PHP AVANCE

1.1 POO

1.2 POO

2.1 POO

2.2 POO

3.1 Exceptions et Gestion d'erreurs

3.2 Fichiers : UPLOAD, data mining

4.1 WebServices

4.2 Mail et Graphics GD

Jour 1, matin : Introduction à la POO en PHP

1. Résumée du cours

- Gestion d'erreurs
- Téléchargement de fichiers
- XML
- SOAP
- Graphics

Environnement objet sur PHP :

- Eclipse
- Zend Studio

2. PHP et la POO

- Langage procédural
- Introduit objet dans la version 4
- Modifié dans la version 5

On peut programmer en PHP sans utiliser objet, c'est une question de style et de marché.

➔ Mais pour utiliser certaines fonctions, il faut employer la syntaxe objet.

➔ Style de programmation : une classe, un fichier

- Problème avec beaucoup d'includes.

⇒ Include_once, Require

3. Classe et objet

En JS il n'y avait pas cette distinction : objet prototype.

Java respecte la définition de la POO, en distinguant clairement la classe de l'objet. La classe n'est pas un objet spécial comme en JS, mais une sorte de container d'un autre type.

Si JS définissait le prototype par son constructeur, avec le mot fonction, PHP possède le mot Class. Par exemple, pour créer la classe Magasin :

```
class Magasin {  
}
```

Créer cette fonction dans la page magasin.php et créer la page testMagasin :

```
include "magasin.php";  
$m = new Magasin();  
echo ($m);
```

Si l'on essaye de faire un echo de \$m, on a une erreur. Nous avons besoin de connaître un petit plus sur les fonctions...

4. Méthodes

On écrit normalement une méthode de la façon suivante:

```
function write($produit ) {  
    // ...  
}
```

5. Méthode `__toString`

Pour ne pas afficher le code de l'objet, PHP permet la méthode `toString` :

```
class Magasin {  
  
    function __toString() {  
        return "** Magasin";  
    }  
  
}
```

Modifier alors la classe magasin pour inclure une méthode spécifique qui permet l'affichage d'une classe avec les valeurs que l'on souhaite : la méthode `__toString()`.

PHP possède un ensemble de méthodes spéciaux qui commencent avec le double underscore. La méthode `__toString()` est conçue pour permettre le bon fonctionnement d'un echo sur l'objet, et ainsi elle doit retourner une String. Coder alors cette méthode :

```
<?php  
class Magasin {  
  
    function __toString() {  
        return "** Magasin";  
    }  
}  
?>
```

6. Les propriétés

On les déclare dans le corps de la classe avec le mot `var` :

```
class Magasin {  
    public $v1 = 7;  
    function __toString() {  
        return "** Magasin : " . $this->v1;  
    }  
}
```

Elles sont des propriétés de l'objet, étant accessibles avec le mot `$this` : `$this->v1` ;

8. Visibilité

Var, Public

Private

Protected : visível na classe ou nas subclasses

Créer la variable \$v1 comme public et comme privé.

9. Getters et setters

Encapsulation de \$v1.

10. Les constructeurs

Constructeur dans PHP5 : `__construct()` (diffère de PHP 4, où l'on a le même nom de la classe)

```
class Magasin {
    private $nom;

    function __construct($nom) {
        $this->nom = $nom ;
    }

    function __toString() {
        return "** Magasin : " . $this-> nom;
    }
}
```

Créer un deuxième objet :

```
class Produit {
    public $ nom;
    public $prix= 0;

    function __construct( $nom, $prix ) {
        $this-> nom = $ nom;
        $this->prix = $prix;
    }

    function getPrix() {
        return "Prix : {$this->prix}";
    }
}
```

7. La méthode magique autoload

On écrit maintenant la classe Produit avec la même logique. Or, dans la classe appelant il faut inclure une include pour chaque.

La solution est donnée par la méthode __autoload. Il faut créer un fichier par classe avec exactement le même nom.

```
function __autoload($ClassName) {
    include 'classes/' . $ClassName . '.php';
}
```

11. Projet 1 : Magasin-produit

Créer 2 magasins avec 3 et 4 produits.

Magasin :

```
class Magasin {
    private $nom;
    public $tabProduit = Array() ;

    function __construct($nom) {
        $this->nom = $nom ;
    }

    function nouveauProduit(Produit $produit) {
        $this->tabProduit[]=$produit ;
    }

    function afficheProduits() {
        foreach ($this->tabProduit as $p){
            echo '<br> >> ' . $p;
        }
    }

    function __toString() {
        echo "<br><br> ** Magasin " . $this-> nom;
        $this->afficheProduits();
        return "<br>";
    }
}
```

Produit.php :

```
Class Produit {
    public $nom;
    private $prix;

    function __construct($nom, $prix) {
        $this->nom = $nom;
        $this->prix = $prix;
    }

    function __toString() {
        return " Produit : " . $this->nom . " , prix : " . $this->prix;
    }

    function getPrix() {
        return "Prix : {$this->prix}";
    }
}
```

testMagasin.php

```
function __autoload($ClassName) {
    include 'classes/' . $ClassName . '.php';
}
$m1 = new Magasin("Darty");
$p1 = new Produit("Ipod", 222);
$m1->nouveauProduit($p1);
$p2 = new Produit("Plasma", 5423);
$m1->nouveauProduit($p2);
echo ($m1);

$m2 = new Magasin("Fnac");
$p1 = new Produit("Ipod", 222);
$m2->nouveauProduit(new Produit("Ipod", 205));
$m2->nouveauProduit(new Produit("Archos", 44));
$m2->nouveauProduit(new Produit("CD", 125));
echo ($m2);
```

12. Projet 2 : Magasin-produit-personnes

Créer 1 Client et faire des courses dans les deux magasins et comparer les coûts.

testMagasin :

```
function __autoload($ClassName) {
    include 'classes/' . $ClassName . '.php';
}
$m1 = new Magasin("Darty");
$p1 = new Produit("Ipod", 222);
$m1->nouveauProduit($p1);
$p2 = new Produit("Plasma", 5423);
$m1->nouveauProduit($p2);
$m1->nouveauProduit(new Produit("ecouteurs ", 21));
$m1->nouveauProduit(new Produit("Ixus", 145));
$m1->nouveauProduit(new Produit("Pentax", 2002));
echo ($m1);
```

```
$m2 = new Magasin("Fnac");
$p1 = new Produit("Ipod", 222);
$m2->nouveauProduit(new Produit("Ipod", 205));
$m2->nouveauProduit(new Produit("Archos", 44));
$m2->nouveauProduit(new Produit("CD", 125));
echo ($m2);
```

```
$c1 = new Client("Pierre");
$c1->nouveauProduit($m1->getProduit(0));
$c1->nouveauProduit($m1->getProduit(1));
$c1->nouveauProduit($m1->getProduit(4));
$c1->afficheTotal();
```

Magasin :

```
class Magasin {
    private $nom;
    private $tabProduit = Array() ;

    function __construct($nom) {
        $this->nom = $nom ;
    }

    function nouveauProduit(Produit $produit) {
        $this->tabProduit[]=$produit ;
    }

    function afficheProduits() {
        foreach ($this->tabProduit as $p){
            echo '<br> >> ' . $p;
        }
    }

    public function getProduit( $ind ){
        return $this->tabProduit[$ind];
    }

    function __toString() {
        echo "<br><br> ** Magasin " . $this-> nom;
        $this->afficheProduits();
        return "<br>";
    }
}
```

Produit :

```
Class Produit {
    public $nom;
    private $prix;

    function __construct($nom, $prix) {
        $this->nom = $nom;
        $this->prix = $prix;
    }

    function getPrix() {
        return $this->prix;
    }

    function __toString() {
        return " Produit : " . $this->nom . " , prix : " . $this->prix;
    }
}
```

Client :

```
class Client {
    private $nom;
    public $tabProduit = Array() ;

    function __construct($nom) {
        $this->nom = $nom ;
    }

    function nouveauProduit(Produit $produit) {
        $this->tabProduit[]=$produit ;
    }

    function afficheProduits() {
        foreach ($this->tabProduit as $p){
            echo '<br> >> ' . $p;
        }
    }

    function afficheTotal() {
        echo "<br><br> --> Total de " . $this-> nom . " est de " . $this->getTotal();
    }

    function getTotal() {
        $tot=0;
        foreach ($this->tabProduit as $p){
            $tot += $p->getPrix();
        }
        return $tot;
    }

    function __toString() {
        echo "<br><br> ** Magasin " . $this-> nom;
        $this->afficheProduits();
        return "<br>";
    }
}
```

Jour 1, après-midi

1. Gestion des sessions

Sessões HTTP servem para preservar dados em acessos subsequentes, através de registros de variáveis de sessão. Usamos basicamente três funções:

- `session_start()`; // crée ou recupère une session
- `session_register( "VARIÁVEL" );` → Ou se referer directement `$_SESSION['toto'] = 1 ;`
- `session_destroy()`;

2. Méthodes de propagation

Il y a deux modes de propagation de l'identifiant de session :

- Cookies
- Paramètre URL

Le module de session supporte les deux techniques. La méthode par cookies est optimale, mais étant donné le peu de fiabilité (les clients peuvent les refuser, ou les effacer), on ne peut pas se contenter de cette technique. La deuxième méthode place l'identifiant de session directement dans l'URL.

PHP est capable de gérer ceci de manière transparente, lorsque vous le compilez avec l'option `--enabletrans-sid`. Dans ce cas, les URL relatives seront modifiées pour contenir l'identifiant de session automatiquement. Sinon, vous pouvez toujours utiliser la constante `SID`, qui sera définie si le client n'envoie pas le cookie approprié. `SID` prend la forme de `session_name=session_id`, ou bien, c'est une chaîne vide.

Verifier avec `phpinfo()`

session

Session Support	enabled
Registered save handlers	files user

Directive	Local Value	Master Value
session.auto_start	Off	Off
session.bug_compat_42	Off	Off
session.bug_compat_warn	On	On
session.cache_expire	180	180
session.entropy_length	0	0
session.gc_divisor	1000	1000
session.gc_maxlifetime	1440	1440
session.gc_probability	1	1
session.name	PHPSESSID	PHPSESSID
session.referer_check	<i>no value</i>	<i>no value</i>
session.save_handler	files	files
session.save_path	C:\PROGRA~1\EASYPH~1\tmpl	C:\PROGRA~1\EASYPH~1\tmpl
session.serialize_handler	php	php
session.use_cookies	On	On
session.use_only_cookies	Off	Off

session.use_trans_sid	Off	Off
------------------------------	------------	------------

3. Paramètre url

L'appel à **session_start()** doit être la première ligne, car il envoie des headers.

On travaille avec des variables qui sont passées dans la url : utiliser **\$_POST** (mieux que **\$_GET**). Les variables de contrôle de la session se trouvent dans **\$_SESSION**.

Exemple :

Index

Page x

Nom -----> **\$_SESSION** ['Nom']
Mdp

```
$_SESSION['Nom'] = $_POST['Nom']  
$_SESSION['active'] = « oui »
```

Remarque : ne pas oublier de tester **isset()**

Si les cookies ne marchent pas, utiliser la constante **SID** ou la fonction **session_id()**

A retenir à propos de la gestion des sessions sur PHP

- 1) **session_start()** doit être présente dans toutes les pages concernées par la session.
- 2) **session_start()** doit être la première instruction d'une page php (avant toute balise HTML).
- 3) Sauvegarder les variables de session dans le tableau global **\$_SESSION**.
- 4) Ne pas oublier de tester si les variables **\$_SESSION**, **\$_POST** ou **\$_GET** sont renseignées avec **isset()**.
- 5) Pour sauver des objets, il faut leur mettre dans la session. & Est l'adressage par référence.

Projet 3 : Faire l'achat dans la page suivante

TestMagasin1 :

```
<?php
function __autoload($ClassName) {
    include 'classes/' . $ClassName . '.php';
}
session_start();?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Document sans titre</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body>
<?php
$m1 = &$_SESSION ['m1'];
$m1 = new Magasin("Darty");
$p1 = new Produit("Ipod", 222);
$m1->nouveauProduit($p1);
$p2 = new Produit("Plasma", 5423);
$m1->nouveauProduit($p2);
$m1->nouveauProduit(new Produit("ecouteurs ", 21));
$m1->nouveauProduit(new Produit("Ixus", 145));
$m1->nouveauProduit(new Produit("Pentax", 2002));
echo ($m1);

?>
<p>&nbsp;</p>
<p><a href='testMagasin2.php'>Faire les courses</a></p>
</body>
</html>
```

TestMagasin2 :

```
<?php function _autoload($ClassName) {
    include 'classes/' . $ClassName . '.php';
}
session_start();?><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Document sans titre</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body>
<?php
$c1 = new Client("Pierre");
$c1->nouveauProduit($_SESSION['m1']->getProduit(0));
$c1->nouveauProduit($_SESSION['m1']->getProduit(1));
$c1->nouveauProduit($_SESSION['m1']->getProduit(4));
$c1->afficheTotal();
?>
</body>
</html>
```

Jour 2 : POO

0. Gestion de sessions

Exercice :

Créer un site avec 4 pages : accueil, login, valider, deja.

J'ai crée aussi la classe Utilisateur.php, que vous devez copier.

Gestion de session : login, mdp et acces (oui/non)

La page d'accueil dit bonjour au login:

```
<?php
session_start();
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Document sans titre</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
</head>

<body>
<?php
if ( !empty($_SESSION['login']) )
    echo "Bonjour " . $_SESSION['login'];
?>
<p>&nbsp;</p>
<p>Soyez le bienvenu</p>
<p>&nbsp;</p>
<p><a href="login.php">Acc&egrave;s restreint</a> </p>
</body>
</html>
```

Login :

```
<?php
session_start();

if ( !empty( $_SESSION['acces'] ) )
    if ( $_SESSION['acces'] == "oui" )
        header ("Location:deja.php");

?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Document sans titre</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
</head>

<body>

<form method = "post" action = "valider.php">
<fieldset>
<legend>Accès réservé aux personnes autorisées: Veuillez vous identifier.</legend>
Login <input type = "text" name = "login">
Pass <input type = "password" name = "mdp">
<input type = "submit" name = "envoi" value = "Entrer"/>
</fieldset>
</form>
<?php
if ( $_SESSION['acces'] == "non" )
    echo "<b>Veuillez réessayer...</b>";
?>
</body>
</html>
```

Valider :

```
<?php
session_start();
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Document sans titre</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
</head>

<body>
<?php
include 'classes/Utilisateur.php';

$u = new Utilisateur( $_POST['login'], $_POST['mdp'] );
if ( $u->existe() ){
    $_SESSION['acces'] = "oui";
    $_SESSION['login'] = $_POST['login'];

    echo "Soyez le bienvenue <b>" . $_SESSION['login'] . "</b>";
} else {
    $_SESSION['acces'] = "non";
    header ("Location:login.php");
}
?>
<p>Pour mettre à jour les pages du site, cliquez ici... </p>
</body>
</html>
```

Deja :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Document sans titre</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body>
Vous êtes déjà connecté: cliquez ici pour accéder au catalogue...
</body>
</html>
```

1. Méthodes et propriétés statiques

```
static public $i ;  
static public maMethode()
```

```
constantes : const
```

2. Projet 4 : Afficher le total de produits des deux magasins

```
class Magasin {  
    private $nom;  
    private $tabProduit = Array() ;  
    static private $totalProduits;  
  
    function __construct($nom) {  
        $this->nom = $nom ;  
    }  
  
    function nouveauProduit(Produit $produit) {  
        $this->tabProduit[]=$produit ;  
        self::$totalProduits++;  
    }  
  
    function afficheProduits() {  
        foreach ($this->tabProduit as $p){  
            echo '<br> >> ' . $p;  
        }  
    }  
  
    public function getProduit( $ind ){  
        return $this->tabProduit[$ind];  
    }  
  
    static function totalProduits() {  
        echo "<br><br><br><br>TOTAL DE PRODUITS DANS LES MAGASINS: " .  
self::$totalProduits;  
    }  
  
    function __toString() {  
        echo "<br><br> ** Magasin " . $this-> nom;  
        $this->afficheProduits();  
        return "<br>";  
    }  
}
```

```

<?php

function __autoload($ClassName) {
    include 'classes/' . $ClassName . '.php';
}
$m1 = new Magasin("Darty");
$p1 = new Produit("Ipod", 222);
$m1->nouveauProduit($p1);
$p2 = new Produit("Plasma", 5423);
$m1->nouveauProduit($p2);
$m1->nouveauProduit(new Produit("ecouteurs ", 21));
$m1->nouveauProduit(new Produit("Ixus", 145));
$m1->nouveauProduit(new Produit("Pentax", 2002));
echo ($m1);

$m2 = new Magasin("Fnac");
$p1 = new Produit("Ipod", 222);
$m2->nouveauProduit(new Produit("Ipod", 205));
$m2->nouveauProduit(new Produit("Archos", 44));
$m2->nouveauProduit(new Produit("CD", 125));
echo ($m2);

echo $l1;

$c1 = new Client("Pierre");
$c1->nouveauProduit($m1->getProduit(0));
$c1->nouveauProduit($m1->getProduit(1));
$c1->nouveauProduit($m1->getProduit(4));
$c1->afficheTotal();

Magasin::totalProduits();

?>

```

3. Héritage

```

Class fils extends pere {
}

```

3. Les constructeurs dans l'Héritage

Dans ce cas, c'est au fils de passer les paramètres nécessaires au constructeur du père.

Mot réservé parent. Et pour se référer à une méthode d'une classe, utiliser :: car le -> est pour l'objet.

Alors :

```
parent::__construct()
```

Exercice :

Créer la classe Livre (avec auteur).

```
Class Livre extends Produit{
    private $auteur;

    function __construct($nom, $prix, $auteur) {
        parent::__construct($nom, $prix);
        $this->auteur = $auteur;
    }

    function getPrix() {
        return $this->prix;
    }

    function __toString() {
        return parent::__toString() . ", auteur : " . $this->auteur;
    }
}
```

```
<?php
```

```
function __autoload($ClassName) {
    include 'classes/' . $ClassName . '.php';
}
$m1 = new Magasin("Darty");
$p1 = new Produit("Ipod", 222);
$m1->nouveauProduit($p1);
$p2 = new Produit("Plasma", 5423);
$m1->nouveauProduit($p2);
$m1->nouveauProduit(new Produit("ecouteurs ", 21));
$m1->nouveauProduit(new Produit("Ixus", 145));
$m1->nouveauProduit(new Produit("Pentax", 2002));
echo ($m1);

$m2 = new Magasin("Fnac");
$p1 = new Produit("Ipod", 222);
$m2->nouveauProduit(new Produit("Ipod", 205));
$m2->nouveauProduit(new Produit("Archos", 44));
$m2->nouveauProduit(new Produit("CD", 125));
$m2->nouveauProduit(new Livre("Route et noir", 22, "Stendal"));
echo ($m2);
```

```
echo $l1;
```

```
$c1 = new Client("Pierre");
$c1->nouveauProduit($m1->getProduit(0));
```

```
$c1->nouveauProduit($m1->getProduit(1));  
$c1->nouveauProduit($m1->getProduit(4));  
$c1->afficheTotal();  
?>
```

4. Redéfinition des méthodes

Dans la nouvelle méthode, on peut appeler parent :: methode()

⇒ Le mot self :: se réfère à la classe même

5. Classes abstraites

abstract class

6. Interface

Interface chargeable {}

→ é implements ?

7. Destructeurs

Libérer des ressources

__destruct()

8. Copier un objet

En PHP4, il s'agissait de fournir la même adresse :

```
class CopyMe {}  
$first = new CopyMe();  
$second = $first;  
// PHP 4: $second and $first are 2 distinct objects  
// PHP 5 plus: $second and $first refer to one object
```

PHP5 :

```
class CopyMe {}  
$first = new CopyMe();  
$second = clone $first;  
// PHP 5 plus: $second and $first are 2 distinct objects
```

Pour contrôler ce qui est copié dans un objet : `__clone()`

9. Comparaison d'objets

Avec l'opérateur d'identité `===` on teste si les objets font référence à la même instance de la même classe.

Exercice :

Créer deux magasins. Copier le premier. Tester l'égalité.

```
$a = new Test;  
$b = new Test;  
$c = $a;  
// $c = clone($a);
```

```
if ($a == $b) echo "*1";  
else echo "*2";
```

```
if ($a === $b) echo "*3";  
else echo "*4";
```

```
if ($a == $c) echo "*5";  
else echo "*6";
```

```
if ($a === $c) echo "*7";           // com =  
else echo "*8";                     // com clone
```

10. Serialization

Pour passer des objets entre des pages.
Deux methodes : `__sleep()` et `__wakeup()`

```
<?php
class base_class {
    var $var1;
    var $var2;

    function __construct ($value) {
        $this->var1 = $value;
        $this->var2 = $value * 100;
    }

    function calc_pow ($exp){
        return pow ($var1, $exp);
    }

    function __toString() {
        return 'v1 : ' . $this->var1 . ' v2 : ' . $this->var2 ;
    }

    function __sleep(){
        // Return an array that contains the name of all the variables to be saved
        return array ('var1');
    }

    function __wakeup(){
        // Reconstruct $var2
        $this->var2 = $this->var1 * 100;
    }
}

$bc = new base_class(5);

$cdSerial = serialize($cd);

$params = urlencode($cdSerial);

echo "<A HREF='genre.php?var=$params'>Afficher le genre</A>";
```

Jour 2, après-midi

1. Projet

Construction d'une classe de connexion abstraite pour l'interrogation des principales bases de données.

Jour 3, matin : Les fichiers et les exceptions

1. Extensions PHP

Le monde PHP propose 150 librairies PHP de qualité très variable. Nous allons nous pencher sur : les moteurs de template, le module GD, PDFlib et l'incontournable PEAR.

Nous allons voir aussi: XML, RSS et SOAP. Mais avant, nous allons voir la gestion des fichiers.

2. Utilisation d'un gestionnaire de fichier

Fonctions disponibles:

fopen() :	Opens a handle to a local file, or a file from an URL
fread() :	Reads a block of data from a file
fgets() :	Reads one single line from a file
fwrite() :	/
fputs() :	Writes a block of data to a file
fclose() :	Closes the opened file handle
feof() :	Returns true when the end of the file has been reached

PHP possui várias funções para o tratamento de arquivos a fim de facilitar sua manipulação. A primeira coisa que se tem que saber é que para se manipular arquivos, tem-se que abrir e fechar o arquivo. O PHP possui as seguintes funções para se abrir e fechar arquivos:

- `fopen( "nome_do_arquivo", "mode", [diretório] )` – Abre o arquivo
- `fclose( fp )` - Fecha o arquivo

Estes modos de abrir o arquivo podem ser os seguintes:

r somente leitura; posiciona-se no início do arquivo.

r+ leitura e escrita; posiciona-se no início do arquivo.

w somente escrita; posiciona-se no início do arquivo e o trunca para tamanho zero. Caso o arquivo não exista o PHP tenta criá-lo.

w+ leitura e escrita; posiciona-se no início do arquivo e o trunca para tamanho zero. Caso o arquivo não exista o PHP tenta criá-lo.

a somente escrita; posiciona-se no final do arquivo. Caso o arquivo não exista o PHP tenta criá-lo.

a+ leitura e escrita; posiciona-se no final do arquivo. Caso o arquivo não exista o PHP tenta criá-lo.

Após abrir um arquivo, pode-se ler o conteúdo ou escrever em seu conteúdo, de acordo com a forma que se abriu o arquivo.

- `fread(fp, tamanho )`
- `fgets(fp, tamanho )`
- `fwrite(fp, string, [tamanho] )`

5. Travailler avec des fichiers est facile

Soit on crée un fichier de façon traditionnelle :

```
$fp = fopen("/tmp/teste.txt", "w");  
fwrite($fp, "bla bla bla \n");  
fclose($fp);
```

Soit on crée avec une méthode rapide :

```
file_put_contents ( "test.txt" , "Bonjour" );  
  
$fp = fopen ('test.txt', 'r');  
if (!$fp) {  
    die ("Le fichier n'a pas put être ouvert.");  
}  
  
$line = fgets($fp);  
echo $line;  
fclose($fp);
```

Mais on peut aussi lire un script :

```
$file = fopen ('accueil.php', 'r');  
while (!feof ($file)) {  
    $line = fgets ($file, 1024);  
    echo $line;  
}  
  
fclose($file);
```

ou

```
$fp = fopen ($nomedoarquivo, "r");  
$contents = fread ($fd, filesize ($nomedoarquivo));  
fclose ($fp);
```

On peut aussi retirer ses balises HTML :

```
$file = fopen ('accueil.php', 'r');
while (!feof ($file)) {
    $line = fgets ($file, 1024);
    echo $line;
}

fclose($file);
```

Ou on peut alors utiliser une fonction qui lit tout et affiche tout:

```
$fp = fopen('accueil.php','r');
fpassthru($fp);
On peut telecharger en changeant le type :
```

```
<?php
// $file = $_POST['date'].'.txt';
$file = 'rss.xml';
$fp = fopen($file, "r");
header("Content-Type:application/xls");
//header("Content-Disposition:attachment; filename=$_POST['date'].xls");
header("Content-Disposition:attachment; filename=$file");
// Notice we changed the file name and type
header("Content-Transfer-Encoding:binary");
fpassthru($fp);
?>
```

6. Travailler avec des expressions régulières

```
Ereg
Eregi
Ereg_replace
```

Simple literal character matches

If the character involved is not special, Perl will match characters in sequence. The example pattern `/abc/` matches any string that has the substring 'abc' in it.

Character class matches: [*<list of characters>*]

Will match a single instance of any of the characters between the brackets. For example, `/[xyz]/` matches a single character, as long as that character is either x, y, or z. A sequence of characters (in ASCII order) is indicated by a hyphen, so that a class matching all digits is `[0-9]`.

Predefined character class abbreviations

The patterns `\d` will match a single digit (from the character class `[0-9]`), and the pattern `\s` matches any whitespace character.

Multiplier patterns

Any pattern followed by * means: "Match this pattern 0 or more times."

Any pattern followed by ? means: "Match this pattern exactly once."

Any pattern followed by + means: "Match this pattern 1 or more times."

Anchoring characters

The caret character ^ at the beginning of a pattern means that the pattern must start at the beginning of the string; the \$ character at the end of a pattern means that the pattern must end at the end of the string. The caret character at the beginning of a character class [^abc] means that the set is the complement of the characters listed (that is, any character that is not in the list).

Escape character '\'

Any character that has a special meaning to regex can be treated as a simple matching character by preceding it with a backslash. The special characters that might need this treatment are:

. \ + * ? [] ^ \$ () { } = ! < > | :

Parentheses

A parenthesis grouping around a portion of any pattern means: "Add the substring that matches this pattern to the list of substring matches."

Si "abc" est trouvé quelque part dans la chaîne \$string :

```
ereg ("abc", $string);
```

```
$string = "xabcd";
```

```
if (ereg ("abc", $string, $reg ) ) echo "oui $reg[0] <br>"; else echo "non<br>";
```

Si "abc" est trouvé au début de la chaîne \$string. :

```
ereg ("^abc", $string);
```

Si "abc" est trouvé à la fin de la chaîne \$string :

```
ereg ("abc$", $string);
```

**Insère une balise
 au début de la chaîne \$string :**

```
$string = ereg_replace ("^", "<br />", $string);
```

**Insère une balise
 à la fin de la chaîne \$string :**

```
$string = ereg_replace ("$", "<br />", $string);
```

Supprime toutes les nouvelles lignes de \$string :

```
$string = ereg_replace ("\n", "", $string);
```

Cherche le titre dans une page :

```
if (eregi ("<title>(.*?)</title>", $line, $out)) {  
    $title = $out[1];  
    break;  
}
```

7. Travailler avec des fichiers distantes

Quelques wrappers de fichier comme http permettent d'accéder des fichiers distants.

On peut faire des includes des fichiers distants

Il faut rendre cela possible dans apache

```
<body>
<p>Bonjour</p>
<table width="800" border="1">
  <tr>
    <td><?php include ("http://www.fnac.com"); fopen(); ?>
  </td>
</tr>
</table>
```

Lire la balise title :

```
$title = "xxx";
$file = fopen ("http://www.php.net/", "r");
if (!$file) {
    echo "<p>Impossible d'ouvrir le fichier distant.\n";
    exit;
}
while (!feof ($file)) {
    $line = fgets ($file, 1024);
    /* This only works if the title and its tags are on one line */
    if (eregi ("<title>(*)</title>", $line, $out)) {
        $title = $out[1];
        break;
    }
}
fclose($file);
echo "*****<br>";
echo "titre: " . $title;
```

8. Téléchargement d'un fichier

Créer un formulaire

```
<form action="testUpload2.php" method="POST" enctype="multipart/form-data">
  <input type="hidden" name="MAX_FILE_SIZE" value="25000"> Image à télécharger:
  <input type="file" name="fichier">
  <font size="1">Sélectionner parcourir pour sélectionner un fichier </font>
  <input type="submit" value="Télécharger">
</form>
```

Avant de copier le fichier, nous devons tester si l'utilisateur n'essaye pas de travailler sur des fichiers qu'il n'aurait pas le droit comme etc/password. On utilise la fonction `is_uploaded_file()`. Si cette fonction retourne TRUE, copier le fichier vers son dossier de destination avec la fonction `copy()`.

- **The enctype attribute is used in the form tag.** You must set this attribute to multipart/form-data when uploading a file to ensure that the file arrives correctly.
- **A hidden field is included that sends a value (in bytes) for MAX_FILE_SIZE.** If the user tries to upload a file that is larger than this value, it won't upload. You can set this value as high as 2MB. If you need to upload a file larger than that, you need to change the default setting for `upload_max_filesize` in `php.ini` to a larger number before sending a value larger than 2MB for `MAX_FILE_SIZE` in the hidden field.
- **The input field that uploads the file is of type file.** Notice that the field has a name — `user_file` — as do other types of fields in a form. The filename that the user enters into the form is sent to the processing program and is available in the built-in array called `FILES`.

When the user submits the form, the file is uploaded to a temporary location. The script that processes the form needs to copy the file to another location because the temporary file is deleted as soon as the script is finished.

```
$_FILES['fieldname']['name']
$_FILES['fieldname']['type']
$_FILES['fieldname']['tmp_name']
$_FILES['fieldname']['size']
```

For example, suppose that you use the following field to upload a file, as shown in the previous section.

If the user uploads a file named `test.txt` by using the form, the resulting array that can be used by the processing program looks something like this:

```
$_FILES[user_file][name] = test.txt
$_FILES[user_file][type] = text/plain
$_FILES[user_file][tmp_name] = D:\WINNT\php92C.tmp
$_FILES[user_file][size] = 435
```

On doit mouvoir le fichier téléchargé de son dossier temporaire vers un dossier fixe :

```
move_uploaded_file(path/tempfilename,path/permfilename);
```

The *path/tempfilename* is available in the built-in array element `$_FILES['fieldname']['tmp_file']`. The *path/permfilename* is the path to the file where you want to store the file. The following statement moves the file uploaded in the input field, given the name `user_file`, shown earlier in this section:

```
move_uploaded_file($_FILES['user_file']['tmp_name'], 'c:\data\new_file.txt');
```

```
$uploaddir = './image/';
$uploadfile = $uploaddir . basename($_FILES['fic_img']['name']);
echo '<pre>';

if(!ereg('image',$_FILES['fic_img']['type'])) {
    echo "<b>Le fichier n'est pas une image. Essayez un autre fichier.</b><br>";
} elseif (move_uploaded_file($_FILES['fic_img']['tmp_name'], $uploadfile)) {
    echo "Le fichier est valide, et a été téléchargé avec succès. Voici plus d'informations :\n";
} else {
    echo "Attaque potentielle par téléchargement de fichiers.
        Voici plus d'informations :\n";
}
echo 'Voici quelques informations de déboguage :';
print_r($_FILES);
echo '</pre>';
```

Après-midi

1. Simple XML

Représente le fichier XML comme un objet PHP. Étant donné le fichier :

```
<clients>
  <client>
    <name>John Doe</name>
    <account_number>87234838</account_number>
  </client>
  <client>
    <name>Janet Smith</name>
    <account_number>72384329</account_number>
  </client>
</clients>
```

Le code suivant imprime son contenu :

```
$clients = simplexml_load_file('clients.xml');
foreach ($clients->client as $client) {
    print "$client->name has account number $client->account_number<br>";
}
```



```

$pFile = new SimpleXMLElement('http://localhost/testObjet/rss.xml', null, true);

foreach ($pFile->channel as $pChild)
{
    echo "<h1>" . $pChild->title . "</h1>\n";
    echo "<p>\n";
    echo $pChild->description . "<br />\n";
    printf('Visiter noter site <a href="%s">%s</a><br />' . "\n", $pChild->link, $pChild->link);
    echo "</p>\n";

    foreach ($pFile->channel->item as $pItem)
    {
        echo "<p>\n";

        foreach ($pItem->children() as $pChild)
        {
            switch ($pChild->getName())
            {
                case 'title':
                    echo "<b>$pChild</b><br />\n";
                    break;

                case 'link':
                    printf('<a href="%s">%s</a><br />' . "\n", $pChild, $pChild);
                    break;

                default:
                    echo nl2br($pChild) . "<br />\n";
                    break;
            }
        }
        echo "</p>\n";
    }
}

```

Jour 4, matin : le reste

1. Spécification XML 2.0

```
<?xml version="1.0"?>
<rss version="2.0">
  <channel>
    <title>Lift Off News</title>
    <link>http://liftoff.msfc.nasa.gov/</link>
    <description>Liftoff to Space Exploration.</description>
    <language>en-us</language>
    <pubDate>Tue, 10 Jun 2003 04:00:00 GMT</pubDate>
    <lastBuildDate>Tue, 10 Jun 2003 09:41:01 GMT</lastBuildDate>
    <docs>http://blogs.law.harvard.edu/tech/rss</docs>
    <generator>Weblog Editor 2.0</generator>
    <managingEditor>editor@example.com</managingEditor>
    <webMaster>webmaster@example.com</webMaster>
    <ttl>5</ttl>

    <item>
      <title>Star City</title>
      <link>http://liftoff.msfc.nasa.gov/news/2003/news-starcity.asp</link>
      <description>How do Americans get ready to work with Russians aboard the
        International Space Station? They take a crash course in
        culture, language
        and protocol at Russia's Star City.</description>
      <pubDate>Tue, 03 Jun 2003 09:39:21 GMT</pubDate>
      <guid>http://liftoff.msfc.nasa.gov/2003/06/03.html#item573</guid>
    </item>

    <item>
      <title>Space Exploration</title>
      <link>http://liftoff.msfc.nasa.gov/</link>
      <description>Sky watchers in Europe, Asia, and parts of Alaska and
        Canada
        will experience a partial eclipse of the Sun on Saturday, May
        31st.</description>
      <pubDate>Fri, 30 May 2003 11:06:42 GMT</pubDate>
      <guid>http://liftoff.msfc.nasa.gov/2003/05/30.html#item572</guid>
    </item>

  </channel>
</rss>
```

2. Créer un fichier XML pour RSS:

```
<?xml version="1.0" encoding="iso-8859-1"?><rss version="2.0">
<channel>
<title>Teste de transfert</title>
<link>http://www.moi-transfert.com</link>
<description>Actualité des transferts avec XML.</description>
<copyright>© Tito</copyright>
<language>fr</language>
<image> <title>Teste de transfert RSS</title>
<url>Erreur ! Référence de lien hypertexte non valide.>
<link>http://www.moi-transfert.com</link></image>
<pubdate>Wed, 25 Jun 2008 00:02:17 +0100</pubdate>
<item><title>titre1</title><link></link></item>
<item><title>titre2</title><link></link></item>
<item><title>titre3</title><link></link></item>
<item><title>titre4</title><link></link></item>
<item><title>titre5</title><link></link></item>
</channel></rss>
```

3. Lecture du flux : simple

```
$url = 'http://localhost/testObjet/rss.xml';
$rss = simplexml_load_file($url);
foreach ($rss->channel->item as $article){
    echo '<a href="' . $article->link . '"'> . $article->title . '</a><br>';
}
```

LE MONDE :

```
$url = 'http://www.lemonde.fr/rss/sequence/0,2-3208,1-0,0.xml';
$rss = simplexml_load_file($url);
foreach ($rss->channel->item as $article){
    echo '<a href="' . $article->link . '"'> . $article->title . '</a><br>';
}
```

4. PDF

J'ai pris FPDF de www.fpdf.org.

Module objet avec les méthodes :

[AcceptPageBreak](#) - accepte ou non un saut de page automatique

[AddFont](#) - ajoute une nouvelle police

[AddLink](#) - crée un lien interne

[AddPage](#) - ajoute une nouvelle page

[AliasNbPages](#) - définit un alias pour le nombre de pages

[Cell](#) - imprime une cellule

[Close](#) - termine le document

[Error](#) - erreur fatale

[Footer](#) - pied de page

[FPDF](#) - constructeur

[GetStringWidth](#) - calcule la longueur d'une chaîne

[GetX](#) - renvoie la position x courante

[GetY](#) - renvoie la position y courante

[Header](#) - en-tête

[Image](#) - imprime une image

[Line](#) - trace une ligne

[Link](#) - place un lien

[Ln](#) - saut de ligne

[MultiCell](#) - imprime du texte avec saut de ligne

[Output](#) - sauve ou envoie le document

[PageNo](#) - numéro de page

[Rect](#) - trace un rectangle

[SetAuthor](#) - définit l'auteur du document

[SetAutoPageBreak](#) - fixe le mode saut de page automatique

[SetCompression](#) - active ou désactive la compression

[SetCreator](#) - définit le créateur du document

[SetDisplayMode](#) - fixe le mode d'affichage

[SetDrawColor](#) - définit la couleur de tracé

[SetFillColor](#) - définit la couleur de remplissage

[SetFont](#) - fixe la police

[SetFontSize](#) - fixe la taille de la police

[SetKeywords](#) - définit les mots-clés associés au document

[SetLeftMargin](#) - fixe la marge gauche

[SetLineWidth](#) - fixe l'épaisseur des traits

[SetLink](#) - définit la destination d'un lien interne

[SetMargins](#) - fixe les marges

[SetRightMargin](#) - fixe la marge droite

[SetSubject](#) - définit le sujet du document

[SetTextColor](#) - définit la couleur du texte

[SetTitle](#) - définit le titre du document

[SetTopMargin](#) - fixe la marge haute

[SetX](#) - fixe la position x courante

[SetXY](#) - fixe les positions x et y courantes

[SetY](#) - fixe la position y courante

Text - imprime une chaîne

Write - imprime du texte en mode flot

Installation :

- Copier dans la racine fpdf.php
- Copier dans la racine le logo tutorial pb...
- Copier dans la racine les 2 fichiers tutorial/20k_c...
- Copier le dossier fonts

Exemple 1 : Premier PDF

```
require('fpdf.php');
$pdf=new FPDF();
$pdf->AddPage();
$pdf->SetFont('Arial','B',16);
$pdf->Cell(40,10,'Hello World !');
$pdf->Output();
```

Exemple 2 : Avec Héritage

```
require('fpdf.php');

class PDF extends FPDF{

    function Header(){
        //Logo
        $this->Image('logo_pb.png',10,8,33);
        $this->SetFont('Arial','B',15);
        //Décalage à droite
        $this->Cell(80);
        //Titre
        $this->Cell(30,10,'Titre',1,0,'C');
        $this->Ln(20);
    }

    function Footer(){
        //Positionnement à 1,5 cm du bas
        $this->SetY(-15);
        $this->SetFont('Arial','I',8);
        //Numéro de page
        $this->Cell(0,10,'Page '.$this->PageNo().'/{nb}',0,0,'C');
    }
}

//Instanciation de la classe dérivée
$pdf=new PDF();
$pdf->AliasNbPages();
$pdf->AddPage();
$pdf->SetFont("Times","",12);
for($i=1;$i<=40;$i++)
    $pdf->Cell(0,10,'Impression de la ligne numéro '.$i,0,1);
```

```
$pdf->Output();
```

Exemple 4 : inclusion de fichiers

```
require('fpdf.php');

class PDF extends FPDF{
function Header(){
    global $titre;

    $this->SetFont('Arial','B',15);
    $w=$this->GetStringWidth($titre)+6;    //Calcul de la largeur du titre et positionnement
    $this->SetX((210-$w)/2);
    $this->SetDrawColor(0,80,180);        //Couleurs du cadre, du fond et du texte
    $this->SetFillColor(230,230,0);
    $this->SetTextColor(220,50,50);
    $this->SetLineWidth(1);                //Epaisseur du cadre (1 mm)
    $this->Cell($w,9,$titre,1,1,'C',1);    //Titre centré
    $this->Ln(10);
}

function Footer(){
    $this->SetY(-15);                        //Positionnement à 1,5 cm du bas
    $this->SetFont('Arial','I',8);
    $this->SetTextColor(128);
    $this->Cell(0,10,'Page '.$this->PageNo(),0,0,'C'); //Numéro de page
}

function TitreChapitre($num,$lib){
    $this->SetFont('Arial','',12);
    $this->SetFillColor(200,220,255);        //Couleur de fond
    $this->Cell(0,6,"Chapitre $num : $lib",0,1,'L',1); //Titre
    $this->Ln(4);
}

function CorpsChapitre($fichier){
    $f=fopen($fichier,'r');
    $txt=fread($f,filesize($fichier));
    fclose($f);
    $this->SetFont("Times","",12);
    $this->MultiCell(0,5,$txt);                //Sortie du texte justifié
    $this->Ln();
    $this->SetFont("",'I');
    $this->Cell(0,5,'(fin de l\'extrait)');
}

function AjouterChapitre($num,$titre,$fichier){
    $this->AddPage();
    $this->TitreChapitre($num,$titre);
    $this->CorpsChapitre($fichier);
}
}

$pdf=new PDF();
```

```

$titre='Vingt mille lieues sous les mers';
$pdf->SetTitle($titre);
$pdf->SetAuthor('Jules Verne');
$pdf->AjouterChapitre(1,'UN ÉCUEIL FUYANT','20k_c1.txt');
$pdf->AjouterChapitre(2,'LE POUR ET LE CONTRE','20k_c2.txt');
$pdf->Output();

```

5. Courrier

Libmail

```

<?php
    include 'inc/libmail.php';

    $m= new Mail(); // create the mail
    $m->From("test@gmail.com");
    $m->To("titomp2222@hotmail.com");
    $m->Subject("test email");
    ob_start();
    echo "<pre>";
    echo "test <br>";
    $error = ob_get_contents();
    ob_end_clean();
    $message = "Nous vous adressons un message :<br><br>${error}<br><br>Merci de le
r&eacute;gler au plus vite,<br>";
    $m->Body($message);
    $m->Send();
    echo "***** MAIL ENVOYE *****";
    return ;

```

6. Projet

Sans trop se préoccuper de la maquette, faire un projet pour un site de citations.

- 1) Créer la table citations avec
Citation, auteur, type (mort ou guerre)

- 2) Gérer un fichier RSS pour le flux.

- 3) Accueil avec trois liens : mort, guerre ou tout.

- 4) Dans la page liste, créer un bouton pour gérer pdf.

MORT

« Quand il s'agit d'un mortel, il faut attendre sa dernière journée avant de l'appeler heureux » (Sophocle)

« Pour bien aimer une vivante, il faut l'aimer comme si elle devait mourir demain » (Proverbe arabe)

« Une journée bien employée donne un bon sommeil; une vie bien employée procure une mort tranquille » (Léonard de Vinci, Carnets)

« La mort fait que l'on devient plus attentif à la vie » (Paulo Coelho, l'Alchimiste)

« Si un homme n'a pas découvert ce pour quoi il serait prêt à mourir, il n'est pas fait pour vivre » (Luther King)

« Qui apprendrait aux hommes à mourir, leur apprendrait à vivre » (Montaigne, Essais)

« Il faut choisir : savoir mourir ou ne pas vivre. Ceux qui craignent le plus la mort sont ceux qui n'ont jamais vécu. » (Raoul Follereau, Si le Christ demain...)

“La vie, c'est le temps que nous mettons à mourir” (Pr Louis Vincent Thomas, anthropologue)

« Accomplis chaque acte de ta vie comme s'il devait être le dernier » (Marc Aurèle, Pensées)

« Ce n'est pas la mort qui est redoutable mais la mort avec déshonneurs » (Epictète, Entretiens)

« Meilleur l'homme, moins il craint la mort » (Samuel Johnson, Boswell's life)

« Tu ne meurt pas de ce que tu es malade; tu meurt de ce que tu es vivant » (Montaigne)

« Le drame avec la vie, c'est qu'on en sors pas vivant. Alors la vraie question, c'est de savoir comment on va mourir » (Michel Audiard)

"A long terme, nous serons tous morts" (Keynes)

GUERRE

« Une mort est une tragédie , un million de morts est une statistique... »

Joseph Stalin.

« Je pense que des gens seraient vivants aujourd'hui s'il y avait la peine de mort... »

Nancy Reagan.

« La manière la plus rapide de mettre fin à une guerre est de la perdre... »

George Orwell.

« J'ignore quelles armes seront utilisées dans la Troisième Guerre mondiale, mais la Quatrième Guerre mondiale se déroulera à coup de bâtons et de pierres... »

Albert Einstein.

« La guerre, c'est une chose trop grave pour la confier à des militaires... »

Georges Clemenceau.

« La guerre est une série de désastres qui a pour résultat un vainqueur... »

Georges Clemenceau.

« La guerre est un jeu auquel on joue le sourire aux lèvres. Si vous n'arrivez pas à faire un sourire normal, faites un grand sourire. Si vous n'arrivez pas à faire un grand sourire, tenez-vous à l'écart jusqu'à ce que vous y arriviez... »

Winston Churchill.

« Etre prêt à la guerre est l'un des meilleurs moyens de préserver la paix... »

George Washington.

« La mort est une chose très ennuyeuse et consternante, et je ne peux que vous conseiller de ne pas y avoir affaire... »

W. Somerset Maugham.

« Un grand nombre de ceux qui vivent méritent de mourir. Et un grand nombre de ceux qui meurent méritent de vivre... »

J.R.R.Tolkien.

« Il reste enfin le plus ancien et le plus profond des désirs, la Grande évasion : l'évasion face à la mort... »

J.R.R.Tolkien.

« Quant à toi, Vie, je pense que tu es les restes de nombreuses morts, Je suis moi-même sûrement déjà mort dix mille fois... »

Walt Whitman.

« Les hommes sont en guerre les uns contre les autres car ils sont en guerre contre eux-mêmes... »

Francis Meehan.

« Ce n'est pas que j'aie peur de mourir, c'est juste que je préférerais ne pas être là quand ça arrivera... »

Woody Allen.

« Il est difficile d'être patient avec ceux qui disent "La mort n'existe pas" ou "La mort n'a aucune importance". La mort existe, et ce qui existe a de l'importance... »

C.S.Lewis.

« La mort d'un homme concerne plus ceux qui lui survivent que lui-même... »

Thomas Mann.

« Je ne crois pas en un destin qui frappe les hommes quels que soient leurs actes, mais je crois en un destin qui les frappe s'ils n'agissent pas... »

G. K. Chesterton.

« Un homme qui refuse de mourir pour une cause, ne mérite pas de vivre... »

Martin Luther King.

« Une fois la partie terminée, le roi et le pion retournent dans la même boîte... »

Proverbe italien.

« L'humanité doit mettre fin aux guerres, sans quoi les guerres mettront fin à l'humanité... »

John Fitzgerald Kennedy.

« Les guerres n'ont jamais fait de mal à personne, sauf à ceux qui meurent... »

Salvador Dali.

« Nous devons nous améliorer en remportant des victoires sur nous-mêmes. Il doit y avoir des affrontements, et nous devons gagner... »

Edward Gibbon.

« Dans la vie, comme à la guerre, nous ne réussissons qu'en choisissant un objectif unique et primordial et en assujettissant toutes les considérations à cet objectif... »

Dwight D. Eisenhower.

« L'expérience est un professeur cruel car elle nous fait passer notre examen avant de nous apprendre notre leçon... »

Vernon Sanders Law.

« Il vaut mieux tomber de tout son long vers l'avant que de se pencher en arrière trop loin... »

James Thurber.

« Le courage, c'est d'être mort de peur, mais de se mettre quand même en selle... »

John Wayne.

« La mort est plus universelle que la vie : tout le monde meurt, mais tout le monde ne vit pas... »

A. Sachs.

« On n'arrête pas le progrès. La preuve, c'est qu'à chaque guerre on se fait tuer par de nouvelles armes... »

Will Rogers.

« Un de ces jours, on organisera une guerre et personne ne viendra... »

Carl Sandburg.

« Heureusement que la guerre est si abominable, sans quoi nous risquerions de trop l'apprécier... »

Robert E. Lee.

« La guerre est l'un des fléaux que Dieu a décidé d'infliger aux hommes... »

Cardinal Richelieu.

« A la guerre, rien ne vaut la victoire... »

General Douglas MacArthur.

« La guerre ne permet pas de savoir qui a raison, mais uniquement qui survit... »

Anders Russell.

« Le but de la guerre n'est pas de mourir pour son pays, mais de faire en sorte que ces bâtards d'ennemies meurent pour le leurs... »

General G. C. Patton

« Les hommes braves sont tous des vertébrés, ils sont mous à l'extérieur et durs à l'intérieur... »

Lewis Carroll.

« Le soldat n'est pas homme de violence. Il porte les armes et risque sa vie, Pour des fautes qui ne sont pas les siennes. Son mérite est d'aller au bout de sa parole, tout en sachant qu'il est voué à l'oubli... »

A. de Saint-Exupéry.

« Tuez un homme, vous êtes un assassin; Tuez des milliers d'hommes, vous êtes un héros... »

Beilby Porteus.

« Aux yeux des fondateurs des grands empires, les hommes ne sont pas des hommes, ce sont des instruments... »

Napoléon 1^{er}.

« Il y a des héros ; il n'y a pas de peuples de héros ; il n'y a pas d'armées de héros. Les soldats n'ont jamais marché que sous peine de mort... »

Anatole France.

« Les braves mourront ici, les lâches iront mourir en Sibérie... »

Davout à Eylau.

« Je ne connais qu'une seule chose qui soit plus gênant lors d'une guerre que d'avoir des alliés : c'est de ne pas en avoir... »

Wiston Churchill.

« J'étais trop faible pour me défendre, alors, j'ai attaqué... »

Robert E. Lee.

9. La classe Exception PAS

PHP 5 introduit la classe Exception avec des méthodes suivants :

Description :

getMessage() Get the message string that was passed to the constructor.

getCode() Get the code integer that was passed to the constructor.

getFile() Get the file in which the exception was generated.

getLine() Get the line number at which the exception was generated.

getTrace() Get a multidimensional array tracing the method calls that exception, including method, class, file, and argument data.

getTraceAsString() Get a string version of the data returned by getTrace().

__toString() Called automatically when the Exception object is used
Returns a string describing the exception details.

10. Le mécanisme

PHP 5 permet d'utiliser le mécanisme de try/throw/catch

```
class SQLException extends Exception {
    public $problem;
    function __construct($problem) {
        $this->problem = $problem;
    }
}
try {
    ...
    throw new SQLException("Couldn't connect to database");
    ...
} catch (SQLException $e) {
    print "Caught an SQLException with problem $obj->problem";
} catch (Exception $e) {
    print "Caught unrecognized exception";
}
```

10. Lever une exception

```
if ( xxx ) {
    throw new Exception( "yyy" );
}
```

11. Intercepteurs

Métodos :

__get(\$property) Invoked when an undefined property is accessed

__set(\$property, \$value) Invoked when a value is assigned to an undefined property

__isset(\$property) Invoked when isset() is called on an undefined property

__unset(\$property) Invoked when unset() is called on an undefined property

`__call( $method, $arg_array )` Invoked when an undefined method is called

12. Lever une exception

```
function __construct( $file ) {  
    $this->file = $file;  
    if ( ! file_exists( $file ) ) {  
        throw new Exception( "file '$file' does not exist" );  
    }  
    $this->xml = simplexml_load_file($file);  
}
```