

Pages dynamiques avec PHP

Code: php-html

Originaux

[url: http://tecfa.unige.ch/guides/tie/html/php-html/php-html.html](http://tecfa.unige.ch/guides/tie/html/php-html/php-html.html)

[url: http://tecfa.unige.ch/guides/tie/pdf/files/php-html.pdf](http://tecfa.unige.ch/guides/tie/pdf/files/php-html.pdf)

Auteurs et version

- [Daniel K. Schneider](#) - [Vivian Synteta](#) - [Olivier Clavel](#)
- Version: 1.7 (modifié le 27/11/07 par DKS)

Prérequis:

- Savoir construire un formulaire HTML
Module technique précédent: [html-forms](#)
- Avoir une idée du standard "CGI"
Module technique précédent: [cgi-intro](#)
- Connaître les bases du langage PHP
Module technique précédent: [php-intro](#)

Activités:

Module d'exercices: act-quiz2

Objectifs:

- Ce module montre comment écrire des simples pages Web dynamiques
- Traiter des formulaires avec PHP
- Savoir programmer un forum simple
- Ecrire le résultat d'un formulaire dans un fichier

1. Table des matières détaillée

1. Table des matières détaillée.....	3
2. Traitement de simple formulaires avec PHP	5
2.1 Traitement de formulaires avec PHP I (Calcul)	5
Exemple 2-1: Un simple quiz et POST vers un autre fichier PHP 5	
A.La récupération des variables d'un formulaire 7	
B.Calcul et affichage des résultats 8	
C.Empêcher l'accès direct à cette page PHP (sans données) 9	
2.2 Traitement de formulaires avec PHP II	10
Exemple 2-2: Checkboxes avec PHP - arrays 10	
Exemple 2-3: Checkboxes avec PHP - multiples variables 11	
2.3 Traitement de formulaires avec PHP III	13
Exemple 2-4: Prix calcul d'une voiture avec un seul fichier PHP 13	
A.Le formulaire HTML (seulement une partie): 13	
B.A retenir (pour la suite) 14	
C.Comment traiter la requête venant d'une page par la même page ? 15	
D.Calcul et affichage des résultats 16	
E.A retenir: 17	
F.Si c'était trop compliqué: 17	
Exemple 2-5: Un simple quiz et POST vers le même fichier PHP 17	
2.4 Tester l'existence de variables et valeurs POST/GET	18
2.5 Gestion de sessions	19
Exemple 2-6: Limiter l'accès répétée à une page 19	
3. Annotation d'une page (écrire dans un fichier et inclure)	20
Exemple 3-1: Un simple système d'annotation 20	
4. Questionnaires on-line et récupération dans un fichier	23
Exemple 4-1: Questionnaire et résultats dans fichier 23	
A.le formulaire 23	
B.Afficher le contenu d'un fichier 26	
5. Autres format que HTML	27
A.Exemple de génération VRML 27	
B.Exemple de génération d'images PNG 27	

6. Conseils pour le débogage..... 28

2. Traitement de simple formulaires avec PHP

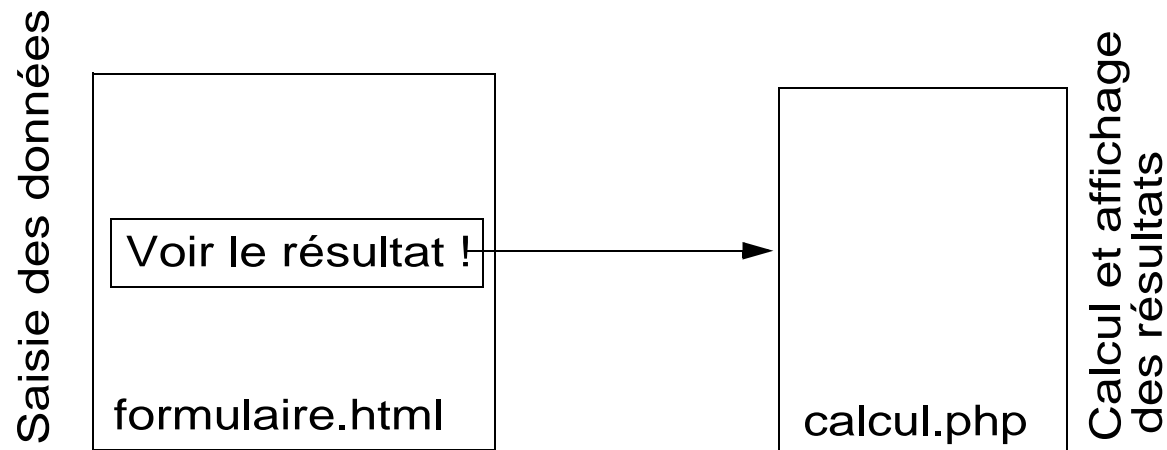
2.1 Traitement de formulaires avec PHP I (Calcul)

Exemple 2-1: Un simple quiz et POST vers un autre fichier PHP

[url: Voir: /guides/php/examples/simple-calculateur/formulaire.html](#)

[url: Source: /guides/php/examples/simple-calculateur/formulaire.text](#)

- Cet exemple montre:
 - comment traiter un formulaire HTML
 - comment calculer et afficher un résultat.



Le formulaire HTML (seulement une partie):

```
<form action="calcul.php" method="post" >
```

Quelles sont vos connaissances de HTML ?

```
<input type="radio" name="choice" value="1" checked>faibles
```

```
<input type="radio" name="choice" value="2">moyennes
```

```
<input type="radio" name="choice" value="3">bonnes
```

```
<br>
```

Indiquez votre expertise en programmation:

```
<input type="radio" name="choice2" value="1" checked>absente
```

```
<input type="radio" name="choice2" value="2">moyenne
```

```
<input type="radio" name="choice2" value="3">bonne
```

```
<P>
```

```
<input type="submit" value="Voir le résultat!">
```

```
</form>
```

Quelles sont vos connaissances de HTML ? faibles moyennes bonnes

Indiquez votre expertise en programmation: absente moyenne bonne

NAME="choice"

NAME="choice2"

La page calcul.php récupère les valeurs et calcule le résultat.

A. La récupération des variables d'un formulaire

Les données en provenance des formulaires sont stockées dans des variables spéciales de php qu'on appelle des arrays "super globaux" (car ils sont accessible depuis n'importe quel endroit du programme).

Selon la méthode choisie pour la transmission du formulaire (POST ou GET), on utilise la variable `$_POST` ou `$_GET`.

Les valeurs sont indexés dans ces arrays par le nom donné à l'input avec l'attribut NAME.

Dans notre exemple, on peut donc récupérer la réponse aux questions dans la variable `$_POST`:

```
$choice = $_POST['choice'];  
$choice2 = $_POST['choice2'];
```

- Dans notre exemple, nous avons deux variables PHP:
 `$choice` et `$choice2`
- Pour les autres array super-globaux, voir la documentation php (à propos des variables et des variables prédéfinies, chapitre 7 de la documentation officielle.)

B. Calcul et affichage des résultats

url: Voir: </guides/php/examples/simple-calculateur/calcul.php>

- Le traitement des résultats consiste à additionner le score des deux réponses puis à donner un feed-back en fonction de celui-ci.

```
<?php
```

```
// Récupération des variables du formulaire
```

```
$choice = $_POST['choice'];
```

```
$choice2 = $_POST['choice2'];
```

```
// Ensuite on calcule le score
```

```
$score = $choice + $choice2;
```

```
// Et on donne le résultat en fonction du score obtenu
```

```
echo "<h3>Votre score est de " . $score . "</h3>";
```

```
// Et finalement un commentaire
```

```
if ($score < 3) {
```

```
    echo "<p>Vous &ecirc;tes un d&eacute;butant</p>";
```

```
} elseif ($score < 5) {
```

```
    echo "<p>Vous avez un niveau moyen</p>";
```

```
} else {
```

```
    echo "<p>Vous &ecirc;tes un expert !</p>";
```

```
}
```

```
?>
```


A retenir:

- On utilise la directive echo pour afficher du HTML.
Les \$variables dans un string "echo" sont substituées par PHP.
- Notez comment générer du HTML conditionalisé (le message est différent selon le score de obtenu).

C. Empêcher l'accès direct à cette page PHP (sans données)

- (1) `if (isset($_POST['choice'])) then { } else { echo "désolé"; }`
- (2) Alternativement: `if (!isset($_POST['choice'])) {echo "désolé"; exit; }`

2.2 Traitement de formulaires avec PHP II

Exemple 2-2: Checkboxes avec PHP - arrays

[url: /guides/php/examples/simple-calcul4/formulaire4.text](#)

[url: /guides/php/examples/simple-calcul4/formulaire4.html](#)

Voici un extrait du formulaire html:

```
<form action="calcul4.php" method=post>
Quels sont vos couleurs préférées?
<br>
<input type="checkbox" name="choice[]" value="Red">Red
<table bgcolor="red" width="50"><tr><td>&nbsp;</td></tr></table>

<input type="checkbox" name="choice[]" value="Blue">Blue
<table bgcolor="blue" width="50"><tr><td>&nbsp;</td></tr></table>

<input type="checkbox" name="choice[]" value="Green">Green
<table bgcolor="green" width="50"><tr><td>&nbsp;</td></tr></table>
.....
<input type="checkbox" name="choice[]" value="Black">Black
<table bgcolor="black" width="50"><tr><td>&nbsp;</td></tr></table>

<input type="submit" value="Voir le résultat!">
</form>
```

- Notez bien la syntaxe pour stocker les choix dans un array: "choice[]"

Voici le code php qui traite le formulaire.

```
<?php
$choice = $_POST['choice'];

echo("<h3>Vos couleurs préférées sont </h3>");

for ($i=0;$i<sizeof($choice);$i++) {
    if (isset($choice[$i])) {
        echo("$choice[$i] - ");
    }
}
?>
```

Exemple 2-3: Checkboxes avec PHP - multiples variables

- L'exemple 2-2 "Checkboxes avec PHP - arrays" [10] ci-dessus montre comment efficacement traiter un série de checkboxes
- L'exemple ici montre comment faire la même chose d'un façon **inefficace**. On crée une variable pour chaque checkbox

[url: /guides/php/examples/simple-calculate/formulaire3.text](#)

[url: /guides/php/examples/simple-calculate/formulaire3.html](#)

Voici un extrait du code HTML:

```
<form action="calcul3.php" method=post>
Quels sont vos couleurs préférées?
<input type="checkbox" name="choice1" value="Red">Red
<table bgcolor="red" width="50"><tr><td>&nbsp;</td></tr></table>

<input type="checkbox" name="choice2" value="Blue">Blue
<table bgcolor="blue" width="50"><tr><td>&nbsp;</td></tr></table>
.....
<input type="checkbox" name="choice6" value="Black">Black
<table bgcolor="black" width="50"><tr><td>&nbsp;</td></tr></table>
<input type="submit" value="Voir le résultat!">
</form>
```

Voici un extrait du code PHP:

```
<?php
echo "<h3>Vos couleurs préférées sont </h3>";

if (isset($_POST['choice1'])) {echo $_POST['choice1'] . " - ";}
if (isset($_POST['choice2'])) {echo $_POST['choice2'] . " - ";}
if (isset($_POST['choice3'])) {echo $_POST['choice3'] . " - ";}
if (isset($_POST['choice4'])) {echo $_POST['choice4'] . " - ";}
if (isset($_POST['choice5'])) {echo $_POST['choice5'] . " - ";}
if (isset($_POST['choice6'])) {echo $_POST['choice6'];}
?>
```

2.3 Traitement de formulaires avec PHP III

Exemple 2-4: Prix calcul d'une voiture avec un seul fichier PHP

url: voir: </guides/php/examples/calculate-demo/prix-bagnole.php>

url: source: </guides/php/examples/calculate-demo/prix-bagnole.php>

- Cet exemple montre:
 - comment traiter un formulaire HTML avec du code dans la même page.

A. Le formulaire HTML (seulement une partie):

```
<FORM METHOD="POST" ACTION="<? echo $_SERVER['PHP_SELF'] ?>">
    .....
    <br>
    Entrez le cout de votre assurance/ Insurance<br>
    <input type="text" name=assurance size=8 value=0><br>

    Entrez le prix que vous ont coûté vos accidents / Accidents<br>
    <input type="text" size=8 name=accidents value=0><br>
    .....
    Entrez le coût de vos amendes ... / fines<br>
    <input type="text" size=8 name=amendes value=0><br>
    Votre Nom / Your name
    <input type="text" size=40 name=nom><br>
    <HR width=250><BR>
    <input type="submit" value="Alors / Let's see !?!" name="process">
</FORM>
</center>
```

B. A retenir (pour la suite)

- ACTION="<? echo \$_SERVER['PHP_SELF'] ?>"
dit qu'il faut poster le résultat vers le même fichier (`$_SERVER` est un array super-global qui contient les variables du server. L'index '`PHP_SELF`' contient l'url du fichier courant).
- C'est la seule chose qui change par rapport à l' exemple 2-1 "Un simple quiz et POST vers un autre fichier PHP" [5]
- Notez `name="process"` dans le "submit".
Cette variable nous sera utile plus tard.

C. Comment traiter la requête venant d'une page par la même page ?

- d'abord ce n'est pas une nécessité
(on aurait pu appeler une autre page PHP)
- il suffit de tester si une variable définie implicitement par le formulaire est présente. Dans notre cas c'est \$process:

```
<?php
if (!isset($_POST['process'])) {
?>
//... ici on affiche le formulaire (directement en HTML)
//    uniquement si $process = FALSE.

    <FORM METHOD="POST" ACTION="<? echo $PHP_SELF ?>">
    .....
    </FORM>

?>
else {
//... ici on effectue les calculs
}
?>
```

A retenir:

- on peut même “mixer” du HTML à l'intérieur d'une instruction PHP
- Le “if” fonctionne un peu près comme tous les langages “C-like”.
A la place du “if - then - else” on aurait pu utiliser un simple “if” plus “exit”, voir l'exemple 4-1
“Questionnaire et résultats dans fichier” [23]

D. Calcul et affichage des résultats

Voici le code (sans la phase de récupération des variables):

```
// calculate the cost per month
$cost = round(((((((($accidents + $assurance) + (((($consokilo / 100.0) * $kilomois) *
$prixcarbu) * 12.0)) + $vignette) + $tcs) + $autoroutes) + $entretien) + $amendes) /
12.0);

echo "<h2>Resultat/Result</h2>";

// print the name if we got one
if ($nom) { echo "$nom, votre "; } else { echo "Votre ";}

// print the results
echo "bagnole vous coutera environ $cost francs par mois / Your car costs about $cost
Swiss francs/month.<p>";

// give a short comment
if ($cost < 10) {
    $evaluation = "Heh c'est pas sérieux / Mhh this doesn't look serious !";
}
elseif ($cost < 400.0) {
    $evaluation = "Vous en sortez bien / This is fine :)";
}
else {
    $evaluation = "Vous ne vous en sortez pas bien, pensez aux transports communs / You
pay quite a lot!";
}
echo "$evaluation <p>";
```


E. A retenir:

- Ici on ne mixe pas HTML avec PHP (on aurait pu).
- On utilise l'instruction echo pour afficher du HTML.
Les \$variables dans un string sont substituées par PHP.
- Notez comment générer du HTML conditionné à 2 endroits
(pour afficher le nom si le champs a été rempli et pour donner un commentaire sur le coût).

F. Si c'était trop compliqué:

Voir la version intégrée de l' exemple 2-1 "Un simple quiz et POST vers un autre fichier PHP"
[5]:

Exemple 2-5: Un simple quiz et POST vers le même fichier PHP

[url: Voir: /guides/php/exemples/simple-calculateur/form-calcul.php](#)

[url: Source: /guides/php/exemples/simple-calculateur/form-calcul.text](#)

2.4 Tester l'existence de variables et valeurs POST/GET

Dans PHP vous pouvez tester si une variable "POST/GET" existe ou si elle a une valeur non-zéro ou non-vidé (string):

2 méthodes pour tester si une entrée existe dans un array \$_POST ou \$_GET

1. "array_key_exists()"

```
if (array_key_exists('first', $_POST)) { .... faire ...};
```

2. "isset()" pour voir si une variable existe:

- utiliser pour décider s'il faut afficher un formulaire ou traiter le formulaire

```
if (isset($_POST['first']) ) { .... faire ....}
```

La différence est que `array_key_exists` retourne TRUE même quand la valeur est NULL alors que `isset` retourne FAUX.

ATTENTION, Si vous utilisez des `<input type="text">` il faut aussi tester si la variable en question est vide car elle existera avec les tests ci-dessus.

3. "empty()" pour tester si une variable est zéro, vide **ou** inexistante

- utiliser pour décider si l'utilisateur a rempli un champs

```
if (empty ($input) ) { ... ne rien faire} else { ... faire ...}
```

Donc `empty()` retourne TRUE si la valeur est soit: "", 0, "0", NULL, FALSE, array(),

2.5 Gestion de sessions

- PHP permet de gérer des sessions, c.a.d. de préserver certaines données le long d'accès successifs à une page.
- Chaque visiteur reçoit un identificateur unique (la "sessions id"). Cet id est soit stocké dans un cookie (dans le client) ou alternativement propagé par l'URL.
- On peut stocker un nombre illimité de variables dans le array `$_SESSION`

Exemple 2-6: Limiter l'accès répétée à une page

url: <http://tecfa.unige.ch/guides/php/examples/sessions/>

```
session_start();
if (!isset($_SESSION['count'])) {
    $_SESSION['count'] = 0;
} else {
    $_SESSION['count']++;
}
if ($_SESSION['count'] > 2) {
echo '<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://
/www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">';
    echo "<html> <body>";
    echo "Désolé c'est terminé, vous ne pouvez pas envoyer 2 fois le meme
questionnaire";
    echo "</body> </html>";
    exit;
}
// .... continuer ici le code qui gère l'accès 1 et 2
```

3. Annotation d'une page (écrire dans un fichier et inclure)

- Il s'agit d'un simple système d'annotation
- Il existe des solutions mysql/php plus utiles (mais plus compliquées)

Exemple 3-1: Un simple système d'annotation

Usage:

- Créer une page php avec un contenu (ici: forum.php)
- Inclure le module "annotate.php" (voir page suivante!)
- Créer un fichier *.php.comment (ici: forum.php.comment)
 - dans lequel le module va écrire les commentaires
 - sur certains systèmes ce fichier doit être "writable" par le serveur (uid=nobody ou apache, ou autoriser tout le monde à écrire (mode 666).
 - à Tecfa le répertoire/fichier doit être à vous (configuration php = safe_mode)

Le fichier exemple

[url: http://tecfa.unige.ch/guides/php/examples/forum/forum.php](http://tecfa.unige.ch/guides/php/examples/forum/forum.php)

[url: http://tecfa.unige.ch/guides/php/examples/forum/forum.text](http://tecfa.unige.ch/guides/php/examples/forum/forum.text)

Voici un texte Il peut être aussi long ou compliqué que vous voulez !

```
<hr>
```

```
<? require("annotate.php"); ?>
```

Le module d'annotation

- Renommez ce fichier *.text en fichier *.php pour l'inclusion dans votre page.

[url: http://tecfa.unige.ch/guides/php/examples/forum/annotate.text](http://tecfa.unige.ch/guides/php/examples/forum/annotate.text)

```
$secret = "zap";
```

```
$mode = "a";

// the module can't be called by itself
if ( basename($_SERVER['PHP_SELF']) == "annotate.php" ) {
    exit;
}

// if the password is correct we overwrite
if ($password == $secret) {
    $mode = "w";
    $fp = fopen (basename($PHP_SELF) . ".comment", $mode);
    fwrite ($fp, $message);
    fclose ($fp);
}

// if there is a message we append it to a file called file.comment
// make sure that file.comment is writeable by the server)
if ($message) {
    /* uncomment the next two lines to strip out html from input */
    /* $name = strip_tags($name); */
    /* $message = strip_tags($message); */
    $name = StripSlashes($name);
    $message = StripSlashes($message);
    $message = ereg_replace("\n\n", "\n<P>", $message);
    $date = date("l, F j Y, H:i");
    $message = "<B>$name </B> -- $date<P> $message <BR><HR>";
    $fp = fopen (basename($PHP_SELF) . ".comment", $mode);
    fwrite ($fp, $message);
    fclose ($fp);
    @readfile(basename(($PHP_SELF . ".comment")));
}
```

A retenir:

- Pour écrire dans un fichier, il faut:
 - que le fichier appartienne à vous, ou encore à “nobody” (le serveur tourne sous ce nom) ou alternativement qu’il soit “world writeable” (chmod 666)
 - `fopen(<nom du fichier>, mode)`
ouvre un fichier et retourne un “handle”
 - `fwrite (<handle>, “string”)`
permet d’écrire (il existe d’autres fonctions pour cela)
 - `fclose (<handle>)`
ferme le fichier (IMPORTANT)
 - `@readfile (<file>)`
insère le contenu d’un fichier

Attention

- Ce type de script peut présenter des graves problèmes de sécurité !
- Dès que vous mettez un champs libre (input text ou textarea) à disposition, il y aura des centaines des messages de spam. Il faut donc limiter l’accès au scripts.
- ***En aucun cas il faut utiliser la fonction `include()` au lieu de `readfile()`***

4. Questionnaires on-line et récupération dans un fichier

Exemple 4-1: Questionnaire et résultats dans fichier

url: voir: <http://tecfa.unige.ch/guides/php/examples/form-file-demo/>

- le fichier new-entry.php contient un formulaire et le code pour rajouter
- le fichier dump_results.php affiche le contenu du fichier

A. le formulaire

- posté et traité avec la même méthode rencontré dans la section 2.3, p. 13
- par contre une partie du HTML est générée par une fonction PHP:

```
$scales = array("food", "work", "love", "leisure", "sports");
function scale ($thing) {
    echo "<TR> <TD align=right>Importance of <STRONG>$thing</STRONG>:</TD> ";
    echo "<TD><select name=$thing>";
    echo "<option value=1>1 - totally unimportant";
    echo "<option value=2>2 - not important";
    echo "<option value=3 selected>3 - rather not important";
    echo "<option value=4>4 - slightly important";
    echo "<option value=5>5 - rather important";
    echo "<option value=6>6 - very important";
    echo "</select>";
    echo "</TD></TR> ";
}
```

```
function dump_scales () {
    global $scales;
    reset($scales);
    do {
        $scale = scale(current($scales));
        echo "$scale\n";
    }
    while (next($scales));
} ?>
<form> <table>
.....
dump_scales();
.....
</table> </form>
Ecrire dans un fichier
// check existance of file (or try to create it)
// a better alternative to touch() would be is_file, is_writable and so on.
$try = touch($file_name);
if (!$try) {
    echo "<p>Sorry I can't open a file, something is wrong";
    exit;
}

// this is the stuff we get from the form, we insert it into an array
$input = array ($login, $password, $fullname, $url, $food, $work, $love,
$leisure, $sports);

// so we can make a big string with tabs between the elements
// note that we add a \n (line break) to the end of the string.
$output_line = implode ($input, " ")."\n";
```



```
// Now open the file (get a file pointer)
// We will append to it and therefore use the "a" option
$output_stream = fopen($file_name, "a");
// and dump the string into the file
$result = fputs ($output_stream, $output_line);

// give feedback
if ($result) {
    echo "<p>Your data have successfully been registered.";
}
else {
    echo "<p>Too bad, the db did not want your data.";
}
// close the file pointer
fclose($output_stream);
?>

<?
// EXIT here ... we don't want to see the form again. If you do, kill the exit
exit;
}
?>
```

A retenir:

- Voir aussi: exemple 3-1 "Un simple système d'annotation" [20]
- `fputs(<handle>, "string")`
permet d'écrire (il existe d'autres fonctions pour cela)

B. Afficher le contenu d'un fichier

.... on ne se fatigue pas: le tout dans un <pre> avec un "include"

```
<BODY>
  <H1>Dump Database Contents</H1>

  <?
    /* Daniel.Schneider@tecfa.unige.ch
    Will dump the contents of the results file
    */
  ?>

  <strong>Results registered so far:</strong>
  <pre>
  <? readfile("results/result.text"); ?>
  </pre>
  .....

</BODY>
```

Attention

- Ce genre de script peut attirer les vendeurs de viagra de la planète entière ...
- **INTERDICTION TOTALE** d'écrire dans un fichier *.php ou *.html. Ce fichier doit s'appeler *.text !!!

5. Autres format que HTML

Principe: vous devez dans la PREMIERE ligne du script définir le content-type:

- Exemple: `Header("Content-type: image/gif");`

A. Exemple de génération VRML

- planter 100 arbres sans se fatiguer

url: voir: </guides/php/exemples/vrml-temple/mixing/>

```
<? Header("Content-type: model/vrml");  
    echo "#VRML V2.0 utf8"; ?>
```

```
PROTO Tree .....
```

```
Transform {  
    translation -5 0 -10  
    children [  
    <?  
    for ($i=0; $i<10; $i++) {  
        for ($j=0; $j<10; $j++) {  
            echo "Tree { translation $i 0 $j }";  
        }  
    }  
    ?>  
    ] }  
}
```

B. Exemple de génération d'images PNG

url: voir: </guides/php/exemples/button>

6. Conseils pour le débogage

(ceci est important !!!)

Affichage de tous les avertissements et erreurs

- Si votre serveur est configuré pour ne pas afficher les simples warnings (par exemple utilisation de variables vide et non-initialisées), vous êtes conseillés de demander à PHP d'afficher un maximum des erreurs
- Insérer au **début de votre code** la ligne suivante !

```
error_reporting(E_ALL);
```

Informations

- Si vous voulez connaître toutes les variables créées par le système auxquelles vous avez accès (y compris les variables/names transmis depuis un formulaire), insérez qq part dans le fichier:

```
phpinfo();
```