Programmation Web

4 septembre 2017

Note globale : Contrôle Continu + DS de promo

Isabelle Gonçalves - isabelle.goncalves@univ-lyon1.fr

Support de cours commun

`http://iutdoua-web.univ-lyon1.fr/~isabelle.goncalves/programmation-web`

Pré-requis

Savoir-faire concernant les sites web statiques : HTML5 + CSS3

- Module d'Introduction au Web :
  `http://champin.net/enseignement/intro-web/`
- Module d'Introduction à GIT :
  `http://champin.net/enseignement/intro-git/`

Rappel pour vos sites web sur le serveur de l'IUT

Vos pages doivent être dans votre répertoire public_html. Elles sont visibles sur : http ://iutdoua-web.univ-lyon1.fr/ votre_login

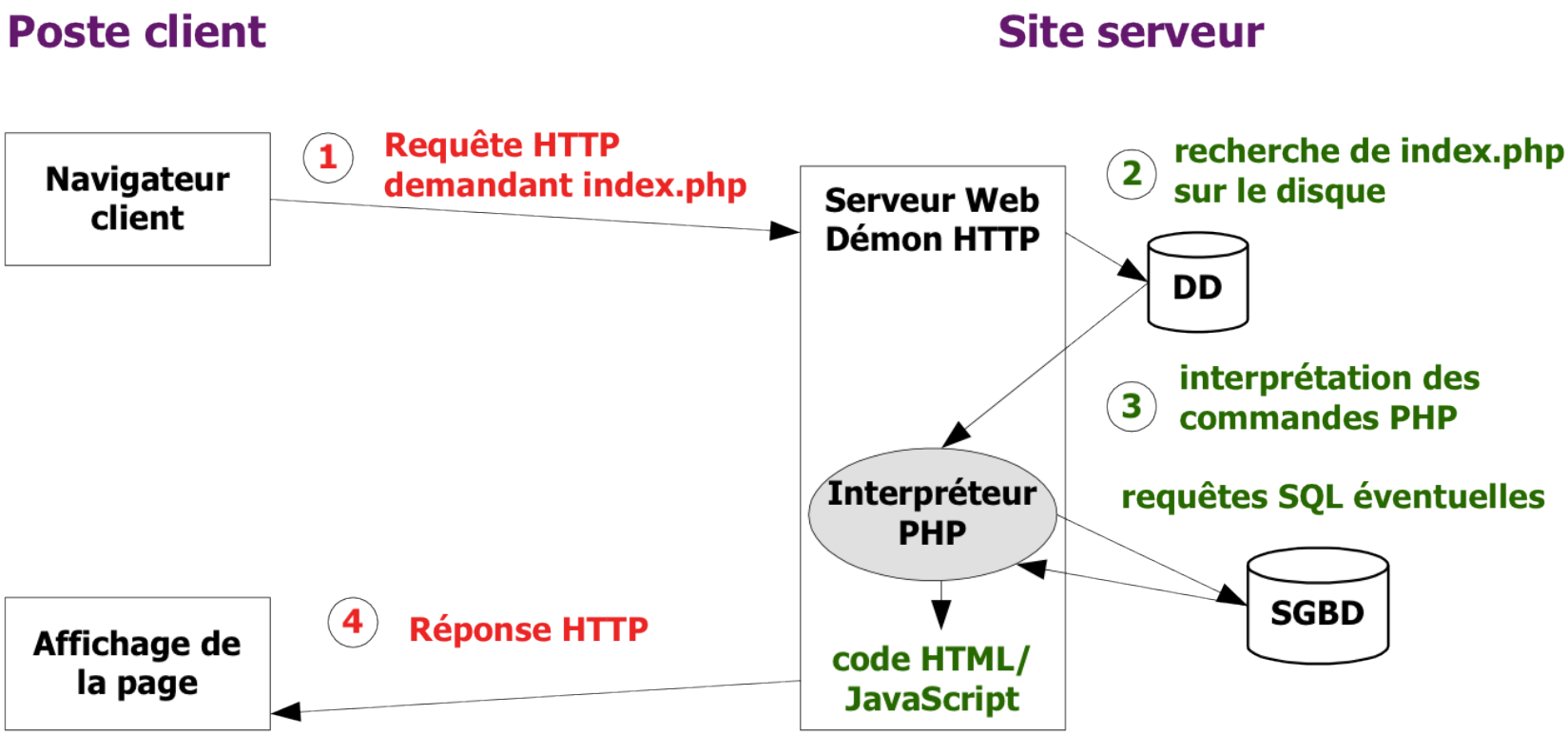
Qu'est-ce que PHP ?

- c'est un langage de script interprété **côté serveur**,
- il permet d'écrire des pages web **dynamiques**,
- indiqué par l'extension de fichier **.php**,
- en combinaison avec un SGBD (ex : MySQL), il permet de stocker et d'accéder à des informations dans une base de données.

Acronyme

PHP : Hypertext Preprocessor

Comment ça marche ?



Page index.**php** sur le disque du serveur

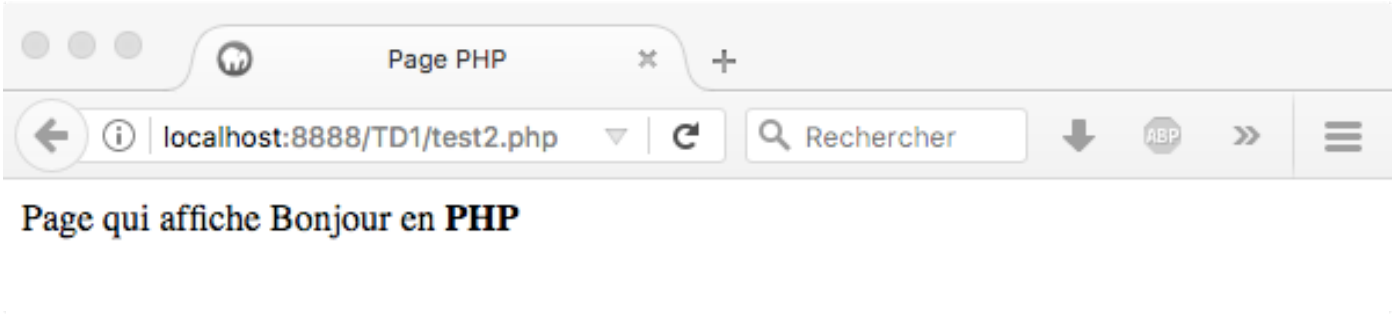
```
<!DOCTYPE html>
<HTML>
<HEAD>
  <meta charset="utf-8">
  <TITLE>Page PHP</TITLE>
</HEAD>
<BODY>
<?php
  echo 'Page qui  affiche Bonjour en <strong>PHP</
     strong>'
?>
</BODY>
</HTML>
```

L'instruction **echo** est un mot-clé de PHP

La même chose avec la syntaxe courte

```
<!DOCTYPE html>
<HTML>
<HEAD>
  <meta charset="utf-8">
  <TITLE>Page PHP</TITLE>
</HEAD>
<BODY>
   <?= 'Page qui affiche Bonjour en <strong>PHP</
      strong>' ?>
</BODY>
</HTML>
```

Page reçue par le client



Affichage du source sur le client :

```
<!DOCTYPE html>
<HTML>
<HEAD>
    <meta charset="utf-8">
    <TITLE>Page PHP</TITLE>
</HEAD>
<BODY>
 Page qui affiche Bonjour en <strong>PHP</strong>
</BODY>
</HTML>
```

divers PHP

- Les commentaires
  - multilignes /* */
  - monolignes # , //
- Les instructions sont séparées par ;
- insensible à la casse pour les noms de fonctions mais pas pour les noms de variables.
- documentation en français accessible à `http://www.php.net/manual/fr`

Les variables

- Leur identifiant est toujours précédé de $
- Pas de déclaration, l'affectation détermine le type de la variable
- Les différents types :
  - booléen (true false)
  - nombre entiers
  - flottant - nombre à virgule
  - chaîne de caractères
  - tableau
  - objet (programmation orientée objet)
- Le cast est possible comme en C.
- Beaucoup de fonctions mathématiques et pour la manipulation des strings (`http://www.php.net/manual/fr/ref.strings.php`)

L'affichage et les guillemets

```php
$age=17;
echo "L'utilisateur a $age ans";
# Le contenu de la variable est interprété

echo 'L\' utilisateur a $age ans';
#$age s'affiche tel quel (pas d'interprétation)

echo 'L\'utilisateur a '. $age .' ans';
# effectue une concaténation avec le contenu de la
    variable
```

Les structures de contrôle

- Les opérateurs de comparaison et les opérateurs logiques identiques au C + AND + OR + NOT
- Les conditions if/else (+elseif) et le switch/case comme en C
- Les boucles for et while (et do-while) comme en C
- Le foreach en plus

Attention

Le test d'égalité s'écrit donc avec un ==, voire ===

Les ternaires

```
$majeur= ($age>=18) ? true : false ;
```

Les tableaux associatifs

- Les clés peuvent être des entiers ou des chaînes de caractères
- Les clés et les valeurs peuvent être de types différents dans un même tableau

```
$prenoms=array('François', 'Michel', 'Nicole');
#crée un tableau numéroté avec des clés qui vont de 0
    à 2
$prenoms[]='Véronique'; # la clé sera la clé entière
    max du tableau + 1 donc ici 3
$coords = array (
    'prenom' => 'François',
    'nom' => 'Dupont',
    'adresse' => '3 rue du Paradis',
    'ville' =>'Paris' );
 $coords['code_postal']='75012'; #creation d'une
    nouvelle case
 $champs = array_keys($coords); #creation d'un nouveau
     tableau contenant les cles
```

Parcourir un tableau

Connaître sa taille ou l'afficher pour debugging :

```
$taille = count($tab);
print_r($tab);
```

Pour parcourir les éléments et/ou les clés :

```
for ($i=0; $i<$taille; $i++)
    echo $prenoms[$i];
foreach ($prenoms as $val)
    echo $val;
foreach ($coords as $val)
    echo $val;
foreach ($coords as $cle => $val)
    echo $cle.' vaut '.$val;
```

Destruction d'un élément du tableau :

```
unset($prenoms[0]); # ça ne décale rien
```

Les constantes

Leur valeur est affectée une fois pour toutes

- on ajoute le mot clé const devant le type.

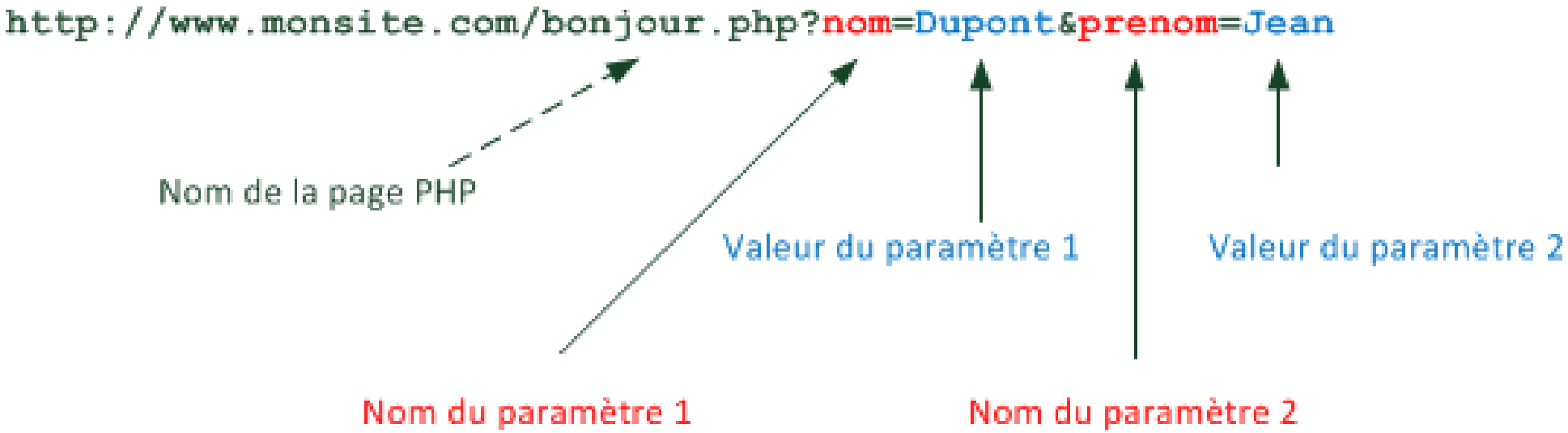
```
const VAR1 = 5 ;
const VAR2 = 'TOTO';
```

- on utilise la fonction define.

```
define (PI, 3.14);
define (VAR2, 'TOTO');
if(defined(VAR1)) {...}
```

La méthode HTTP GET

- A utiliser dans le cas d'une lecture d'information (accès à un article, recherche),
- Les données seront passées via l'URL,
- Elles sont reçues dans un tableau associatif et accessible dans $_GET['nom_du_paramètre']



Acronyme

URL : Uniform Resource Locator dont la taille est limitée.

Transmettre des données via des formulaires

- Les formulaires permettent d'échanger des informations avec le visiteur via l'URL (GET) ou pas (POST).
- Le HTML permet de créer le formulaire et le PHP permet de traiter les informations entrées dans le formulaire.

```
<form method="post" action="traitement.php">
    <!--éléments du formulaire -->
</form>
```

- La balise FORM sert de conteneur pour accueillir les éléments ou composants du formulaire : un bouton, une zone de texte ...
- Exemple : `http://iutdoua-web.univ-lyon1.fr/~isabelle.goncalves/web/formulaire.html`
- Dans traitement.php, les informations envoyées sont dans un tableau associatif et accessible dans $_POST['valeur_de_name']

Faille XSS : Cross Site Scripting

- C'est une injection de code dans une page web, grâce à une variable faillible (non ou mal sécurisée) qui sera affichée.
- Cela peut se faire via un formulaire.
- Elle résulte d'une trop grande confiance accordée aux entrées de l'utilisateur.

Démo

formulaire_injection.html et traitement.php avec ou sans htmlspecialchars et
<script>alert('boom')</script>
dans le champs de saisie.

Vérification des données transmises

- Indispensable pour éviter que le texte puisse ensuite être interprété par le navigateur comme du code javascript ...
- Solutions :
  - neutraliser les caractères spéciaux avec la fonction htmlspecialchars
  - effectuer des transtypages (cast) pour obliger les données à être du bon type ou vérifier leur type avec is_float, is_int, ...
  - limiter la longueur dans les zones de texte à ce qu'on attend même si on voit un peu large
  - vérifier que les valeurs sont dans une gamme attendue avant de lancer des calculs

```
$nb=(int)$_POST['nbenf']);
echo $nb;
$texte=htmlspecialchars($_POST['message']);
echo $texte;
```

A lire pour compléter

- Lire le chapitre du cours "PHP : Les bases".
- Lire le chapitre du cours "Annexe technique : IUT Lyon 1"