

Complément sur les bases du php

Le transtypage : règles de conversion

- Chaîne de caractères \Rightarrow Nombres

```
echo "3" + 1;    //Affiche 4
echo 1 + "3 petits cochons";    //Affiche 4
echo 1 + "-1.3e3";    //Affiche -1299
echo 1 + "marcel";    //Affiche 1
```

- Tous types \Rightarrow Booléen
 - false :
 - Constante FALSE et NULL ;
 - Chaîne vide ou contenant juste un zéro;
 - Nombre zéro ;
 - Tableau vide ;
 - Objet avec aucun champ défini
 - true : les autres données.

Le transtypage : règles de conversion

- Tous types \Rightarrow Chaîne de caractères
 - Depuis booléen :
 - true \rightarrow "1"
 - false \rightarrow "" et ~~non pas le caractère 0~~
 - Depuis un entier : représentation classique en base dix.
 - Depuis Tableaux et objets : affichage du type Array ou Object.

Le transtypage : forcer une conversion

- En le spécifiant :

```
$var = (integer) "12"; // $var contient un nombre  
$var = (string) 12;   // $var contient une chaîne
```

- En utilisant la fonction *settype* (... , ...) :

```
$var = 12;  
settype($var, 'integer'); // $var contient un nombre  
settype($var, 'string');  // $var contient une chaîne
```

- En utilisant les fonctions dédiées :
 - Pour les nombres : *intval(...)*, *floatval(...)*, *doubleval(...)*
 - Pour les chaîne de caractères : *strval(...)*

```
$var = intval("12"); // $var contient le nombre 12  
$var = strval(12);  // $var contient la chaîne "12"
```

Compléments sur les tableaux

- Quelques fonctions pour le parcours de tableaux

`reset($tab)` : place le pointeur sur le premier élément

`current($tab)` : retourne la valeur de l'élément courant

`next($tab)` : place le pointeur sur l'élément suivant

`prev($tab)` : place le pointeur sur l'élément précédant

- Exemple :

```
20 <?php
21 $colors = array("red", "green", "blue");
22 $nbr = count($colors);
23 reset($colors);
24
25 for($i=1; $i<=$nbr; $i++)
26 {
27     echo current($colors). '<br />';
28     next($colors);
29 }
30 ?>
```

red
green
blue

Gestion des fichiers(1)

- Quelques fonctions:

fopen(\$file [, \$mode]) : ouverture du fichier identifié par son nom **\$file** et dans un mode **\$mode** particulier, retourne un identificateur **\$fp** de fichier ou **FALSE** si échec

fclose(\$fp) : ferme le fichier identifié par le **\$fp**

fgets(\$fp, \$length) : lit une ligne de **\$length** caractères au maximum

fgetc(\$fp) : lit un caractère

fputs(\$fp, \$str) : écrit la chaîne **\$str** dans le fichier identifié par **\$fp**

feof(\$fp) : teste la fin du fichier

file_exists(\$file) : indique si le fichier **\$file** existe

filesize(\$file) : retourne la taille du fichier **\$file**

filetype(\$file) : retourne le type du fichier **\$file**

unlink(\$file) : détruit le fichier **\$file**

copy(\$source, \$dest) : copie le fichier **\$source** vers **\$dest**

readfile(\$file) : affiche le fichier **\$file**

rename(\$old, \$new) : renomme le fichier **\$old** en **\$new**

file (\$file) : retourne le fichier **\$file** dans un tableau. Chaque élément du tableau correspond à une ligne du fichier, et les retour-chariots sont placés en fin de ligne.

Gestion des fichiers(2)

- Exemples d'ouverture d'un fichier :

```
$fp = fopen("../fichier.txt","r");  
    // "r" => ouverture en lecture seul
```

```
$fp = fopen("ftp://iutbayonne.univ-pau.fr./pub/fichier.txt","w");  
    // "w" => ouverture en écriture seul
```

```
$fp = fopen("http://www.nexen.com/fichier.txt","a");  
    // "a" => ouverture en écriture avec ajout du contenu à la  
    // fin du fichier
```

Gestion des fichiers(3)

- Exemple lecture d'un fichier :

// ouverture en lecture

```
31 <?php
32 if ($monfichier=@fopen("data.txt",'r'))
33 {
34     while (!feof($monfichier))
35     { // 255 caractères max. ou bien fin de ligne.
36         $ligne = fgets($monfichier,255);
37         echo $ligne.'<br />';
38     }
39     fclose($monfichier);
40 }
41 else
42     {die("impossible d'ouvrir le fichier");}
43 ?>
```

Attention : le caractère @ sert à "échapper" le message d'erreur php

Warning: fopen(data.txt): failed to open stream: No su
impossible d'ouvrir le fichier

Gestion des fichiers(3)

- Exemple d'écriture dans un fichier :

// ouverture en écriture

```
<?php
if ($monfichier=@fopen("monfichier.txt", "w"))
{
    // on écrit deux lignes
    fputs($monfichier,"ligne 1");
    fputs($monfichier,"ligne 2");
    fclose($monfichier);
}
else
{die("impossible d'ouvrir le fichier");}
?>
```

Accès aux dossiers(1)

Il est possible de parcourir les répertoires grâce à ces quelques fonctions :

- chdir(\$str)** : Change le dossier courant en \$str. Retourne TRUE si succès, sinon FALSE.
- getcwd()** : Retourne le nom du dossier courant (en format chaîne de caractères).
- opendir(\$str)** : Ouvre le dossier \$str, et récupère un pointeur \$d dessus si succès, FALSE sinon et génère alors une erreur PHP qui peut être échappée avec @.
- closedir(\$d)** : Ferme le pointeur de dossier \$d.
- readdir(\$d)** : Lit une entrée du dossier identifié par \$d. C'est-à-dire retourne un nom de fichier de la liste des fichiers du dossier pointé. Les fichiers ne sont pas triés. Ou bien retourne FALSE s'il n'y a plus de fichier.
- rewinddir(\$d)** : Retourne à la première entrée du dossier identifié par \$d.

Accès aux dossiers(2)

- Exemple :

```
<?php
if ($dir=@opendir('.')) // ouverture du dossier courant
{
    while($file=readdir($dir)) // lecture d'une entrée
    { echo $file.'<br />'; } // affichage du nom de fichier

    closedir($dir); // fermeture du dossier
}
?>
```

\$dir est un pointeur vers la ressource dossier

\$file est une chaîne de caractères qui prend pour valeur chacun des noms de fichiers retournés par readdir()

PHP et les Images(1)

- **Formats supportés**
JPG, GIF, PNG
- **Fonctionnalités**
Récupération, Création, Modification, Manipulation,

Afin que votre navigateur sache que ce sont des images qu'il va recevoir, il est nécessaire que la première information qu'il reçoit soit le type de l'image.

```
<?PHP
    header("Content-type:image/jpeg"); // pour les fichiers jpeg
    header("Content-type:image/gif");  // pour les fichiers gif
    header("Content-type:image/png");  // pour les fichiers png
?>
```

Attention au header (voir remarque diapo précédente)

PHP et les Images(2)

imagecreate(int *x_size*, int *y_size*) Crée une nouvelle image
retourne un identifiant d'image représentant une image vide(sans couleur) de
largeur *x_size* et longueur *y_size*.

imagecreatetruecolor(int *x_size*, int *y_size*) crée une image noir.

imagedestroy (identifiant image).

libérer l'espace mémoire une fois que l'image crée est terminée

La destruction d'une image n'implique pas sa disparition à l'écran.

imagecolorallocate (resource image, int red, int green, int blue)

Alloue une couleur pour une image et retourne un entier représentant la couleur.

Appelé la première fois cette fonction affecte une couleur au fond de l'image.

imagepng (resource image [, string filename])

Envoie une image PNG vers un navigateur ou un fichier.

Idem pour imagegif, imagejpeg

PHP et les Images(3)

- Un exemple :

```
1 <?php
2 header ("Content-type: image/png");
3 $im = @imagecreate (100, 50)
4     or die ("Impossible d'initialiser la bibliothèque GD");
5 imagecolorallocate ($im, 100, 100, 255);
6 $text_color = imagecolorallocate ($im, 0, 0, 91);
7 imagestring ($im, 2, 5, 5, "un Simple Texte ", $text_color);
8 imagepng ($im);
9 imagedestroy($im);
10 ?>
```



Un texte Simple

La Programmation Orientée Objet(4)

- Quelques fonctions :

- `get_declared_classes()` : retourne un tableau listant toutes les classes définies
- `class_exists($str)` : vérifie qu'une classe dont le nom est passé en argument a été définie
- `get_class($obj)`, `get_parent_class` : retournent le nom de la classe de l'objet `$obj`
- `get_class_methods($str)` : retourne les noms des méthodes de la classe `$str` dans un tableau
- `get_class_vars($str)` : retourne les valeurs par défaut des attributs de la classe `$str` dans un tableau associatif
- `get_object_vars($obj)` : retourne un tableau associatif des attributs de l'objet `$obj` les clés sont les noms des attributs et les valeurs, celles des attributs si elles existent
- `is_subclass_of($obj,$str)` : détermine si l'objet `$obj` est une instantiation d'une sous-classe de `$str`, retourne VRAI ou FAUX
- `method_exists($obj,$str)` : vérifie que la méthode `$str` existe pour une classe dont `$obj` est une instance, retourne VRAI ou FAUX