

```
<?php
```

```
phpinfo();
```

```
/* fabien.hermenier@unice.fr */
```

```
?>
```

Rappel

serveur HTTP

**écoute un port et réponds à des requêtes HTTP
envoyées par un navigateur**

GET /index.html HTTP/1.0

le serveur cherche index.html sur son disque
et l'envoie

POST, DELETE, ...

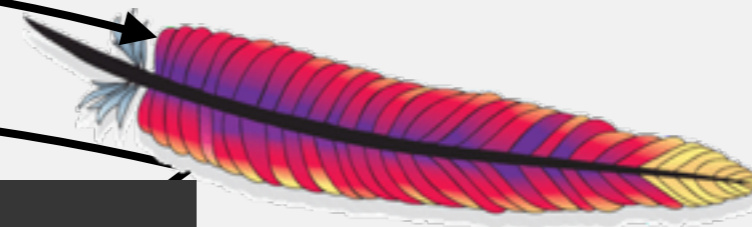
1 milliard de pages chez facebook ?



: Hypertext Preprocessor

- langage compilé à la volée
- produit du contenu (HTML, images, ...) dynamiquement au travers d'un serveur HTTP
- 20 ans d'existence
- 6ème langage le plus utilisé (1er pour le Web)

GET /foo.php HTTP/1.0



Apache
http server



/var/www/foo.php

```
...  
<body>  
Les nombres de 1 à 10:  
<ul>  
<li>1</li><li>2</li><li>3</li>  
</ul>  
</body>  
</html>
```

```
...  
<body>  
Les nombres de 1 à 10:  
<ul>  
<?php  
for ($i = 1; $i <= 3; $i++) {  
    echo «<li>$i</li>»;  
}  
?>  
</ul>  
</body>  
</html>
```



...

<h1>Quelques factorielles</h1>

```
<?php
/**
 * Calcul le factoriel du nombre en argument
 */
function factoriel($n) {
    if ($n == 0) {
        return 1;
    } else {
        return $n * factoriel($n - 1);
    }
}

echo "<ul>";
for ($i = 0; $i <= 5; $i++) {
    $res = factoriel($i);
    echo "<li>Le factoriel de $i est $res</li>";
}
echo "</ul>";
?>
```

...



Quelques factorielles

- Le factoriel de 0 est 1
- Le factoriel de 1 est 1
- Le factoriel de 2 est 2
- Le factoriel de 3 est 6
- Le factoriel de 4 est 24
- Le factoriel de 5 est 120

Les types

scalaires

entier, réel, string, booléen

composés

tableau, objet

spéciaux

ressources, NULL

```
<?php
$x = 3;
$y = $x * 2 + 5;
$x = 'foo';
$z = '5';
$y = $z - 3;
?>
```

Typage dynamique !

Les tableaux

usuels

indexé par un ensemble d'entiers

```
$tab = array();  
$tab = array(1, 'toto', 7);  
echo $tab[0]; //affiche 1  
$tab[0] = 'bar';  
$tab[2453] = 42;
```

associatifs

indexé par une chaîne de caractère

```
$tab = array('charlie' => 'chaplin',  
            'homer' => 'simpsons');
```

Les tableaux sont extensibles et peuvent avoir des trous

Manipuler les tableaux

(php.net/manual/fr/ref.array.php)

```
<?php
$notes = array(5,7,8,10);
echo count($notes); //affiche 4

$notes[] = 15;
echo $notes[4]; //affiche 15;

unset($notes[0]);
echo $notes[0]; //affiche 7

$mails = array('fabien' => 'fabien.hermenier@unice.fr');
$mails['dino'] = 'dino.lopez@unice.fr';
echo array_key_exists('fabien', $mails); //true

?>
```

Les chaînes de caractères

(php.net/manual/fr/ref.strings.php)

```
<?php
$var = "world";
echo "hello $var<br/>";
echo 'hello $var<br/>';
echo "hello '$var'<br/>";
echo "hello \"$var\" <br/>";

echo "longueur: " . strlen($var) . "<br/>";

echo "Hello " . strtoupper($var[0]) . substr($var, 1);
?>
```

```
hello world
hello $var
hello 'world'
hello "world"
longueur: 5
Hello World
```



Les boucles

```
<?php
function avgwhile($t) {
    $s = 0;
    $i = 0;
    while ($i < count($t)) {
        $v = $t[$i];
        $s += $v;
        $i++;
    }
    return $s / count($t);
}

function avgFor($t) {
    $s = 0;
    for ($i = 0; $i < count($t); $i++) {
        $s += $t[$i];
    }
    return $s / count($t);
}

function avgForEach($t) {
    $s = 0;
    foreach ($t as $v) {
        $s += $v;
    }
    return $s / count($t);
}
?>
```

```
<?php
$arr = array();
$arr[0] = 1;
$arr[1] = 2;
$arr[3] = 3;

echo 'while: ' . avgwhile($arr) . '<br/>';
echo 'for: ' . avgFor($arr) . '<br/>';
echo 'foreach: ' . avgForEach($arr) . '<br/>';
?>
```



Les boucles

```
<?php
function avgwhile($t) {
    $s = 0;
    $i = 0;
    while ($i < count($t)) {
        $v = $t[$i];
        $s += $v;
        $i++;
    }
    return $s / count($t);
}

function avgFor($t) {
    $s = 0;
    for ($i = 0; $i < count($t); $i++) {
        $s += $t[$i];
    }
    return $s / count($t);
}

function avgForEach($t) {
    $s = 0;
    foreach ($t as $v) {
        $s += $v;
    }
    return $s / count($t);
}
?>
```

```
<?php
$arr = array();
$arr[0] = 1;
$arr[1] = 2;
$arr[3] = 3;

echo 'while: ' . avgwhile($arr) . '<br/>';
echo 'for: ' . avgFor($arr) . '<br/>';
echo 'foreach: ' . avgForEach($arr) . '<br/>';
?>
```

while: 1
for: 1
foreach: 2



Les boucles - II

```
<?php
function annuaire($adr) {
    echo '<u>';
    foreach ($adr as $k => $v) {
        echo "<li>$k: <a href='mailto:$v'>$v</a></li>";
    }
    echo '</u>';
}

$mails = array();
$mails['dino'] = 'dino.lopez@unice.fr';
$mails['marc'] = 'gaetano@polytech.unice.fr';
$mails['ivan'] = 'logre@polytech.unice.fr';
$mails['fabien'] = 'fabien.hermenier@unice.fr';

annuaire($mails);
?>
```

- dino: dino.lopez@unice.fr
- marc: gaetano@polytech.unice.fr
- ivan: logre@polytech.unice.fr
- fabien: fabien.hermenier@unice.fr



Mal organiser son code

```
...  
<h1> Nombres pairs de 0 à 1000</h1>  
  
<?php  
echo '<u1>';  
for ($i = 0; $i < 1000; $i++) {  
    if ($i % 2 == 0) {  
        echo "<li>$i</li>";  
    }  
}  
echo '</u1>';  
</body>  
</html>
```

présentation

m
é
r
e
i
e

NON lisible
composable
ré-utilisable

Mieux organiser son code

```
...
<?php
function nbPairs($m) {
    $nbs = array();
    for ($i = 0; $i < $m; $i++) {
        if ($i % 2 == 0) {
            $nbs[] = $i;
        }
    }
    return $nbs;
}

function html_liste($vals) {
    echo '<ul>';
    foreach ($vals as $v) {
        echo "i>$v</li>";
    }
    echo '</ul>';
}
?>
<h1> Nombres pairs de 0 à 1000</h1>
<?php
html_liste(nbPairs(1000));
?>
...
```

Bien organiser son code

avec des bibliothèques ré-utilisables

inc/numbers.inc:

```
<?php
function nbPairs($m) {
    ...
}
?>
```

inc/pres.inc:

```
<?php
function html_liste($val) {
    ...
}
?>
```

pairs.php

```
<?php
    require_once('inc/numbers.inc');
    require_once('inc/pres.inc');
?>
...
<h1> Nombres pairs de 0 à 1000</h1>
<?php
html_liste(nbPairs(1000));
?>
...
```


TOP

5

(des erreurs)

Mozilla Firefox

Nouvel onglet

file:///home/fhermeni/login.php

Google

Ouverture de login.php

Vous avez choisi d'ouvrir :

login.php
 qui est un fichier de type : script PHP (574 octets)
 à partir de : /home/fhermeni

Que doit faire Firefox avec ce fichier ?

Ouvrir avec Navigateur Web Firefox (défaut)

Enregistrer le fichier

Toujours effectuer cette action pour ce type de fichier.

Annuler OK

Downloads

Music

Public

Templates

login.php

Tools-9.2.2-33.tar.gz

```

du: i
du: i
n acc
du: i
n non
du: i
n acc
du: i
du: i
cordé
du: i
dée
du: impossible de lire le répertoire «./proc/25686/ns»: Perm
du: impossible d'accéder à «./proc/25792/task/25792/fd/3»: Auc
ier de ce type
du: impossible d'accéder à «./proc/25792/task/25792/fdinfo/3»: Aucun fichier ou
dossier de ce type
du: impossible d'accéder à «./proc/25792/fd/3»: Aucun fichier ou dossier de ce t
ype
du: impossible d'accéder à «./proc/25792/fdinfo/3»: Aucun fichier ou dossier de
ce type
du: impossible de lire le répertoire «./root»: Permission non accordée
6,3G
fhermeni@ubuntu:/$
  
```

« login.php » sélectionné (574 octets)

```
function nbPairs($m) {
    $nbs = array();
    for ($i = 0; $i < $M; $i++) {
        if ($i % 2 == 0) {
            $nb[] = $i;
        }
    }
    return $nbs;
}
```

```
<?php
    require_once('inc/numbers.inc');
    require_once('inc/pres.inc');
?>
...
<h1>Nombres pairs de 0 à 1000</h1>
<?php
html_liste(nbPairs(1000));
?>
...
```

Nombres pairs de 0 à 1000



```
function nbPairs($m) {
    $nbs = array();
    for ($i = 0; $i < $M; $i++) {
        if ($i % 2 == 0) {
            $nb[] = $i;
        }
    }
    return $nbs;
}
```

```
<?php
    require_once('inc/numbers.inc');
    require_once('inc/pres.inc');
?>
...
<h1>Nombres pairs de 0 à 1000</h1>
<?php
html_liste(nbPairs(1000));
```



déclaration



```
function nbPairs($m) {
    $nbs = array();
    for ($i = 0; $i < $m; $i++) {
        if ($i % 2 == 0) {
            $nbs[] = $i;
        }
    }
    echo $nbs;
}
```

```
<?php
    require_once('inc/numbers.inc');
    require_once('inc/pres.inc');
?>
...
<h1>Nombres pairs de 0 à 1000</h1>
<?php
html_liste(nbPairs(1000));
?>
...
```

Nombres pairs de 0 à 1000

Array



```
function nbPairs($m) {
    $nbs = array();
    for ($i = 0; $i < $m; $i++) {
        if ($i % 2 == 0) {
            $nbs[] = $i;
        }
    }
    echo $nbs;
}
```

```
<?php
    require_once('inc/numbers.inc');
    require_once('inc/pres.inc');
?>
...
<h1>Nombres pairs de 0 à 1000</h1>
<?php
html_liste(nbPairs(1000));
?>
```

echo != return

Array



```
function html_liste($values) {  
    foreach($values as $v) {  
        echo "$v\n";  
    }  
}
```

```
<?php  
    require_once('inc/numbers.inc');  
    require_once('inc/pres.inc');  
?>  
...  
<h1> Nombres pairs de 0 à 10</h1>  
<?php  
html_liste(nbPairs(10));  
?>  
...
```

Nombres pairs de 0 à 1000

02468




```
function html_liste($values) {
    foreach($values as $v) {
        echo "$v\n";
    }
}
```

```
<?php
    require_once('inc/numbers.inc');
    require_once('inc/pres.inc');
?>
...
<h1> Nombres pairs de 0 à 10</h1>
<?php
html_liste(nbPairs(10));
?>
```

‘ \n ’ ! = ‘
 ’

02468




```
function nomComplet($nom, $prenom) {  
    return $nom + $prenom;  
}  
echo nomComplet('Hermenier', 'Fabien');
```

0



```
function nomComplet($nom, $prenom) {  
    return $nom + $prenom;  
}  
  
echo nomComplet('Hermenier', 'Fabien');
```



Debugger du php

```
function nbPairs($m) {
    $nbs = array();
    for ($i = 0; $i < $n; $i++) {
        if ($i % 2 == 0) {
            $nb[] = $i;
        }
    }
    var_dump($nbs);
    return $nbs;
}
```

```
<?php
    require_once('inc/numbers.inc');
    require_once('inc/pres.inc');
?>
...
<h1> Nombres pairs de 0 à 1000</h1>
<?php
html_liste(nbPairs(1000));
?>
...
```

Nombres pairs de 0 à 1000

array(0) {}



```
$ sudo tail -f /var/log/apache2/error.log
```

```
[Thu Jan 16 01:35:45.161615 2014] [:error] [pid 30110] [client 127.0.0.1:58809] PHP
Notice: Undefined variable: n in /home/fhermeni/public_html/pairs.php on line 5
```

Comment valider le code html + php ?



- pratique pour produire dynamiquement des pages HTML
- langage laxiste demandant de l'attention
- décomposer votre code !

<http://www.php.net/manual/fr/>