

```
<?php
```

```
/* Persistance côté serveur */
```

```
file_get_contents('cours.txt')
```

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

```
?>
```

des données doivent survivre à la fin d'un script

- identification de connexions
- tweets/messages sur un blog
- ...

Comment

stocker
accéder
rechercher



la persistance en PHP

sessions

pour suivre les visiteurs

**systeme de
fichiers**

stockage de données à bas niveau

api XML

manipulation de données structurées (XML)

...

les sessions

`$_SESSION`

- tableau associatif
- un tableau par visiteur
- accessible sur toutes les pages

`session_start()`

- charge la session
- à exécuter **avant** d'avoir générer du code,
(nécessite de modifier le header de la réponse)

`session_destroy()`

- détruit la session

en pratique ...

```
<?php
/* inc/counter.inc */
function incVisit() {
    if (!array_key_exists('nb', $_SESSION)) {
        $_SESSION['nb'] = 0;
    }
    $_SESSION['nb']++;
}

function nbVisits() {
    return $_SESSION['nb'];
}

function cleanCounter() {
    $_SESSION = array();
    session_destroy();
}
?>
```

```
<?php
session_start();

require_once("inc/counter.inc");

incVisit();
echo "Vous êtes venu »;
echo nbVisits()
echo " fois";
if (isset($_GET['stop'])) {
    cleanCounter();
}
?>
```

Manipuler les fichiers

(<http://php.net/manual/fr/ref.filesystem.php>)

à grain fin

fopen,
fread,
fwrite,
fclose,

...

à gros grain

file_get_contents
file_put_contents

```
<?php
$fp = fopen('data2.txt','a');
$cnt = fread($fp, filesize('data2.txt'));
if (!fwrite($fp, $cnt.'1')) {
    echo "Pb de permissions ?";
}
fclose($fp);
?>
```

```
<?php
$cnt = file_get_contents('data2.txt');
$cnt = $cnt."1";
file_put_contents('data2.txt', $cnt);
?>
```

et les répertoires

(<http://fr2.php.net/manual/fr/ref.dir.php>)

à grain fin

`opendir`,
`readdir`,
`closedir`,
`mkdir`,

...

à gros grain

`glob`, `scandir`

```
<?php
$dp = opendir('.');
while (($f = readdir($dp)) !== false) {
    echo 'Fichier: '.$f.'<br/>';
}
closedir($dp);
?>
```

```
<?php
foreach(glob("/*.php") as $f) {
    echo "Fichier $f<br/>";
}
?>
```

Manipuler des données structurées

XML:

```
<?xml version='1.0' encoding='utf-8'?>
<teachers>
  <teacher root="1" email="fabien.hermenier@unice.fr">
    <firstname>Fabien</firstname>
    <lastname>Hermenier</lastname>
    <group>4</group>
    <www>https://sites.google.com/site/hermenierfabien/</www>
    <twitter>fhermeni</twitter>
  </teacher>
  <teacher>...</teacher>
</teachers>
```

JSON:

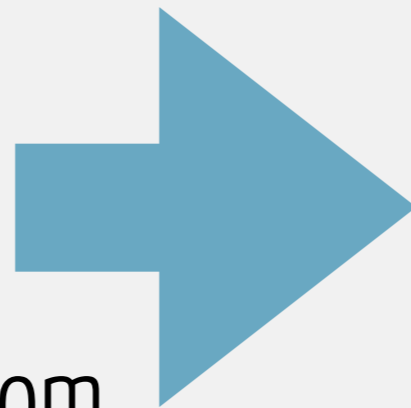
```
{ 'fabien.hermenier@unice.fr' :
  { firstname: 'Fabien',
    lastname: 'Hermenier',
    group: '4',
    www : 'https://sites.google.com/site/hermenierfabien/',
    twitter: 'fhermeni',
    root : true
  },
  ...
}
```


simpleXML

API pour transformer un document XML en
un ensemble d'objets PHP

règles de conversions

noeud XML
attribut XML
noeud fils
noeuds ayant le même nom



objet SimpleXMLElement
entrée dans un tableau associatif
attributs portant le nom du fils
tableau

```
<?xml version='1.0' encoding='utf-8'?>
<users group='4'>
  <user login='foo'>
    <email>foo@unice.fr</email>
    <password>foo-p</password>
  </user>
  <user login='bar'>
    <email>bar@unice.fr</email>
    <password>bar-p</password>
  </user>
</users>
```

```
<?php
$users =
simplexml_load_file("users.xml");
$g = $users['group'];
$foo = $users->user[0];
$l = (string)$foo['login'];
$e = (string)$foo->email;
echo "Group: ".$g."<br/>";
echo "Login: $l<br/>";
echo "Email: $e<br/>";
?>
```

en pratique

```
<?xml version='1.0' encoding='utf-8'?>
<users group='4'>
  <user login='foo'>
    <email>foo@unice.fr</email>
    <password>foo-p</password>
  </user>
  <user login='bar'>
    <email>bar@gmail.com</email>
    <password>bar-p</password>
  </user>
</users>
```

```
<?php
$users =
simplexml_load_file("users.xml");
echo "Membre du groupe ".$users['group'];
echo "<ul>";
foreach ($users->user as $u) {
    $m = (string) $u->email;
    $l = (string)$u['login'];
    echo "<li>$l: $m</li>";
}
echo "</ul>";
```

Membre du groupe 4

- foo: foo@unice.fr
- bar: bar@gmail.com



modifier du XML

```
<?php
function add_user($xml, $login, $email, $password) {
    $u = $xml->addChild('user');
    $u['login'] = $login;
    $u->addChild('email', $email);
    $u->addChild('password', md5($password));
}

$users = simplexml_load_file("users.xml");
add_user($users, 'baz', 'baz@unice.fr', 'aPassword');
if (!$users->asXML('users.xml')) {
    echo "Impossible de sauvegarder le fichier";
}
?>
```

```
<?xml version='1.0' encoding='utf-8'?>
<users group='4'>
  <user login='foo'>...</user>
  <user login='bar'>...</user>
  <user login='baz'>
    <email>baz@unice.fr</email>
    <password>2cd4675.../password>
  </user>
</users>
```

requêtes sur du XML

- des identifiants sont ils correct ?
- quels sont les messages postés depuis hier ?

```
<?php
function isRegistered($xml, $login, $password) {
    foreach ($xml->user as $u) {
        $l = (string)$u['login'];
        $p = (string)$u->password;
        if ($l == $login && $p == $password) {
            return true;
        }
    }
    return false;
}
?>
```

à l'ancienne ...

XPath

(http://www.w3schools.com/xpath/xpath_syntax.asp)

langage déclaratif pour sélectionner un **ensemble de noeuds**

- tous les noeuds 'email'
`user/email`
- tous les attributs 'login'
`user/@login`
- les noeuds 'user' dont le text du noeud fils 'email' commence par 'b'
`user[starts-with(email, 'b')]`
- les 'user' dont les identifiants sont ...
`user[@login='foo' and password='foo']`

\$xml->xpath()

retourne **toujours** un tableau de `simpleXMLElement` (ou `false` si problème)

```
<?php
function isRegistered($xml, $l, $p) {
    $res = $xml->xpath("user[@login='$l' and password='$p']");
    return count($res) == 1;
}

function getUser($xml, $l) {
    $r = $xml->xpath("user[@login='$l']");
    return (count($r) == 0 ? false : $r[0]);
}

function getLogins($xml) {
    $logins = array();
    foreach ($xml->xpath("user/@login") as $l) {
        $logins[] = (string)$l;
    }
    return $logins;
}
?>
```

TOP

4

des erreurs

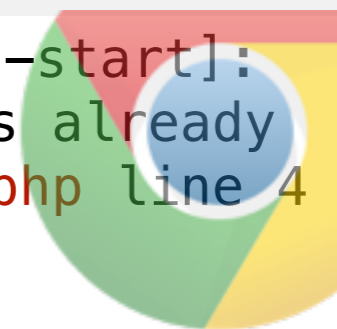

```
<?php
echo "Nb of visits: " . nbVisits();

session_start();
require_once("inc/counter.inc");

incVisit();

if (isset($_GET['stop'])) {
    cleanCounter();
}
?>
```

Warning: session_start() [function.session-start]:
Cannot send session cache limiter - headers already
sent (output started at line 2) in counter.php line 4

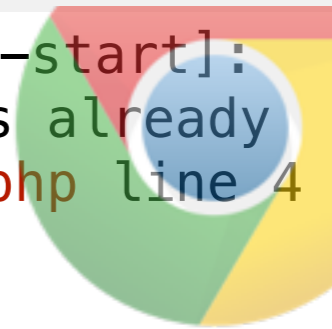


```
<?php
echo "Nb of visits: " . nbVisits();

session_start();
require_once("inc/counter.inc");
```

!\ pas de contenu
avant session_start()

```
Warning: session_start() [function.session-start]:  
Cannot send session cache limiter - headers already  
sent (output started at line 2) in counter.php line 4
```



```
<?php
function add_user($xml, $login, $email, $password) {
...
}

function getUser($xml, $l) {
    $r = $xml->xpath("user[@login='$l']");
    return (count($r) == 0 ? false : $r[0]);
}

$users = simplexml_load_file("users.xml");
add_user($users, 'baz', 'baz@unice.fr', 'aPassword');
$users->asXML('data.xml');

//Recharge le fichier
$users = simplexml_load_file("users.xml");
var_dump(getUser($xml, 'baz'));
?>
```

bool(false)



```
<?php
function add_user($xml, $login, $email, $password) {
    ...
}

function getUser($xml, $l) {
    $r = $xml->xpath("user[@login='$l']");
    return (count($r) == 0 ? false : $r[0]);
}

$xml = simplexml_load_file("users.xml");
```

#!/ permissions pour apache
de manipuler vos fichiers

bool(false)



```
<?php
function isRegistered($xml, $login, $password) {
    foreach($xml->user as $u) {
        $l = $u['login'];
        $p = $u->password;
        if ($l === $login && $p === $password) {
            return true;
        }
    }
    return false;
}

$xml = simplexml_load_file("users.xml");
$b = isRegistered($xml, "foo", "foo-p");
echo "foo ? " . ($b? "ok": "ko");
?>
```

```
<?xml version='1.0' encoding='utf-8'?>
<users group='4'>
  <user login='foo'>
    <email>foo@unice.fr</email>
    <password>foo-p</password>
  </user>
</users>
```

ko



```
<?php
function isRegistered($xml, $login, $password) {
    foreach($xml->user as $u) {
        $l = $u['login'];
        $p = $u->password;
        if ($l === $login && $p === $password) {
            return true;
        }
    }
}
```

/!\ il faut caster un noeud
pour récupérer son contenu:
(string)\$u->password

```
<users group=4 >
<user login='foo'>
  <email>foo@unice.fr</email>
  <password>foo-p</password>
</user>
</users>
```



```
<?php
function getUser($xml, $l) {
    $r = $xml->xpath("user[@login='$l']");
    return (count($r) == 0 ? false : $r);
}
?>
```

```
<?
$u = getUser($xml, "foo");
echo "Email: ". $u->email;
?>
```

Email:



```
<?php
function getUser($xml, $l) {
    $r = $xml->xpath("user[@login='$l']");
    return (count($r) == 0 ? false : $r);
}
?>
```

```
<?
$u = getUser($xml, "foo");
echo "Email: ". $u->email;
?>
```

#!/\ xpath() renvoie toujours
un tableau: \$r[0]