

PHP

CM2-1 : PHP « intermédiaire »

Mickaël Martin Nevot

V4.3.0



Cette œuvre de [Mickaël Martin Nevot](#) est mise à disposition selon les termes de la [licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Partage à l'Identique 3.0 non transposé](#).

PHP

- I. Présentation
- II. PHP I
- III. XML
- IV. Regexp
- V. **PHP II**
- VI. MySQL
- VII. POO
- VIII. PDO
- IX. Hacking
- X. PHP « avancé »

Structure de langage

- **Structures de langage :**

≠ fonction

- echo, return, include, require, etc.

- Utilisation de parenthèses non conseillée (même avec plusieurs arguments) :

- Gain de performances :

```
echo $a; // Plutôt que echo ($a);
```

```
echo '1, ', '2'; // Plutôt que echo '1, ' . '2';
```

- Erreur en cas de plusieurs arguments :

```
// Cause une erreur.
```

```
echo ($a, $b);
```

- Conversion en expression :

```
// Erreur vous voulez retourner une référence !
```

```
return ($a);
```

Retour sur return !

- Depuis une fonction : termine et retourne l'argument
- Depuis un script inclus : retourne par la fonction d'inclusion au script appelant
- Depuis le script principal : arrêt du script



Quelques fonctions

- Teste la **définition d'une variable** :

- `isset($var_1, $var_2, ...)`

`!isset($var) || $var == false`

- Teste si une variable est vide : `empty($var)`

- Crée un **MD5** : `md5($str, ...)`

- Génère un **identifiant unique** : `uniqid($prefix = ..., ...)`

- Génère un **nombre aléatoire** : `mt_rand(...)`

- Envoie un **en-tête HTTP** : `header($str, ...)`

- Redirection du navigateur :

- `$str = 'Location: ' . $url`

- Initialise une **session** : `session_start()`

Pas d'affichage avant

Important !

Avant d'utiliser un header

GET et POST

- HTTP (*hypertext transfer protocol*) :
 - Protocole de communication (client-serveur)
 - Méthodes (commandes) :
 - **GET** : construction d'URL dynamique
 - **POST** : passage « caché » des valeurs

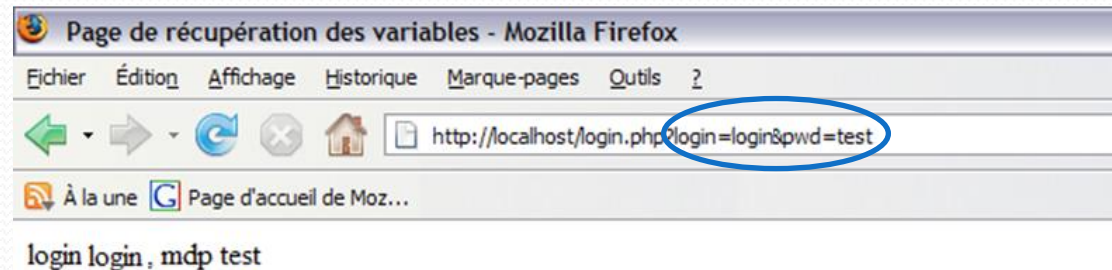
URL (*uniform resource locator*) : littéralement « localisateur uniforme de ressource », est une chaîne de caractères utilisée pour adresser les ressources du Web, aussi appelée **adresse Web**

- Variables « superglobales » : `$_GET`, `$_POST`
 - Récupération de valeurs passées via le protocole HTTP et les méthodes **GET** et **POST**

GET et POST

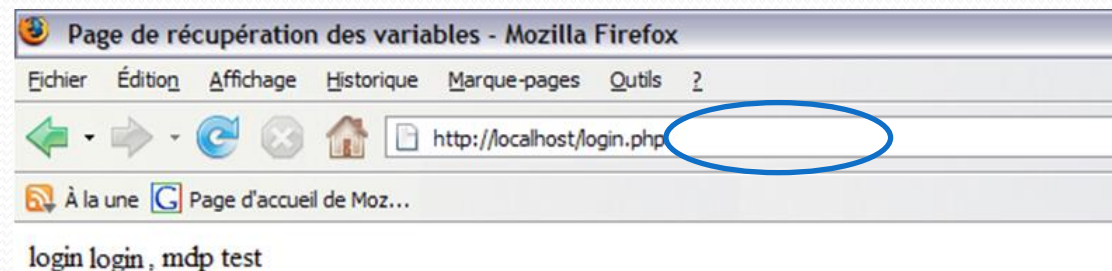
- **GET :**

```
if (isset($_GET['login'], $_GET['pwd']))
{
    echo 'login ', $_GET['login'], ', mdp ', $_GET['pwd'];
}
```



- **POST :**

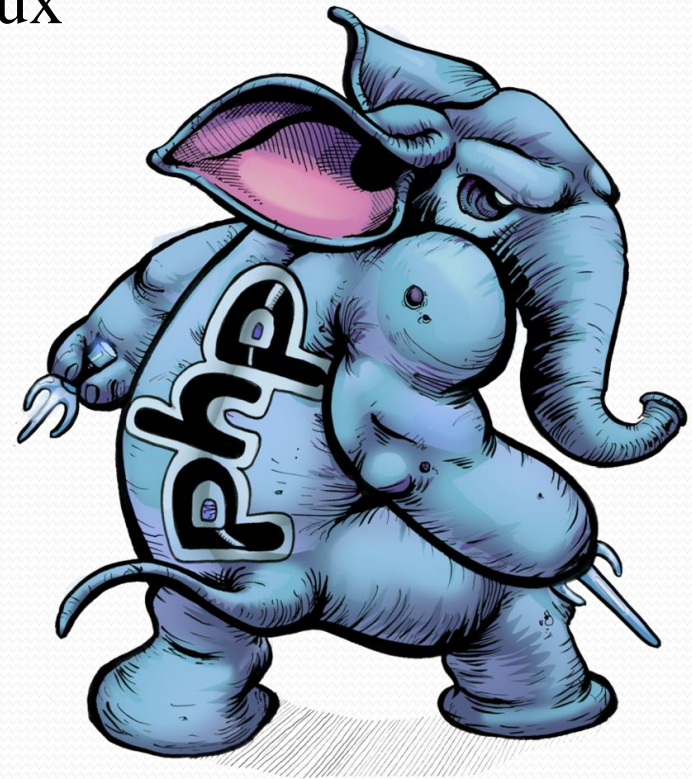
```
if (isset($_POST['login'], $_POST['pwd']))
{
    echo 'login ', $_POST['login'], ', mdp ', $_POST['pwd'];
}
```



GET et POST

- Il est possible de passer des tableaux

```
<input type="text" name="arr[foo]" value="1">
<input type="text" name="arr[bar]" value="2">
...
$arr = $_POST['arr'];
/*
array (
    'foo' => '1',
    'bar' => '2',
);
*/
```



Attention aux *gotchas* ! Certains caractères spéciaux dans l'index seront convertis afin de pouvoir être transmis selon la méthode choisie

E-mail

- Fonction :

- `mail($to, $subject, $message, $headers = ..., ...)`

- Envoi d'un *e-mail* :

- Destinataire(s) : `$to`
 - Sujet du mail : `$subject`
 - Message à envoyer : `$message`
 - Texte simple
 - HTML
 - Texte simple et HTML (multi-format)
 - En-têtes : `$headers`



E-mail multi-format

- *E-mail* en texte **simple** :
 - Toujours lisible : compatibilité totale
 - Très léger
- *E-mail* en **HTML** :
 - Mise en forme complexe (police, image, lien, etc.)
- *E-mail* **multi-format (meilleure solution)** :
 - Affichage en texte simple en cas d'incompatibilité HTML
 - Affichage en HTML autrement

En-tête et frontière

- En-tête :

Peut être fausse

- Adresse de l'expéditeur : `'From: exp1@fai.fr'`
- Adresse de réponse : `'Reply-To: exp2@fai.fr'`
- Destinataire(s) en copie : `'Cc: email_1, email_2'`
- Destinataire(s) en copie cachée : `'Bcc: email_1, email_2'`
- *Internet media type* : `'Content-type: type; sous-type'`

- Frontière (*boundary*) :

- Permet de faire la séparation texte simple / HTML
- *Content-type* de l'entête :
`'Content-type: multipart/alternative; boundary=frontière'`
- Au début de chaque type de message :
 - Frontière préfixée de deux tirets accolés
 - Après la frontière, modification des *headers* appliqués localement (au moins le *content-type* du message)

Exemple d'e-mail

```
$to = 'to_mail@fai.org';
$from = 'my_mail@fai.org';
$reply = 'reply_mail@fai.org';
$subject = 'Test mail';

$boundary = md5(uniqid(mt_rand()));
$headers = 'From: Name <' . $from . '>' . "\n";
$headers .= 'Return-Path: <' . $reply . '>' . "\n";
$headers .= 'Content-type: multipart/alternative; boundary="' . $boundary . '"';

$message_text = 'Hello world!';
$message_html = '<html><body><strong>Hello world!</strong></body></html>';
$message = '--' . $boundary . "\n";
$message .= 'Content-Type: text/plain; charset=utf-8' . "\n\n";
$message .= $message_text . "\n\n";
$message .= '--' . $boundary . "\n";
$message .= 'Content-Type: text/html; charset=utf-8' . "\n\n";
$message .= $message_html . "\n\n";

mail($to, $subject, $message, $headers);
```

Fichier

- **Ouverture** : `fopen($file, $mode, ...)`

- Fichier à ouvrir : `$file`

renvoie un pointeur sur le fichier



- Spécifie le type d'accès désiré au fichier : `$mode`

- Ouvre un fichier à son début : `'r', 'r+'`

- Ouvre un fichier à sa fin : `'a', 'a+'`

- Crée/écrase et ouvre un fichier : `'w', 'w+'`

- Crée et ouvre un fichier : `'x', 'x+'`

- **Fermeture** : `fclose($handle)`

Ouvre en lecture dans tous les cas et en plus en écriture en cas de présence de + dans le mode



Fichier

- **Lecture** (jusqu'à `$length` octets du fichier `$handle`, ou jusqu'à la fin de la ligne courante par défaut pour `fgets(...)`):
 - `fread($handle, $length)` ← Nombre de caractères à lire
 - `fgets($handle, $length = ...)`
- **Écriture** : Valeur retournée par `fopen()`
 - `fwrite($handle, $str, ...)` ← `fputs(...)` est un alias de `fwrite(...)`
 - `fputs($handle, $str, ...)` ← Chaîne de caractères à écrire
- Test si fichier/répertoire existe : `file_exists($filename)`
- Copie du fichier : `copy($source, $dest, ...)`
- Renomme un fichier/répertoire : `rename($old, $new, ...)`
- Efface un fichier : `unlink($filename)`

Chargement HTML d'un fichier

- Le nom original du fichier (sur la machine du client) :

```
$_FILES['variable']['name']
```

- Le nom temporaire du fichier (sur le serveur) :

```
$_FILES['variable']['tmp_name']
```

- *L'Internet media type* du fichier :

```
$_FILES['variable']['type']
```

- La taille du fichier (en octets) :

```
$_FILES['variable']['size']
```

- L'éventuel code d'erreur :

```
$_FILES['variable']['error']
```

```
<form action="send_fichier.php" method="post" enctype="multipart/form-data">
  <p>
    Votre fichier : <input type="file" name="mon_fichier"></input>
    <input type="hidden" name="MAX_FILE_SIZE" value="20000"></input>
  </p>
</form>
// ...
move_uploaded_file($_FILES['variable']['tmp_name'],
    $uploads_dir . '/' . $_FILES['variable']['name']);
```

Variables dynamiques

- Nom de variable affecté et utilisé dynamiquement :

```
$var = 'hello';  
$hello = 'world';  
echo ${$var}; // Affiche : world.
```

...

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

```
function insert() { // Code. }  
function delete() { // Code. }  
function update() { // Code. }  
function select() { // Code. }
```

```
if ('insert' == $champ_cache  
|| 'delete' == $champ_cache  
|| 'update' == $champ_cache  
|| 'select' == $champ_cache)  
{  
    $champ_cache();  
}
```



Crédits

Auteur

Mickaël Martin Nevot

mmartin.nevot@gmail.com



Carte de visite électronique

Relecteur

- Christophe Delagarde
(christophe.delagarde@univ-amu.fr)
- Pierre-Alexis de Solminihac (pa@solminihac.fr)

Cours en ligne sur : www.mickaël-martin-nevot.com

