmettre des données

Avec l'URL : contrôler les données reçues

- Contrôler la valeur des paramètres

```
1 <?php
2 $_GET['repeter'] = (int) $_GET['repeter'];
3 ?>
```

- Il faut lire cette instruction de droite à gauche.
- Le(int) entre parenthèses est comme un tuyau de conversion. Tout ce qui transite à travers ressort forcément en une valeur de type int(entier).
- Si la valeur est bien un entier alors elle n'est pas modifiée.
- En revanche, si la variable contient autre chose qu'un entier elle est transformée en entier.
 - ✓ Ici, comme PHP ne peut pas associer de valeur à grenouille, il lui donne 0.
- Après avoir exécuté cette instruction, la variable `$_GET['repeter']` contient maintenant forcément un nombre entier.
- Il ne reste plus qu'à vérifier que ce nombre est bien compris entre 1 et 100 à l'aide d'une condition



Partie 3 : Transmettre des données

Avec l'URL : contrôler les données reçues

- Contrôler la valeur des paramètres
 - Le code final
- Il faut toujours vérifier que :
 - tous les paramètres que vous attendiez sont là
 - qu'ils contiennent bien des valeurs correctes comprises dans des intervalles raisonnables.

```
1 <?php
2 if (isset($_GET['prenom']) AND isset($_GET['nom']) AND isset($_GET['repeter']))
3 {
4     // 1 : On force la conversion en nombre entier
5     $_GET['repeter'] = (int) $_GET['repeter'];
6
7     // 2 : Le nombre doit être compris entre 1 et 100
8     if ($_GET['repeter'] >= 1 AND $_GET['repeter'] <= 100)
9     {
10         for ($i = 0 ; $i < $_GET['repeter'] ; $i++)
11         {
12             echo 'Bonjour ' . $_GET['prenom'] . ' ' . $_GET['nom'] . ' !<br />';
13         }
14     }
15 }
16 else
17 {
18     echo 'Il faut renseigner un nom, un prénom et un nombre de répétitions !';
19 }
20 ?>
```

Partie 3 : Transmettre des données

Avec l'URL : contrôler les données reçues

- Contrôler la valeur des paramètres

- Résumé

- ✓ Une URL représente l'adresse d'une page web (commençant généralement par `http://`).
 - ✓ Lorsqu'on fait un lien vers une page, il est possible d'ajouter des paramètres sous la forme `bonjour.php?nom=Dupont&prenom=Jean` qui seront transmis à la page.
 - ✓ La page `bonjour.php` dans l'exemple précédent recevra ces paramètres dans un array nommé `$_GET` :
 - ❖ `$_GET['nom']` aura pour valeur « Dupont » ;
 - ❖ `$_GET['prenom']` aura pour valeur « Jean ».
 - ✓ Cette technique est très pratique pour transmettre des valeurs à une page
 - ❖ Il faut prendre garde au fait que le visiteur peut les modifier très facilement.
 - ❖ Il ne faut donc pas faire aveuglément confiance à ces informations, et tester prudemment leur valeur avant de les utiliser.
 - ✓ La fonction `isset()` permet de vérifier si une variable est définie ou non.
 - ✓ Le transtypage est une technique qui permet de convertir une variable dans le type de données souhaité. Cela permet de s'assurer par exemple qu'une variable est bien un `int` (nombre entier).

Partie 3 : Transmettre des données Avec l'URL

- Exercice :

Créez un lien dans votre fichier index.php :

- ce lien pointe la page cible.php et doit contenir les variables suivantes :
 - ✓ Votre nom
 - ✓ Votre prénom
 - ✓ Votre âge
- La page cible.php :
 - ✓ Récupère les variables de l'URL
 - ✓ Teste l'existence des variables
 - ✓ Teste le type des variables contenues dans l'URL
 - ✓ Teste les valeurs des différentes variables
 - ✓ Affiche un message contenant les valeurs des données reçues si celles-ci sont correctes
 - ✓ Affiche un message différent si les données reçues ne sont pas correctes