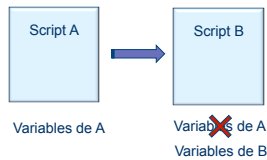


## Cookie / Session

- Durée de vie des variables limitée au script

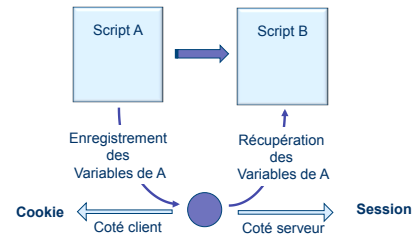


71

S. Lanquétin - L3 : Introduction Technologie pour le Web

## Cookie / Session

- Durée de vie des variables limitée au script



72

S. Lanquétin - L3 : Introduction Technologie pour le Web

## Cookies(I)

- **setcookie()** : définit un cookie envoyé avec entêtes ; true/false
    - Nom
    - Valeur
    - Temps avant expiration depuis 1/1/1970 → time()+nbsecondes
    - Chemin où il est disponible : '/' → sur le domaine, '/rep/' → dans rep
    - Domaine .adresse.fr
    - secure: true → positionné ssi connexion sécurisée (https)  
**\$\_SERVER["HTTPS"]**
    - httponly : true → pas accessibles aux langages de scripts
    - **A indiquer avant tout code HTML**
- ```
bool setcookie ( string $name [, string $value [, int $expire = 0 [, string $path [, string $domain [, bool $secure = false [, bool $httponly = false ]]]]] )
```
- accessible dans **\$\_COOKIE**
  - valeurs de cookies aussi dans **\$\_REQUEST**

73

## Cookies(II)

- Envoi d'un Cookie

```
<?php  
$value = 'test';  
setcookie("TestCookie", $value, time()+3600); //expire dans 1 heure  
?>
```

- Afficher un Cookie

```
<?php  
echo $_COOKIE["TestCookie"];  
?>
```

- Effacer un Cookie

```
<?php  
setcookie("TestCookie", $value, time()-3600);  
?>
```

74

## Sessions (I)

- Moyen de sauvegarder et de modifier des variables
- Permet de sécuriser un site, d'espionner le visiteur
- Informations de sessions conservées en local sur le serveur  
Identifiant de session (cookie, URL)
- **\$\_SESSION** : tableau associatif des valeurs stockées dans les sessions
- Lancer la session avant tout code HTML

75

## Sessions (II)

- **session\_start()** : démarre une session

```
<?php  
session_start();  
if (!isset($_SESSION['compteur'])) {  
    $_SESSION['compteur'] = 0;  
} else {  
    $_SESSION['compteur']++;  
}  
?>
```

- **unset()** : détruit la ou les variables dont le nom est passé en param

```
<?php  
session_start();  
unset($_SESSION['compteur']);  
?>
```

76

## Sessions (III)

- **session\_destroy()** : détruit les données de session et ferme la session
  - Ne détruit pas les variables globales ni le cookie de session
  - Avant : effacer l'identifiant de session et l'éventuel cookie de session

```
<?php
session_start();
$_SESSION = array(); // Détruit toutes les variables de session
// Si vous voulez détruire complètement la session, effacez également le cookie de session
session_destroy(); // Finalement, on détruit la session
?>
```

77

## Sessions (IV)

- **session\_write\_close()** : ferme une session sans la détruire
- **session\_name('nom')** : donne un nom à une session

1 seule session à la fois!

```
<?php
session_name('utilisateur');
session_start();
...
session_write_close(); // Fermeture de la session
session_name('administrateur');
session_start();
... // Accès en lecture à $_SESSION de la session utilisateur
?>
```

78

## Fichiers (I)

- **fopen(\$file, \$mode)** : ouverture du fichier, retourne un identificateur **\$fp**
- Les modes d'ouverture :
- 'r' (lecture seule), 'r+' (lecture et écriture),
  - 'w' (création et écriture seule), 'w+' (création et lecture/écriture),
  - 'a' (création et écriture seule ; place le pointeur de fichier à la fin du fichier),
  - 'a+' (création et lecture/écriture ; place le pointeur de fichier à la fin du fichier)

```
$fp = fopen("../docs/faq.txt", "r");
$fp = fopen("http://www.php.net/", "r");
$fp = fopen("ftp://user:password@toto.fr", "w");
```

79

## Fichiers (II)

Quelques fonctions:

- **fclose(\$fp)** : ferme le fichier identifié par le **\$fp**
- **fgets(\$fp, \$length)** : lit une ligne de **\$length** caractères au maximum
- **fwrite(\$fp, \$str)** : écrit la chaîne **\$str** dans le fichier identifié par **\$fp**
- **fgetc(\$fp)** : lit un caractère
- **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**

80

## Fichiers (III)

- Exemple typique d'affichage du contenu d'un fichier :

```
<?php
$file = "fichier.txt";
$str="";
if( $fd = fopen($file, "r") ) { // ouverture du fichier en lecture
    while ( ! feof($fd) ) { // teste la fin de fichier
        $str .= fgets($fd, 1024);
    } // lecture jusqu'à fin de ligne où des 1024 premiers caractères "*"
    fclose($fd); // fermeture du fichier
    echo $str; // affichage
} else {
    exit("Ouverture du fichier <strong>$file</strong> impossible.");
}
?>
```

81

## Accès aux dossiers (I)

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).

```
<?php
// dossier courant
echo getcwd() . "\n";
chdir('cvs');
// dossier courant
echo getcwd() . "\n";
?>
```

82

## Accès aux dossiers (I)

Quelques fonctions :

- **opendir(\$str)** : Ouvre le dossier **\$str**, et récupère un pointeur **\$d** dessus si succès, FALSE sinon
- **closedir(\$d)** : Ferme le pointeur de dossier **\$d**.
- **readdir(\$d)** : Lit une entrée du dossier identifié par **\$d** ou bien retourne FALSE s'il n'y a plus de fichier.
- **rewinddir(\$d)** : Retourne à la première entrée du dossier identifié par **\$d**.

83

## Accès aux dossiers (III)

- Exemple :

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

84

## URL (I)

- URL (*Uniform Resource Location*)
- Exemples d'URL:  
<http://www.google.fr/?q=cours+php>  
<http://www.manuelphp.com/php/manual.php>
- Pas d'espace
- lettres, chiffres - \_ .
- Autres caractères codés : %**xx**.  
**xx** code hexadécimal.

85

## URL (II)

Exemple par l'URL :

```
<a href="index.php?imprim=yes&user_id=75">Version imprimable</a>
```

- 2 variables transmises au script **index.php**
- ? => séparation paramètres et nom de fichier.
- = sépare nom de paramètre et valeur transmise.
- & sépare deux paramètres.

86

## URL (III)

- **urlencode(\$str)** : code la chaîne **\$str**.  
Caractères interdits remplacés par leur code hexadécimal.  
et espace remplacé par +  
Exemple 1 :

```
echo " <a href='\"$_SERVER[\"PHP_SELF\"]?toto=\".urlencode($toto).\"'>
Toto</a>";
```

- **rawurlencode(\$str)** : code la chaîne **\$str**.  
Caractères interdits remplacés par leur code hexadécimal.  
et espace remplacé par %20  
Exemple 2 :

```
echo " <a href='\"$_SERVER[\"PHP_SELF\"]?toto=\".rawurlencode($toto).\"'>
Toto</a>";
```

- Décoder avec la fonction réciproque adéquate

87

## URL (IV)

Quelques fonctions de codage sur l'URL :

Codage de base :

- **urlencode** : Encode une chaîne en URL.
- **urldecode** : Décode une chaîne encodée URL.

Codage complet :

- **rawurlencode** : Encode une chaîne en URL, selon la RFC3986.
- **rawurldecode** : Décode une chaîne URL, selon la RFC3986.

Codage plus évolué :

- **base64\_encode** : Encode une chaîne en MIME base64.
- **base64\_decode** : Décode une chaîne en MIME base64

88

## URL (V)

- **parse\_url(\$str)** : retourne un tableau associatif contenant les différents éléments de l'URL passée en paramètre.

Exemple :

```
$tab = parse_url("http://www.cia.gov:8080/form.php?var=val");
```

| Champ  | Valeur      |
|--------|-------------|
| scheme | http        |
| host   | www.cia.gov |
| port   | 8080        |
| path   | form.php    |
| query  | var=val     |

89

## URL (VII)

- **parse\_str(\$str,\$tab)** : analyse la chaîne \$str comme si c'était une URL et en extrait les variables et valeurs respectives qui seront alors connues dans la suite du script
- Possibilité de stocker les variables sous forme d'index de tab

Exemple :

```
$str = "nom=jean+pierre&email[]=moi@ici.fr&email[]=moi@labas.com";  
//sans le paramètre $tab  
parse_str($str);  
echo $nom."<br>".$email[0]."<br>".$email[1];  
//avec le paramètre $tab  
parse_str($str,$tab);  
echo $tab["nom"]."<br>".$tab["email"][0]."<br>".$tab["email"][1];
```

jean pierre  
moi@ici.fr  
moi@labas.com

90

## Expressions régulières

## Expressions régulières

- Objectif : traitements complexes sur les chaînes de caractères
- Exemples d'applications :
  - vérifier la validité des chaînes de caractères (email, entrée chiffrée, entrée texte, etc...)
  - extraire des parties bien précises d'une chaîne, d'un texte, d'une page, etc...
  - rendre des emails et url cliquables dans vos textes
  - gérer des éléments spécifiques placés entre vos balises perso, mise en forme, style spécial, ...

92

S. Lanquetin - L3 : Introduction Technologie pour le Web

## Expressions régulières

- Expression rationnelle = *Expression décrivant un langage pouvant être obtenu à partir des mots et des opérations de produit d'ensembles, d'itération et d'union.*
- Expression régulière (Regular expression = REGEXP) = la même chose avec beaucoup plus de fonctionnalités (mais même puissance)
- Au lieu de langage, on parle de motif (pattern)
- Éléments de base pour définir un motif
  - littéraux : a, toto,...
  - | : alternative entre deux motifs
  - \* : 0, 1 ou plusieurs occurrences de ce qui précède

93

S. Lanquetin - L3 : Introduction Technologie pour le Web

## Exemples

- **p(ai|i)n**
  - barre verticale = alternative
  - Tout ce qui s'écrit **pain ou pin**
  - Les parenthèses délimitent la portée de l'alternative
- **a(ba)\*|ca\***
  - tous les mots a ou aba ou abababa...ba ou c ou ca ou caaa...a

94

S. Lanquetin - L3 : Introduction Technologie pour le Web

## Expressions régulières

- Symboles quantificateurs
  - \* : 0, 1 ou plusieurs occurrences de ce qui précède
  - + : 1 ou plusieurs occurrences de ce qui précède
  - ? : 0 ou 1 occurrence de ce qui précède
- Intervalles de reconnaissance
  - **a{3}** : **aaa**
  - **a{2,}** : minimum deux a consécutifs **aa, aaa,...**
  - **a{2,4}** : **aa, aaa, aaaa**

95

S. Lanquétin - L3 : Introduction Technologie pour le Web

## Expressions régulières

- Autres symboles
  - ^ : début de chaîne
  - \$ : fin de chaîne
  - . : n'importe quel caractère (une seule fois)
- Exemple
  - **^(De|A):@**  
Tout ce qui commence par **De:@** ou **A:@**
  - **.?**  
N'importe quel caractère 0 ou 1 fois

96

S. Lanquétin - L3 : Introduction Technologie pour le Web

## Classes de caractères

- **[...]** : choix entre les caractères indiqués entre crochets.
  - **a[rm]e** : **are** ou **ame**,
  - **<h[123456]>** : balises de titre **<h1>** **<h2>** **<h3>**, etc...
- **[...-...]** : tiret = intervalle à l'intérieur de la classe (métacaractère)
  - **<h[1-6]>** : **<h[123456]>**
  - **[a-zA-Z]** : caractères minuscules et majuscules
  - **[-0-9]** : tiret ou chiffres de 0 à 9

97

S. Lanquétin - L3 : Introduction Technologie pour le Web

## Classes complémentées

- **[^...]** : tout sauf ce qui est énuméré ensuite
  - **[^1]** : tout sauf le chiffre 1
  - **[^1-6]** : tout sauf les chiffres de 1 à 6
  - **^[^0-9]** : tout ce qui ne commence pas par un chiffre

98

S. Lanquétin - L3 : Introduction Technologie pour le Web

## Classes prédéfinies

Il existe des séquences types :

- **[[:alnum:]]** : [A-Za-z0-9] – caractères alphanumériques
- **[[:alpha:]]** : [A-Za-z] – caractères alphabétiques
- **[[:digit:]]** : [0-9] – caractères numériques
- **[[:blank:]]** : espaces ou tabulation
- **[[:xdigit:]]** : [0-9a-fA-F] – caractères hexadécimaux
- **[[:graph:]]** : caractères affichables et imprimables
- **[[:lower:]]** : [a-z] – caractères minuscules
- **[[:upper:]]** : [A-Z] – caractères majuscules
- **[[:punct:]]** : caractères de ponctuation
- **[[:space:]]** : tout type d' espace
- **[[:cntrl:]]** : caractères d' échappement
- **[[:print:]]** : caractères imprimables sauf ceux de contrôle

99

## Classes abrégées

| Classe abrégée | Correspondance                               |
|----------------|----------------------------------------------|
| \d             | [0-9]                                        |
| \D             | [^0-9]                                       |
| \w             | [a-zA-Z0-9_]                                 |
| \W             | [^a-zA-Z0-9_]                                |
| \t             | Tabulation                                   |
| \n             | Nouvelle ligne                               |
| \r             | Retour chariot                               |
| \s             | Espace blanc (correspond à \t \n \r)         |
| \S             | Ce qui n'est PAS un espace blanc (!\t \n \r) |
| .              | Tout (Classe universelle)                    |

100

S. Lanquétin - L3 : Introduction Technologie pour le Web

101

S. Lanquétin - L3 : Introduction Technologie pour le Web

- 2 types de regex :
  - POSIX
  - PCRE (perl plus rapide)
    - À utiliser avec les fonctions preg\_...
    - Délimiteurs autour de la regex (#)
    - Création de variables \$1 \$2... avec des () dans la regex pour réutiliser des morceaux de chaînes dans le remplacement
    - \$0 = chaîne entière

102

S. Lanquétin - L3 : Introduction Technologie pour le Web

- L'option la plus utilisée :
  - i pour ne pas faire de différence entre minuscule et majuscule
- (' délimiteur | motif | délimiteur, options ')
- Échappement des métacaractères \
 

```
# ! ^ $ ( ) [ ] { } ? + * . \ |
```

103

### Résumé

Les motifs peuvent être très complexes et contenir des caractères spéciaux.

- [abcdef] : liste de caractères autorisés
- [a-f] : plage de caractères
- [^0-9] : exclusion des caractères de '0' à '9'
- ^ : le motif suivant doit apparaître en début de chaîne
- \^ : recherche du caractère '^' que l'on déspecialise par l'antislash \
- . : remplace un caractère
- ? : rend facultatif le caractère qu'il précède
- | : occurrence de l'élément situé à gauche de cet opérateur ou de celui situé à droite
- + : indique que le caractère précédent peut apparaître une ou plusieurs fois
- \* : pareil que + Mais le caractère précédent peut ne pas apparaître du tout
- {i,j} : retrouve une chaîne contenant entre au minimum i et au maximum j fois le motif qu'il précède
- {i,} : idem mais pas de limite maximum
- {i} : retrouve une séquence d'exactement i fois le motif qu'il précède
- \$ : chaîne finissant par...

104

### PREG

Fonctions :

- preg\_match(\$motif, \$str) : occurrence de \$motif dans la chaîne \$str : renvoi un entier (0 ou 1)
- preg\_match\_all(\$motif, \$str) : nbre d'occurrences de \$motif dans la chaîne \$str : renvoi un entier
- preg\_replace(\$motif, \$newstr, \$str) : remplace les occurrences de \$motif dans \$str par la chaîne \$newstr
- preg\_split(\$motif, \$str) : retourne un tableau des sous-chaînes de \$str délimitées par les occurrences de \$motif

105

### Exemple

- Exemples de motifs PCRE :
  - "[A-Z]#" : recherche toutes les majuscules
  - "[a-zA-Z]#" : recherche toutes les lettres de l'alphabet minuscules ou majuscules
  - "[^aeuyio]#" : exclu les voyelles
  - "#^Le#" : toute chaîne commençant par le mot "Le" suivi d'un espace
  - "#\com\$#" : toute chaîne se terminant par ".com" (déspecialise le point)
- Exemples :
 

```
if ( preg_match("#^.*@wanadoo.fr#", $email) ) {
    echo "Vous êtes chez Wanadoo.";
}
```

```
$email = preg_replace("#@#", "-nospam@", $email);
```
- Ce dernier exemple remplace "moi@ici.fr" en "moi-nospam@ici.fr".

106

### Mail

- La fonction mail envoie un message électronique.

Syntaxe :

```
mail($to, $subject, $message[, $headers, $params]);
```

Exemple :

```
$message = "votre message";
mail("vous@free.fr", "cours PHP", $message);
```

- Retourne true si mail accepté pour livraison false sinon
- Arguments obligatoires : destinataire, sujet du message et message. Entêtes et paramètres additionnels facultatifs (attention : spam).
- Note: programme de messagerie électronique (« mailer ») installé sur le serveur.

## Attention

- Windows :
  - accès direct de PHP à un serveur SMTP
  - . en début de ligne supprimé.
  - remplacez . par un double ..

```
<?php
$text = str_replace("\n.", "\n..", $text);
?>
```

107

S. Lanquétin - L3 : Introduction Technologie pour le Web

## Exemple

```
<?php
// Le message
$message = "Line 1\nLine 2\nLine 3";

// lignes avec plus de 70 caractères découpées en utilisant wordwrap()
$message = wordwrap($message, 70);

// Envoi du mail
mail("toto@u-bourgogne.fr", 'Mon Sujet', $message);
?>
```

108

S. Lanquétin - L3 : Introduction Technologie pour le Web

## Exemple avec entête

```
<?php
$to = 'personne@u-bourgogne.fr';
$subject = 'le sujet';
$message = 'Bonjour !';
$headers = 'From: webmaster@u-bourgogne.fr' . "\r\n" .
'Reply-To: webmaster@u-bourgogne.fr' . "\r\n" .
'X-Mailer: PHP/' . phpversion();

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

109

S. Lanquétin - L3 : Introduction Technologie pour le Web

## Exemple mail html

```
<?php
// Plusieurs destinataires
$to = 'toto@u-bourgogne.fr' . ','; // notez la virgule
$to .= 'titi@u-bourgogne.fr';
// Sujet
$subject = 'Calendrier des anniversaires pour Août';
// message
$message = file_get_contents("entete.php"). '<h1>du code html</h1>'.
file_get_contents("pied.php");
// Pour envoyer un mail HTML, l'entête Content-type doit être définie
$headers = 'MIME-Version: 1.0' . "\r\n";
$headers .= 'Content-type: text/html; charset=iso-8859-1' . "\r\n";
// En-têtes additionnels
$headers .= 'To: Tata <tata@gmail.com>' . "\r\n";
$headers .= 'From: Moi <moi@gmail.com>' . "\r\n";
$headers .= 'Cc: toi@gmail.com' . "\r\n";
$headers .= 'Bcc: lui@gmail.com' . "\r\n";
// Envoi
mail($to, $subject, $message, $headers);
?>
```

110

S. Lanquétin - L3 : Introduction Technologie pour le Web

## Mail html

- Pour envoyer des mails HTML plus complexes, il est recommandé d'utiliser le paquet PEAR [PEAR::Mail\\_Mime](#)

111

S. Lanquétin - L3 : Introduction Technologie pour le Web

## Débogage

- Fonctions d'affichage
  - **echo \$x** : Affichage de la valeur
  - **print \$x** : Affichage de la valeur
  - **var\_export(\$x)** : Affichage de la représentation PHP
  - **print\_r(\$x)** : Affichage du contenu
  - **var\_dump(\$x)** : Affichage du type et du contenu

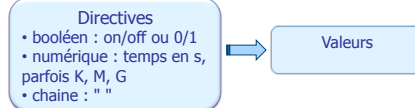
112

S. Lanquétin - L3 : Introduction Technologie pour le Web

## Configuration de PHP

## php.ini

- Fichier dans le répertoire PHP
- Modification avec un éditeur de texte
- Permet de configurer PHP et ses extensions



- Modifications avec httpd.conf, .htaccess, ini\_set()

114

S. Lanquétin - L3 : Introduction Technologie pour le Web

## Recommandations

- short\_open\_tag = Off (<? ?>)
- safe\_mode = Off
- error\_reporting = E\_ALL (niveau d'erreur filtré)
- display\_errors = On dev Off prod  
affichage des erreurs dans la page
- log\_errors = On
- register\_globals = Off (\$nom=\$\_POST['nom'])
- magic\_quotes\_\* =Off

115

S. Lanquétin - L3 : Introduction Technologie pour le Web

## Les classes

## Qu'est ce qu'une classe

Une classe représente un **objet**, avec des variables et des méthodes associées

## Définition d'une Classe

```
<?php
class chien{
    public $nom;
    public function aboyer() {
        echo 'Woof!';
    }
}
?>
```

Définition de la classe chien.

Objet chien :

- 1 attribut : nom
- 1 action : aboyer



## Définition d'une Classe

```
<?php
class chien{
    public $nom;
    public function aboyer() {
        echo 'Ouaf!';
    }
}
?>
```

Nom de la classe.

## Définition d'une Classe

```
<?php
class chien{
    public $nom;
    public function aboyer() {
        echo 'Ouaf!';
    }
}
?>
```

1 Attribut de l'objet :  
nom du chien.

## Définition d'une Classe

```
<?php
class chien{
    public $nom;
    public function aboyer() {
        echo 'Ouaf!';
    }
}
?>
```

1 Action de l'objet (methode):  
aboyer.

## Définition d'une Classe

```
<?php
class chien{
    public $nom;
    public function aboyer() {
        echo 'Ouaf!';
    }
}
?>
```

Fin de la définition de la classe

## Utilisation d'une classe

```
<?php
require 'chien.class.php';
$chiot= new chien();
$chiot->nom= 'Medor';
echo "{$chiot->nom} dit";
$chiot->aboyer();
?>
```

## Utilisation d'une classe

```
<?php
require 'chien.class.php';
$chiot= new chien();
$chiot->nom= 'Medor';
echo "{$chiot->nom} dit";
$chiot->aboyer();
?>
```

Inclus la définition de la classe

## Utilisation d'une classe

```
<?php
require('chien.class.php');
$chiot= new chien();
$chiot->nom= 'Medor';
echo "{$chiot->nom} dit";
$chiot->aboyer();
?>
```

Crée une nouvelle instance de classe

## Utilisation d'une classe

```
<?php
require('chien.class.php');
$chiot= new chien();
$chiot->nom= 'Medor';
echo "{$chiot->nom} dit";
$chiot->aboyer();
?>
```

Met la variable nom à 'Medor'

## Utilisation d'une classe

```
<?php
require('chien.class.php');
$chiot= new chien();
$chiot->nom= 'Medor';
echo "{$chiot->nom} dit";
$chiot->aboyer();
?>
```

Utilisation de la variable nom

## Utilisation d'une classe

```
<?php
require('chien.class.php');
$chiot= new chien();
$chiot->nom= 'Medor';
echo "{$chiot->nom} dit";
$chiot->aboyer();
?>
```

Utilisation de la méthode aboyer()

## Erreur classique

```
$chiot->nom= 'Medor';
```

1 seul \$

```
$chiot->$nom= 'Medor';
```

## Attributs dans la classe

- Pour utiliser les variables de classe dans les fonctions de la classe : **\$this**

```
<?php
class chien{
    public $nom;
    public function aboyer() {
        echo $this->nom.' dit Ouaf!';
    }
}
?>
```

## Constructeur

- **Constructeur** : fonction exécutée dès que la classe est instanciée.
- Création d'un constructeur avec `__construct`
- S'il y a des arguments, ils doivent être passés dès l'instanciation

131

S. Lanquétin - L3 : Introduction Technologie pour le Web

## Exemple de constructeur

Constructeur

```
<?php
class chien{
    public function __construct($texte) {
        $this->nom= $texte;
    }
    public function aboyer() {
        echo $this->nom.' dit Ouaf!';
    }
}
?>
```

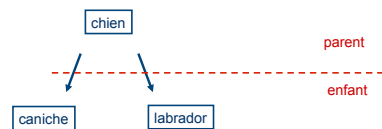
## Exemple de constructeur

```
<?php
...
$chiot= new chien('Medor');
...
?>
```

Les arguments du constructeur sont passés pendant l'instanciation de l'objet.

## Héritage

- Hiérarchie entre les classes



134

S. Lanquétin - L3 : Introduction Technologie pour le Web

## Exemple d'héritage

```
class caniche extends chien{
    public $type;
    public function set_type($taille) {
        if ($taille>38) {
            $this->type = 'Standard';
        } else {
            $this->type = 'Miniature';
        }
    }
}
```

## Exemple d'héritage

```
class caniche extends chien{
    public $type;
    public function set_type($taille) {
        if ($taille>38) {
            $this->type = 'Standard';
        } else {
            $this->type = 'Miniature';
        }
    }
}
```

Le mot clé **extends** indique que la classe caniche est un enfant de la classe chien

## Exemple d'héritage

```
<?php
...
$chiot= new caniche('Oscar');
$chiot->set_type(30); // 30 cm de haut!
echo "Le caniche s'appelle {$chiot->nom}, ";
echo "il est dit de type {$chiot->type}";
echo $chiot->aboyer();
...
?>
```

## Surcharge

```
class caniche extends chien{
    ...
    public function aboyer() {
        echo 'Ouh!';
    }
    ...
}
```

## Constructeurs enfant

- Si la classe enfant a un constructeur, celui de la classe parent est ignoré
- Si la classe enfant n'a pas de constructeur, celui de la classe parent est utilisé
- ...

## Mots clés

- **class** : Déclaration de classe
- **const** : Déclaration de constante de classe
- **function** : Déclaration d'une méthode
- **public/protected/private** : Accès ("public" par défaut)
- **new** : Création d'objet
- **self** : Résolution de portée
- **parent** : Résolution de portée
- **static** : Résolution de portée
- **extends** : Héritage de classe
- **implements** : Implémentation d'une interface (redéclarer toutes les méthodes)

140

S. Lanquétin - L3 : Introduction Technologie pour le Web

## Méthodes prévues

- **\_\_construct()** : Constructeur de la classe
- **\_\_destruct()** : Destructeur de la classe
- **\_\_set()** : Déclenchée lors de l'accès en écriture à une propriété de l'objet
- **\_\_get()** : Déclenchée lors de l'accès en lecture à une propriété de l'objet
- **\_\_call()** : Déclenchée lors de l'appel (non statique) d'une méthode inexistante de la classe
- **\_\_callstatic()** : Déclenchée lors de l'appel (statique) d'une méthode inexistante de la classe

141

S. Lanquétin - L3 : Introduction Technologie pour le Web

## Méthodes prévues

- **\_\_isset()** : Déclenchée si on applique isset() à une propriété de l'objet
- **\_\_unset()** : Déclenchée si on applique unset() à une propriété de l'objet
- **\_\_sleep()** : Exécutée si la fonction serialize() est appliquée à l'objet
- **\_\_wakeup()** : Exécutée si la fonction unserialize() est appliquée à l'objet
- **\_\_toString()** : Appelée lorsque l'on essaie d'afficher directement l'objet : echo \$object;

142

S. Lanquétin - L3 : Introduction Technologie pour le Web

## Méthodes prévues

- **\_\_set\_state()** : Méthode statique lancée lorsque l'on applique la fonction `var_export()` à l'objet
- **\_\_clone()** : Appelé lorsque l'on essaie de cloner l'objet
- **\_\_autoload()** : Cette fonction n'est pas une méthode, elle est déclarée dans le scope global et permet d'automatiser les "include/require" de classes PHP.

143

S. Lanquétin - L3 : Introduction Technologie pour le Web

## Fonctions pour les classes

- 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

144

S. Lanquétin - L3 : Introduction Technologie pour le Web

## Fonctions pour les classes

- Quelques fonctions :
  - **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

145

S. Lanquétin - L3 : Introduction Technologie pour le Web

## Sécurité

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

## Introduction

- Faillies de sécurité dans 8 sites /10
  - Exploitation de données personnelles
  - Exécution de commandes administrateur
  - Modification la logique applicative
  - Planter le site
- Limiter les droits et permissions de l'utilisateur

147

S. Lanquétin - L3 : Introduction Technologie pour le Web

- Validation des données reçues (get, post, entêtes http, cookies...)

Injection SQL

Injection Javascript

- Caractères attendus : convertir et typer les données
- Date : imposer format
- Adresse mail : vérifier contenu @ .
- Limitation du nombre de caractères
- Données épurées : balises html, javascript, php...
- BD considérée comme extérieur
- Liaison BD/serveur web

148

S. Lanquétin - L3 : Introduction Technologie pour le Web

- Toujours tester l'existence, la validité d'un fichier
- Regarder régulièrement les fichiers journaux (logs)
- Contrôle des données côté serveur
- Utiliser des programmes version stable à jour

149

S. Lanquétin - L3 : Introduction Technologie pour le Web

## Code

- Formatage du code
- Vérifier la validité de tous les paramètres : type, caractères autorisés, taille max...
- Utiliser POST au lieu de GET
- Scripts hors ligne
- Filtrer et valider les données lors de leur récupération
  - Valider : vérifier que les données contiennent ce qu'elles doivent (is\_array(), is\_string(), is\_int()...)
- Filtrer : éliminer (htmlspecialchars(), htmlspecialchars())
- Convertir les données dans le format de sortie lors de leur utilisation

150

S. Lanquétin - L3 : Introduction Technologie pour le Web

## Session

- Crypter les données avec TLS (SSL)
- Complexité du mot de passe (imposer...)
- Compteur, délais et seuils pour contrer les attaques de type DOS
- Eviter les identifiants par défaut (root/password)
- Ré-identification pour manipulation importante (paiement, chgmt mot de passe...)
- Crypter l'identifiant de session (session.entropy\_file, session.entropy\_length, session.hash\_function...)

151

S. Lanquétin - L3 : Introduction Technologie pour le Web

## Formulaire : validation des donnée

- Donner une durée de vie au formulaire
  - champ invisible de type date avec une date d'expiration ou de production du formulaire
  - date cryptée (MD5), stockée dans une session

152

S. Lanquétin - L3 : Introduction Technologie pour le Web

## Session

- Cookies : circule si pas sécurisé TLS
- Fin de session
  - Session inactive : inactive depuis 5 à 15 minutes
  - Session longue : session active mais limite maximale de validité atteinte =>réidentification.
  - Deconnexion : possibilité pour l'utilisateur de se déconnecter =>détruire session et cookie de session
  - Erreurs de sécurité : erreur =>fermeture immédiate de toutes les sessions

153

S. Lanquétin - L3 : Introduction Technologie pour le Web

## Organisation du site

- Séparer les zones : zone libre, zone sécurisée...
  - En ligne : code html, php, images...
  - Hors ligne : classes, scripts communs...
  - Hors ligne partagé : bibliothèques php
- En-tête "no-store" =>non stockage des infos dans cache...

```
header('Cache-Control: no-store, no-cache, must-revalidate');
```

```
http://www.mnot.net/cache_docs/index.fr.html
```

154

S. Lanquétin - L3 : Introduction Technologie pour le Web

## Masquer PHP

- Avec un autre langage

```
AddType application/x-httpd-php .asp .py .pl
```

- Avec des types inconnus

```
AddType application/x-httpd-php .bob .toto .111
```

- Sous forme de html

Inconvénient : fichiers HTML analysés et traités par le moteur PHP

```
AddType application/x-httpd-php .htm .html
```