

WEB 1 : Initiation à la programmation WEB





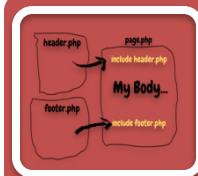
CommitStrip.com

Dev Web / Backend

Dev Web / Backend : Plan



PHP ?



Gestion des fichiers externes (Link)



Comment ça marche ? /
Prérequis



Les Formulaires



Hello World !



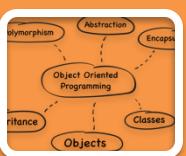
PDO



Un vrai langage de
Programmation...



Sessions & Cookies



Object or not Object ?



Gestion des paquets



Dev Web / Backend : PHP ?

Dev Web / Backend : PHP ?



PHP : Wat iz zat ?



Andi Gutmanns



Rasmus Lerdorf



Zeev Suraski

Rasmus Lerdorf, crée **PHP** en **1994** pour analyser les connexions sur son site Web. Il réalise les 2 premières moutures du langage (v1 et v2).

En **1997**, deux étudiants, **Andi Gutmanns** et **Zeev Suraski**, reprennent le moteur, sortira **PHP 3.0** puis les outils **Zend**.

La première version objet de PHP (la version 4) a été profondément remaniée lors du passage de PHP4.0 à PHP5.0 et s'inspire désormais largement du modèle de **Java**.

La version actuelle de PHP est la **8**. La version **7** est sortie en 2015 et il n'y a jamais eu de **6** !

Dev Web / Backend : PHP ?

PROGRAMMING LANGUAGE ID card

Name
PHP
Type
Interpreted
Birthdate
1994



Platform
Multi-platform
Licence
Open Source : PHP Licence

Langage « **Interprété** » (*Non compilé - Exécuté Instruction par instruction*), spécialisé dans la génération de texte ou de documents :

- **HTML**
- **PDF**
- **Images**

Lecture



HTML ignoré



<?php Code PHP exécuté ?>



HTML et résultats PHP



Affichage par le navigateur

L'interpréteur lit un fichier source **.php** puis génère un flux de sortie avec les règles suivantes :

- Toute ligne située à l'extérieur d'un bloc PHP (*entre <?php et ?>*) est recopiée inchangée dans le flux de sortie
- Le code PHP est interprété et génère éventuellement des résultats intégrés eux aussi au flux de sortie
- Les erreurs éventuelles donnent lieu à des messages d'erreurs qu'on retrouve également dans le flux de sortie (cf. config)
- Une **page html** pure sauvegardée avec l'extension **.php** sera donc non modifiée et renvoyée telle quelle ...

Conçu pour être exécuté par un **serveur HTTP**, il peut fonctionner comme **n'importe quel langage interprété** de façon locale, en exécutant les programmes en ligne de commande

Dev Web / Backend : PHP ?



Beaucoup de **code open source** écrit en PHP circule sur le web

Plusieurs gros **CMS** (Content Management System) utilisent PHP



WORDPRESS



Joomla!™



PrestaShop

De grands **Frameworks** de développement Web aussi sont disponibles en PHP:



Symfony



laravel



phalcon



CakePHP



Slim PHP

Dev Web / Backend : PHP ?

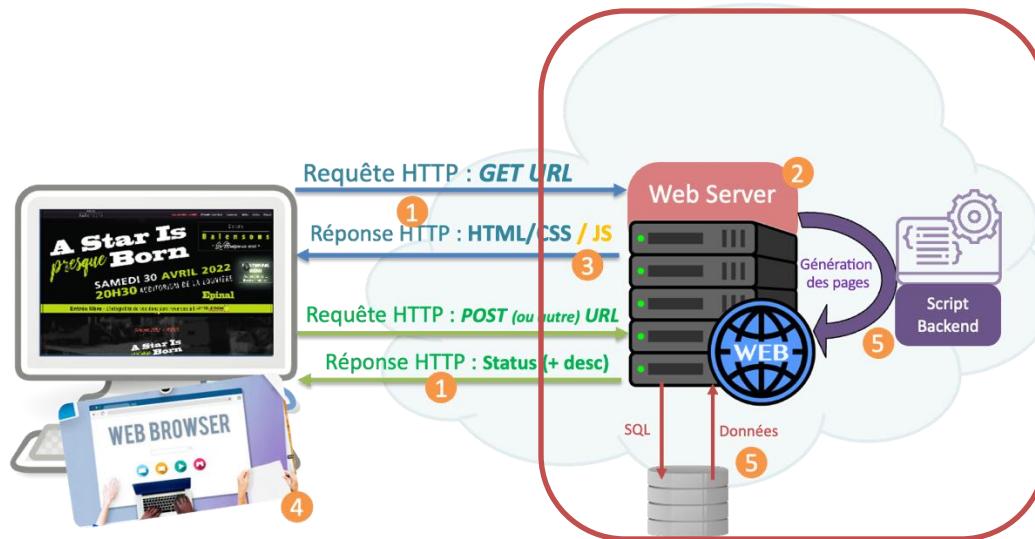


<https://www.php.net/manual/fr/>



Dev Web / Backend : PHP : **Comment ça marche ?**

Dev Web / Backend : PHP : Comment ça marche ?



On fait de la programmation WEB
→ Il nous faut un serveur WEB

PHP est un langage conçu pour être utilisé en Backend sur un Serveur Web

Il est possible de communiquer avec le client et le serveur web sur la même machine en utilisant l'IP localhost

On a à notre disposition un serveur WEB partagé



On va installer un serveur WEB sur nos PCs ...



Dev Web / Backend PHP : Comment ça marche ?

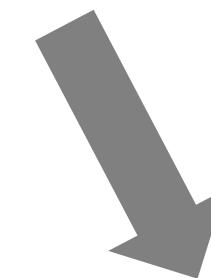
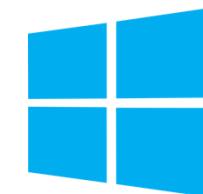
WE HAVE A



LAMP

Linux Apache MySQL PHP

WAMP



MAMP



XAMP





Dev Web / Backend : PHP : **Hello World !**

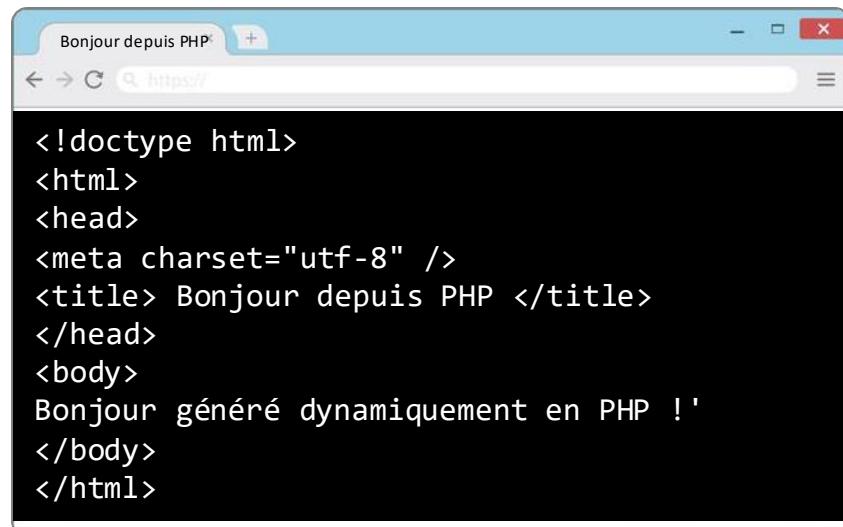
Dev Web / Backend : PHP : Hello World !

EXAMPLES

: Affichage PHP

```
<!doctype html>
<html>
<head>
<meta charset="utf-8" />
<title> Bonjour depuis PHP </title>
</head>
<body>
<?php
    echo 'Bonjour généré dynamiquement en PHP !';
?>
</body>
</html>
```

```
<!doctype html>
<html>
<head>
<meta charset="utf-8" />
<title> Bonjour depuis PHP </title>
</head>
<body>
<?php
    $hello = 'Bonjour généré dynamiquement en PHP !';
    echo $hello ;
?>
</body>
</html>
```

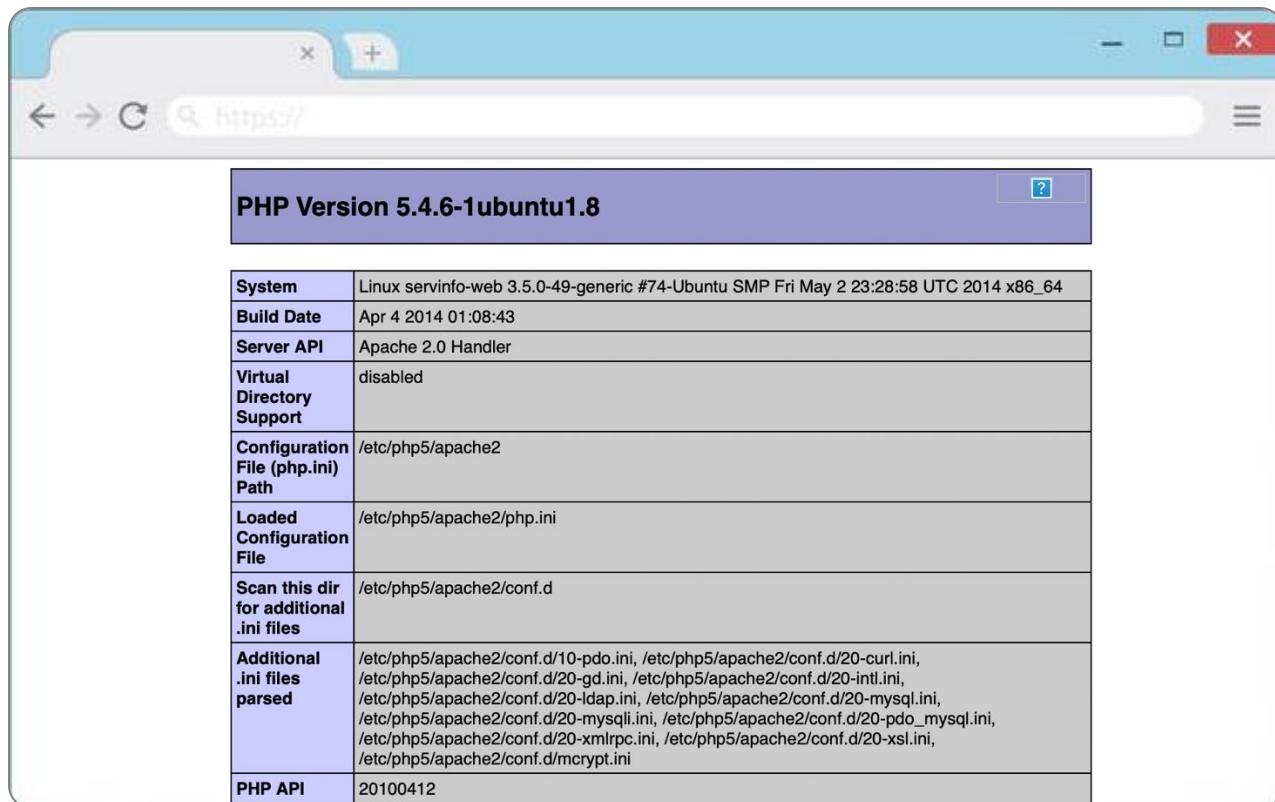


Dev Web / Backend : PHP : Hello World !

EXAMPLES

: phpinfo()

```
<?php phpinfo(); ?>
```



PHP Version 5.4.6-1ubuntu1.8	
System	Linux servinfo-web 3.5.0-49-generic #74-Ubuntu SMP Fri May 2 23:28:58 UTC 2014 x86_64
Build Date	Apr 4 2014 01:08:43
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files parsed	/etc/php5/apache2/conf.d/10-pdo.ini, /etc/php5/apache2/conf.d/20-curl.ini, /etc/php5/apache2/conf.d/20-gd.ini, /etc/php5/apache2/conf.d/20-intl.ini, /etc/php5/apache2/conf.d/20-ldap.ini, /etc/php5/apache2/conf.d/20-mysql.ini, /etc/php5/apache2/conf.d/20-mysqli.ini, /etc/php5/apache2/conf.d/20-pdo_mysql.ini, /etc/php5/apache2/conf.d/20-xmlrpc.ini, /etc/php5/apache2/conf.d/20-xsl.ini, /etc/php5/apache2/conf.d/mcrypt.ini
PHP API	20100412

Dev Web / Backend : PHP : Hello World !

EXAMPLES

: USER_AGENT

```
<!doctype html>
<html>
<head>
<title>
Navigateur
</title>
<body>
Les informations sur le Navigateur sont :
<?php
echo $_SERVER['HTTP_USER_AGENT'];
?>
</body>
</html>
```



Dev Web / Backend : PHP : Hello World !

EXEMPLES

: Imbrication HTML

```
<!doctype html>
<html><head><title>Navigateur</title><body>
Les informations sur le Navigateur sont :
<?php
$AGENT=$_SERVER['HTTP_USER_AGENT'];
echo $AGENT; echo("\n<P>");
if (stristr($AGENT,"MSIE")) {
    ?>
    <b>Vous semblez utiliser Internet Explorer !</b>
    <?php
}
elseif (preg_match("/Firefox/i", $AGENT))
{
    ?>
    <b>Vous semblez utiliser Firefox !</b>
    <?php
}
elseif (preg_match("/chrome/i", $AGENT))
{
    ?>
    <b>Vous semblez utiliser Chrome !</b>
    <?php
}
elseif (preg_match("/Safari/", $AGENT))
{
    ?>
    <b>Vous semblez utiliser Safari !</b>
    <?php
}
else echo "Navigateur Inconnu !";
?>
</P></body></html>
```

ATTENTION: ça peut vite devenir ILLISIBLE



Pour ne pas écrire ce genre de code, quelques solutions :

- Utiliser des fonctions PHP
- Utiliser des Classes et Objets PHP
- Séparer les modèles des Vues
- Séparer les Modèles, les Vues et les Contrôleurs (MVC)
- Utiliser des systèmes de templates comme Twig

S4

```
@var $bootstrapper  
*/  
define('PSI_INTERNAL_XML', false);  
  
if (version_compare("5.2", PHP_VERSION, ">")) {  
    die("PHP 5.2 or greater is required!!!");  
}  
if (!extension_loaded("pcre")) {  
    die("phpSysInfo requires the pcre extension  
        properly.");  
}  
  
require_once APP_ROOT . '/includes/autoload.php';  
  
// Load configuration  
require_once APP_ROOT . '/config.php';  
  
if (!defined('PSI_CONFIG_FILE') || !defined('PSI_DEBUG')) {  
    $tpl = new Template("/templates/html/error_config.html");  
    echo $tpl->fetch();  
    die();  
}  
die();
```

PHP

Hypertext Preprocessor

PHP:

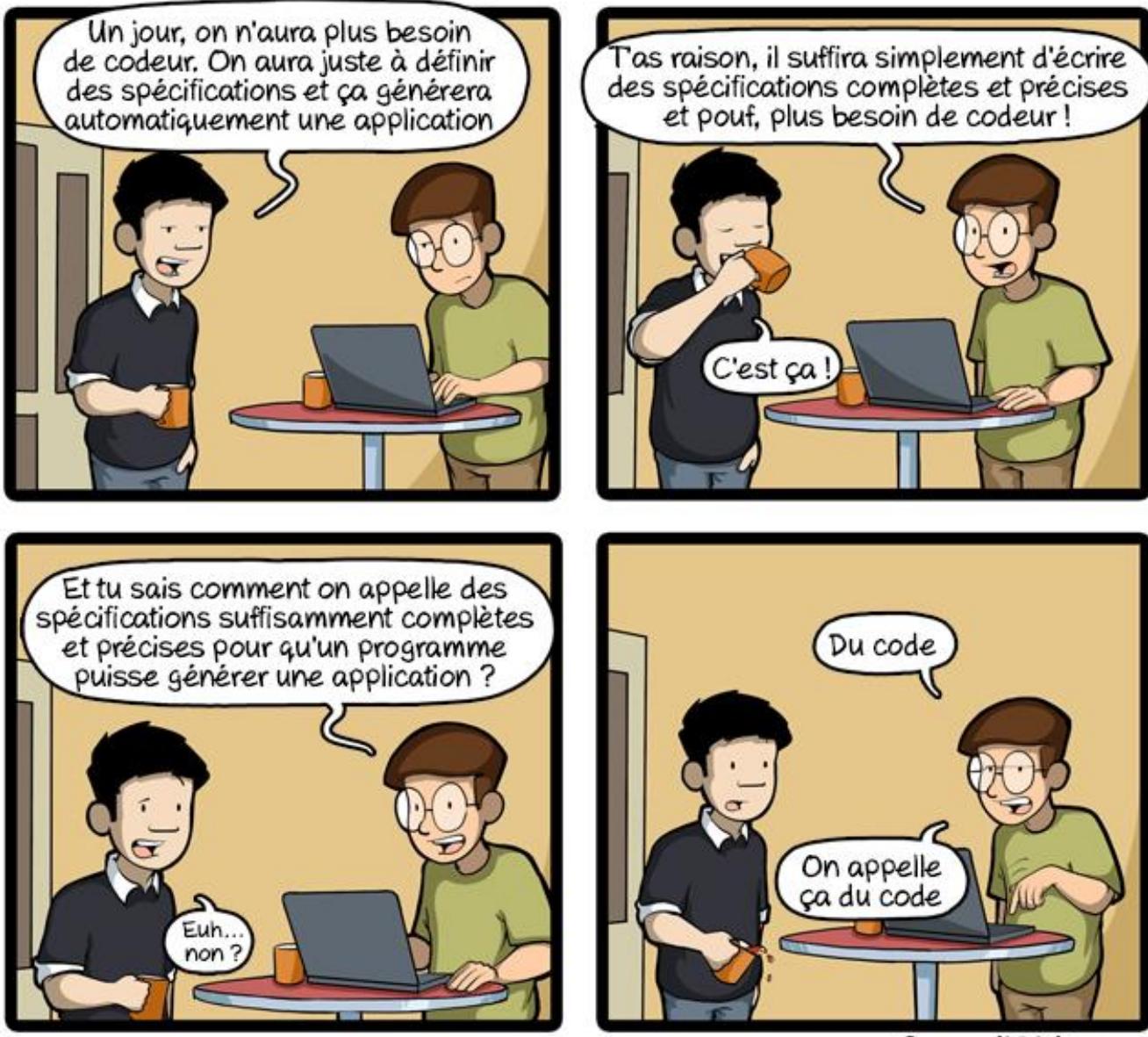
SYNTAX

COMMENTS

VARIABLES

DATA TYPES...

Dev Web / Backend : PHP : Un Langage de Prog.



CommitStrip.com

Dev Web / Backend : PHP : **Un Langage de Prog.**

Généralités / Syntaxe



Oui bon ben PHP, en gros,
c'est du Java avec des « **\$** »
devant les variables et des
« **.** » à la place de « **+** »

- **Proche du C, du Java ou du Perl**
- **Peu structuré au début (pour ne pas dire permissif)**
- **Possibilité de définir des fonctions et des classes**
- **Très bonnes performances pour PHP 7** (*améliorations de l'ordre de 50% de la vitesse d'exécution*)

Les profs de français : "ça c'est un point-virgule, vous l'utiliserez rarement."

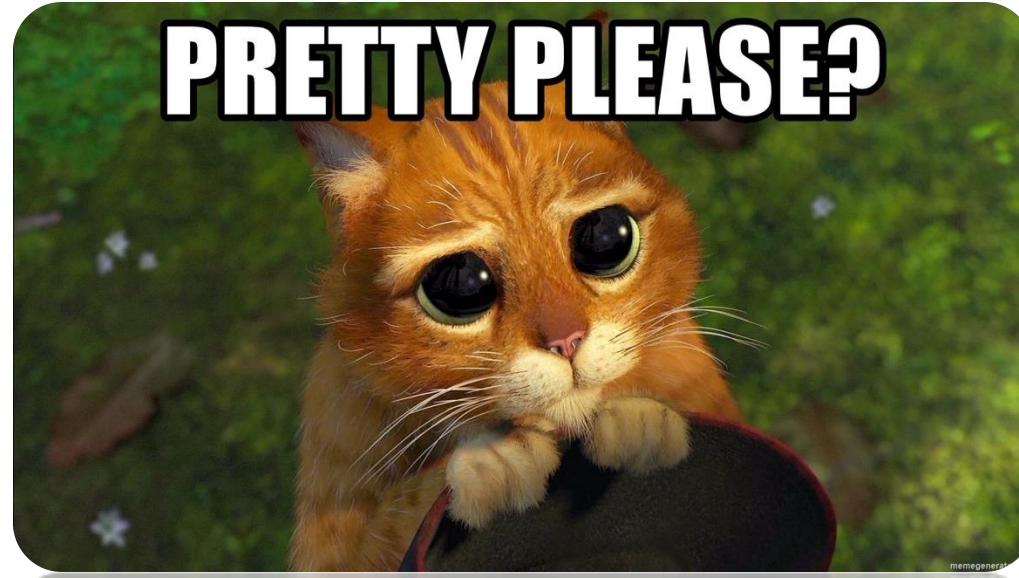


Les développeurs :

//Les commentaires

- Simple ligne : `// Texte ou # Texte`
- Multi-ligne : `/* Texte */`

PRETTY PLEASE?



```
<?php  
echo "Ceci est un test";      // Ceci est un commentaire sur une ligne comme en C.  
                            /* Ceci est un commentaire sur plusieurs lignes,  
                               comme en Java . */  
  
echo "Ceci est encore un test";  
echo "Enfin, le test final";   # Ceci est un commentaire comme en Shell Unix  
?>
```

Dev Web / Backend : PHP : Un Langage de Prog.

Les Variables :

- En PHP, **les variables** sont représentées par le caractère dollar "\$" suivi du nom de la variable.

\$nom_de_la_variable

- Le nom est **sensible** à la casse (i.e. \$x ≠ \$X).
- Un nom de variable valide **doit** commencer par une lettre ou un souligné (_), suivie de lettres, chiffres ou soulignés.



Déclaration simple

```
<?php  
$variable = "une variable PHP";  
// Une autre variable :  
$variable = 1000;
```

??

```
// Récupère la valeur de $_GET['email'] et retourne 'nobody' si elle n'existe pas.  
$mail = $_GET['email'] ?? 'nobody@null';  
// Équivalent à:  
$mail = isset($_GET['email']) ? $_GET['email'] : 'nobody@null';
```

isset() / unset() / isempty()

```
$a = "une variable PHP";  
if(isset($a)) echo "a existe";  
if(!issempty($a)) echo "a existe et n'est pas vide";  
unset($a); echo "a a été supprimée ...";
```

Dev Web / Backend : PHP : Un Langage de Prog.

Typage :

- PHP **ne nécessite pas de déclaration explicite** du type d'une variable.
- Le type d'une variable est déterminé par **le contexte d'utilisation**.

gettype(\$var)

Return :

- **integer**
- **double**
- **string**
- **array**
- **object**
- **class**
- **unknown**

Fonctions de test :

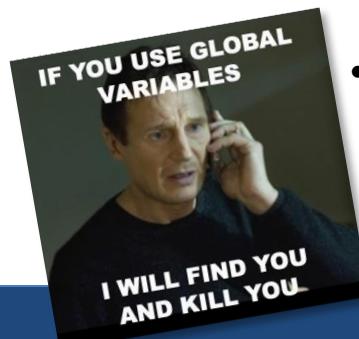
- **is_int()**
- **is_long()**
- **is_double()**
- **is_array()**
- **is_object()**
- **is_string()**

settype(\$var, 'type')

```
$a=3.5;  
settype($a,"integer");  
echo "a = ".$a; // affiche « a = 3 »
```

Portée des variables :

- Par défaut, toutes les variables sont **locales**
- Leur portée se réduit à la fonction ou au bloc de leur déclaration
- Pour déclarer une variable globale, on peut utiliser le tableau **\$_GLOBALS[]**
`<?php $_GLOBALS['MaVar']="Bonjour"; ?>`



Dev Web / Backend : PHP : Un Langage de Prog.

Types de base :

Booléens	<pre><?php \$foo = True; // assigne la valeur TRUE à la variable \$foo ?></pre>
Entiers	<pre><?php \$a = 1234; // nombre entier en base 10 \$a = -123; // nombre entier négatif \$a = 0x1A; //nombre entier en base 16, hexa. (=26 en b10) ?></pre>
NULL	La valeur spéciale « null » représente l'absence de valeur. Une variable avec la valeur NULL n'a pas de valeur. (<i>comme en Java</i>)
Strings	<pre><?php \$firstName = "Jonathan"; \$lastName = "Miller"; \$fullName = \$firstName . ".\$lastName; echo strtoupper(\$fullName), "
"; // JONATHAN MILLER echo substr(\$firstName, 0, 3), "
"; //Jon ?></pre>



Dev Web / Backend : PHP : Un Langage de Prog.

Strings :

Guillemets ou Cotes :

```
<?php  
$var="Hello PHP";  
$machaine="le contenu de \$var est $var<br>";  
echo $machaine; //ou avec des ' '' :  
$mystring='le contenu de $var est '.$var;  
echo $mystring;
```

Accéder à un caractère :

```
<?php echo $chaine[i]; ?>
```

Mettre en majuscules/minuscules:

- **strtoupper()** pour obtenir des majuscules
- **strtolower()** pour mettre en minuscules
- **ucfirst()** pour mettre en majuscule la 1ère lettre d'une chaîne
- **ucwords()** pour mettre en majuscule la 1ère lettre de chaque mot dans une chaîne

La concaténation : « ; »

```
$name = "Yoda";  
echo "Maitre". ". $name;
```

Longueur d'une chaîne :

```
<?php int lg=strlen($chaine); ?>
```

Recherche de sous-chaines ou de motifs dans une chaîne :

- **strstr() / striistr() / ereg() / eregi() / str_contains()** en PHP 8

```
$AGENT=$_SERVER['HTTP_USER_AGENT'];  
if (striistr($AGENT,"MSIE")) {  
    echo "Vous semblez utiliser Internet Explorer !</b>";  
}  
elseif (ereg("Firefox",$AGENT)) {  
    echo "Vous semblez utiliser Firefox !</b>";  
}  
elseif (eregi("chrome",$AGENT))  
    echo "Vous semblez utiliser Chrome !</b>";
```

Dev Web / Backend : PHP : Un Langage de Prog.

Arrays :

Tableaux simples :

```
<?php  
$array = array(1, 2, 3, 4, 5);  
echo $array[0]; // 1  
?>
```

Tableaux associatifs :

```
<?php  
$arr = array("foo" => "bar", 12 => true);  
echo $arr["foo"]; // bar  
echo $arr[12]; // 1  
?>
```

Types :

```
<?php  
$arr = array(5 => 1, 12 => 2);  
$arr[] = 56; // Ceci revient à $arr[13] = 56;  
$arr["x"] = 42; // Ceci ajoute un nouvel élément à  
// l'index "x"  
?>
```

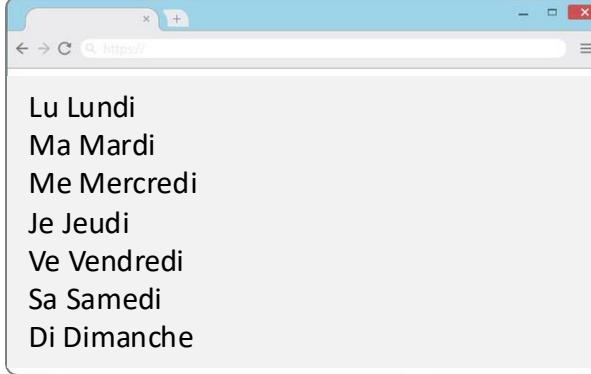


Dev Web / Backend : PHP : Un Langage de Prog.

Arrays :

Parcours des tableaux associatif : foreach :

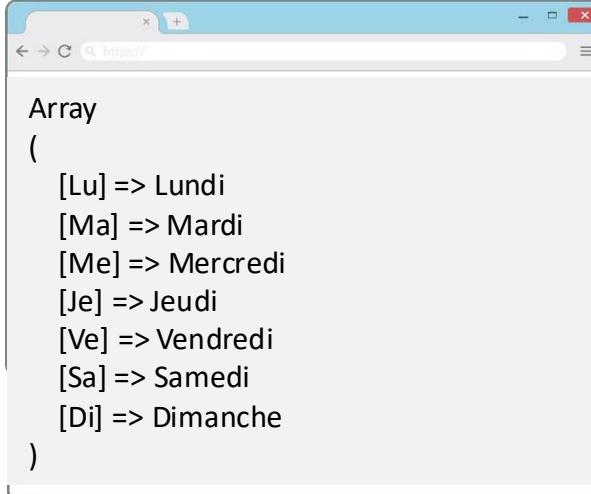
```
<?php  
$jours=array(  
"Lu"=>"Lundi", "Ma"=>"Mardi", "Me"=>"Mercredi", "Je"=>"Jeudi",  
"Ve"=>"Vendredi", "Sa"=>"Samedi", "Di"=>"Dimanche" );  
  
foreach($jours as $key=>$val) echo $key." ".$val."  
\n";
```



Lu Lundi
Ma Mardi
Me Mercredi
Je Jeudi
Ve Vendredi
Sa Samedi
Di Dimanche

print_r() :

```
<?php  
print_r($jours);
```



Array
(
[Lu] => Lundi
[Ma] => Mardi
[Me] => Mercredi
[Je] => Jeudi
[Ve] => Vendredi
[Sa] => Samedi
[Di] => Dimanche
)

Variables : Quelques particularités de PHP :

Valeurs des variables :

```
<?php  
$toto = "Bonjour<br/>\n";  
$var = "toto";  
echo $$var; // Bonjour<br/>
```

eval(\$string)

Permet d'évaluer une instruction php dans une chaîne de caractère.



Faille de sécurité



Les variables Superglobales de PHP :

- **\$_GET[]**, **\$_POST[]** ou **\$_REQUEST[]** qui englobe les 2
- **\$_SERVER[]** : Variables décrivant le client ou la page courante
- **\$_GLOBALS[]** variables globales
- **\$_COOKIE[]** pour les cookies
- **\$_SESSION[]** pour les sessions

Dev Web / Backend : PHP : Un Langage de Prog.

Les Dates :

date() :

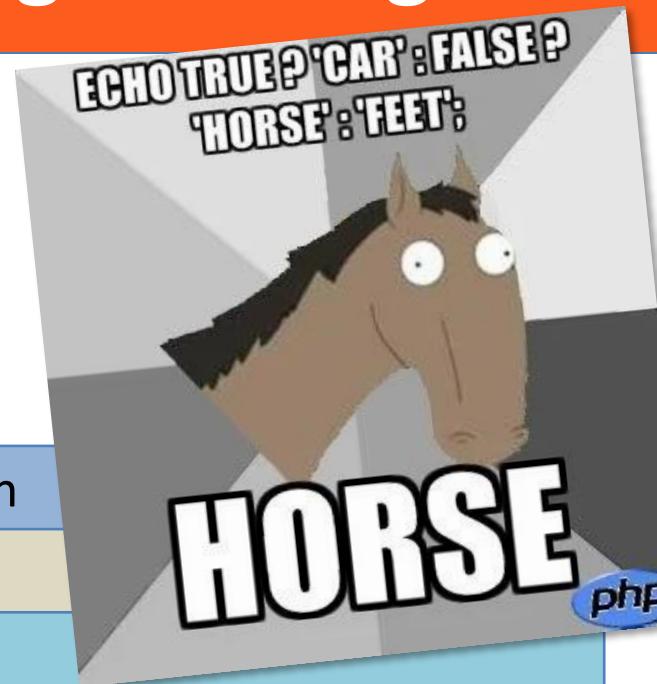
```
<?php  
date_default_timezone_set( 'Europe/Paris' );  
  
echo date( 'd-m-Y' );
```



Manipulations de timestamps → time() :

```
<?php  
$d=strtotime("10:30pm April 15 2014");  
echo "Created date is " . date("Y-m-d h:i:sa", $d);  
?>
```

Les Opérateurs :



++, -- Incrémentation, décrémentation

! Négation

+, -, *, /, % Arithmétique

. Concaténation

<, <=, >, >=, ==, != Comparaison

==, != Comparaison Valeur + Type

AND ou && ET logique

OR ou || OU logique

Les structures de contrôle



Comme en JAVA quoi ...

if

```
<?php  
if (conditional test){  
    // do this;  
}  
else{  
    // do this;  
}  
?>
```

switch

```
<?php  
switch ($variable){  
    case possible result #1 :  
        // do this;  
        break;  
    case possible result #2 :  
        // do this;  
        break;  
    default :  
        // do this;  
}  
?>
```

Les structures de contrôle



Comme en JAVA quoi ...

while

```
<?php  
while (condition is true){  
    // do this;  
}  
?>
```

do while

```
<?php  
do {  
    // do this;  
} while (condition is true);  
?>
```

for

```
<?php  
for (init counter ; conditionel test ; update counter){  
    // do this;  
}  
?>
```

Les structures de contrôle

foreach

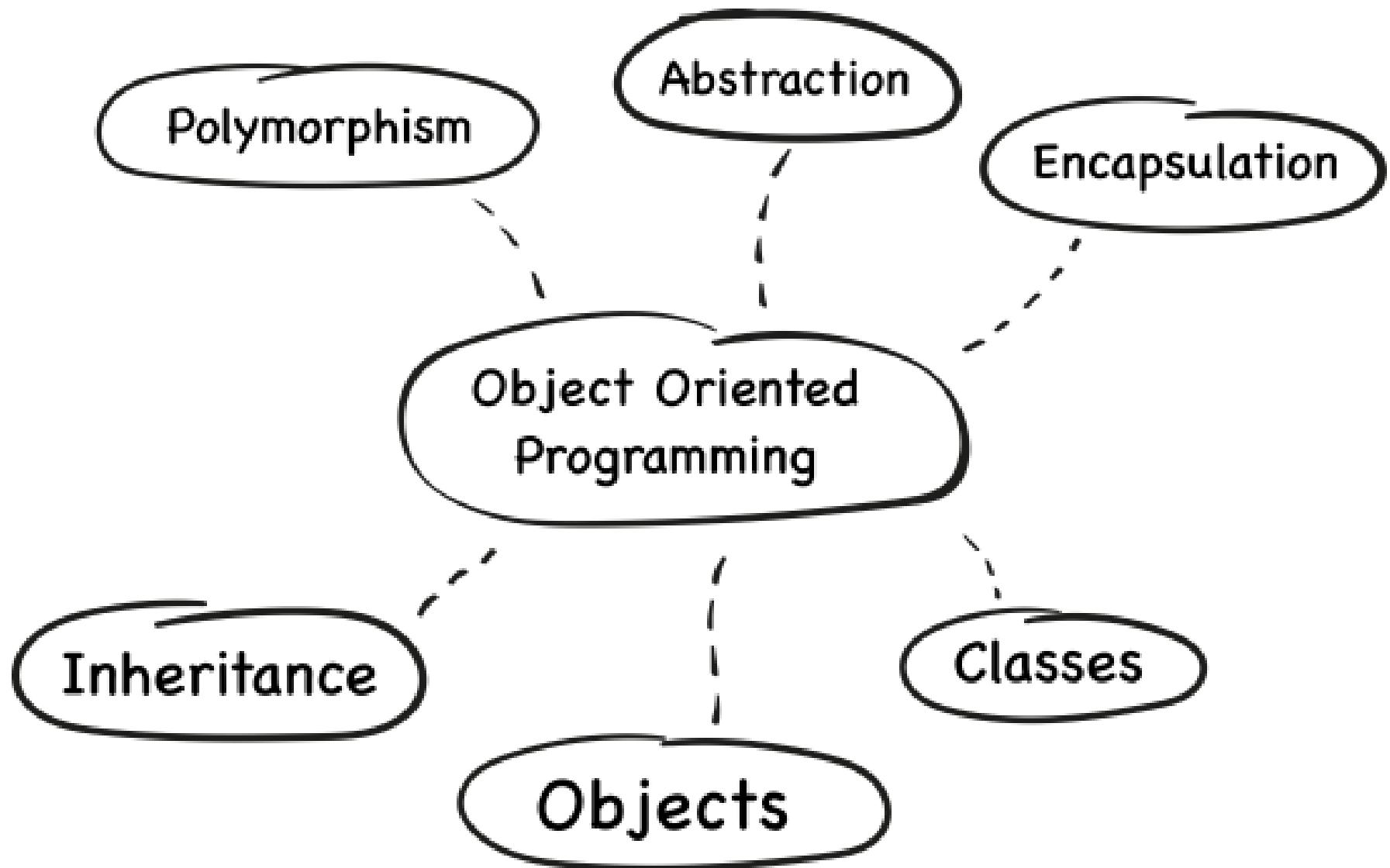
```
<?php
foreach (array_expression as $value){
    // use $value;
}
?>
```

```
<?php
foreach (array_expression as $key => $value){
    // use $key and $value;
}
?>
```

Les fonctions :



```
<?php
function nom_de_la_fonction($arg_1, $arg_2, /* ..., */ $arg_n)
{
    // do this;
    return $retval;
}
?>
```



Dev Web / Backend : PHP : **Object or not Object ?**

Dev Web / Backend : PHP : Object or not Object ?

Evolutions et grands principes :

- Les objets existent en PHP à partir de la **version 4**
- Changements importants en **PHP 5** : → **Java**
 - Introduction **d'interfaces** et de **classes abstraites**
 - Emploi des modifieurs **private** et **public**
 - On retrouve aussi **__toString()**, **__clone()** et un mécanisme de traitement des **exceptions** semblable à celui de Java.
 - Les constructeurs s'appellent désormais : **__construct()** et les destructeurs **__destruct()**
 - Les méthodes portent le mot clef **function** mais ne signalent pas leur type de retour
 - Les **commentaires** de documentation se font à la manière de Java

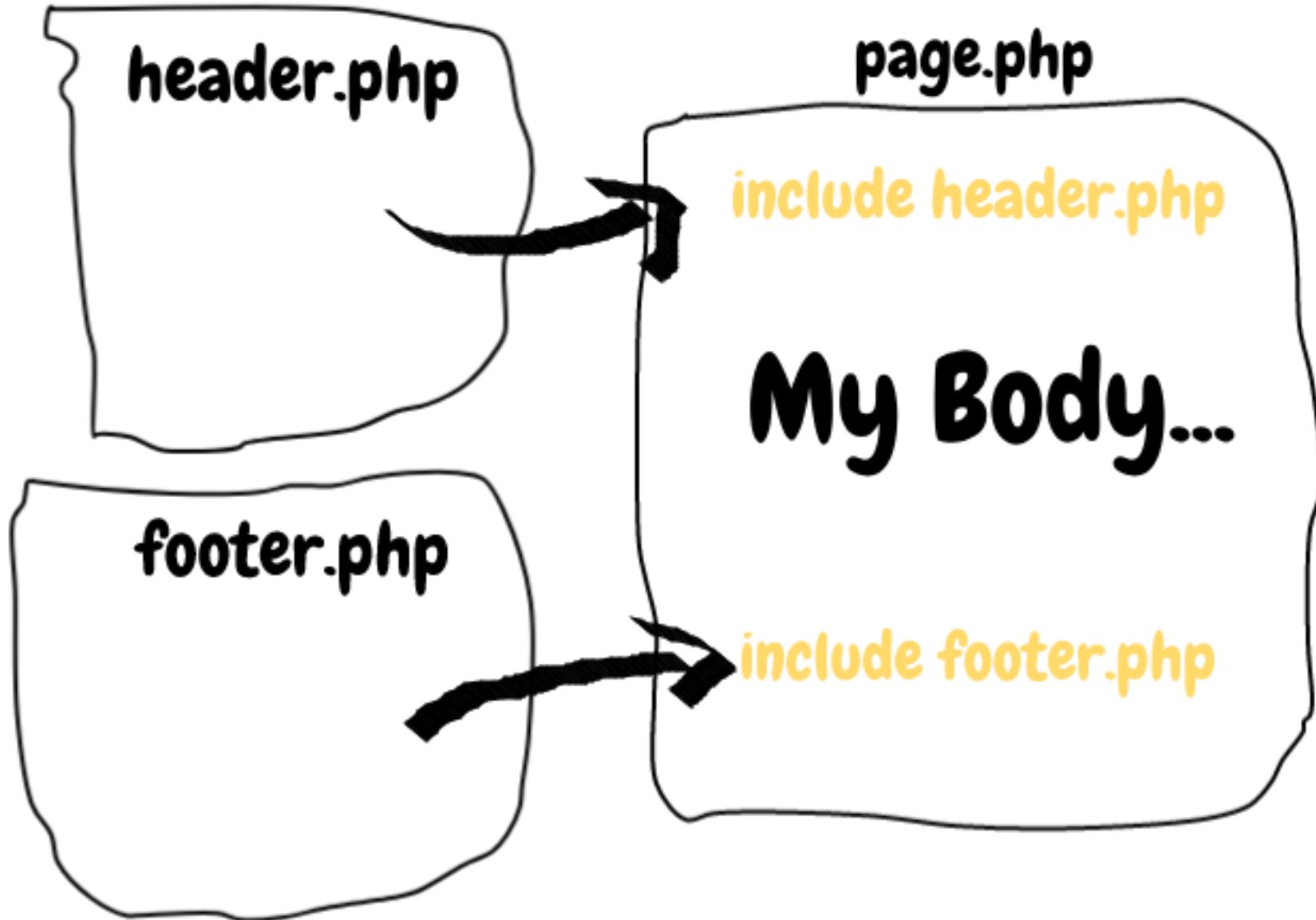
On peut toujours utiliser PHP en mode « **non objet** » mais plusieurs modules – dont le module PDO (BDD) – **manipulent des objets** → Il faut en connaître le fonctionnement ...



Dev Web / Backend : PHP : Object or not Object ?

```
/** Classe Etudiant en PHP */
class Etudiant{
    protected $etudiant_id; /** Identification unique d'un etudiant */
    protected $nom; /** Nom de l'etudiant */
    protected $naissance; /** Date de naissance de l'etudiant */
    public function __construct($id, $nom,$naissance){
        $this->etudiant_id = (int)$id; // cast vers integer
        $this->nom = (string)$nom; // cast vers string
        $this->naissance= (int)$naissance; // cast vers date(timestamp)
    }
    /** * Fonction de comparaison simplifiee entre etudiants */
    public function equals(Etudiant $etudiant){
        return ($this->getId() == $etudiant->getId());
    }
    public function getId(){ return $this->etudiant_id; }
    public function getNom(){ return $this->nom; }
    public function getNaissance(){ return $this->naissance; }
    public function __toString(){
        setlocale(LC_TIME, "fr_FR"); $ne=strftime('%A %d %B %Y',$this->naissance);
        return 'etudiant: id=' . $this->getId().', nom='.$this->getNom()." $ne";
    }
}
/* Test :
date_default_timezone_set('Europe/Paris');
$etu=new Etudiant(234,"Yoda",time());
var_dump($etu); echo "<br/>"; echo $etu;
```

```
object(Etudiant)#1 (3) {
    ["etudiant_id":protected]=> int(234)
    ["nom":protected]=> string(4) "Yoda"
    ["naissance":protected]=> int(1661801345)
}
etudiant: id=234, nom=Yoda Monday 29 August 2022
```



Dev Web / Backend : PHP : Gestion des fichiers externes (Link)

include :

- Semblable aux **include** du **C/C++**
- Réalise une **inclusion physique** du fichier

require et require_once :

- Fonctionnent comme le **include** et le **include_once** respectivement
- Mais le **programme s'arrête** si le fichier inclus n'existe pas

Include_once :

- Identique au include
- **protège** contre d'éventuelles **inclusions multiples** qui pourraient mener à des erreurs (redéclarations, etc.)

```
<?php include_once("connect.php"); ?>
```

```
<?php  
require("malib.php");  
require_once("connect.php");  
?>
```

dirname() :

- Pour savoir dans quel répertoire on se trouve on peut utiliser la fonction PHP **dirname()**

```
<?php  
include_once(dirname(__FILE__) . '/config/config.inc.php');  
?>
```



Dev Web / Backend : PHP : **Les Formulaires**

Dev Web / Backend : PHP : Les Formulaires

Les formulaires permettent d'interagir avec le serveur web en **ENVOYANT** des données

- Le **client** envoie une **requête** demandant la page contenant le formulaire.
- Le **serveur Web construit** la page et le **formulaire**.
- Le **serveur Web renvoie** la page contenant le **formulaire** au **client**.
- Le **navigateur** de l'utilisateur **affiche la page contenant le formulaire**.
L'utilisateur remplit le **formulaire** puis clique sur le bouton de **soumission**.
(Un premier test de validation des données saisies peut être effectué sur le navigateur.)
- Les **données** sont ensuite **envoyées au serveur Web** via le réseau (HTTP méthode **POST** ou **GET**).
- Le **serveur Web réceptionne les données** et les transmet à PHP en les mettant dans la variable **`$_POST`** (si l'attribut '**method**' du formulaire a pour valeur '**post**').
- Le **script destiné à traiter les données est exécuté**. Il va :
 - Vérifier l'authenticité du formulaire.
 - Faire des tests de validité des données.
 - Utiliser les données pour, par exemple les enregistrer dans une base de données.

En cas d'erreur, le formulaire peut être renvoyé à l'utilisateur avec un message détaillant les problèmes rencontrés.

Dev Web / Backend : PHP : Les Formulaires

<FORM> ... </FORM>.

Les formulaires sont délimités par la balise <FORM> ... </FORM>.

Cette balise qui permet de regrouper plusieurs éléments de formulaire (boutons, champs de saisie,...) et qui possède les attributs obligatoires **ACTION** et **METHOD**.

METHOD

METHOD indique sous quelle forme seront envoyées les réponses « **POST** » est la valeur qui correspond à un envoi de données stockées dans le **corps de la requête**, tandis que « **GET** » correspond à un envoi des données **codées dans l'URL**, et séparées de l'adresse du script par un « ? ».

ACTION

ACTION indique l'adresse d'envoi (script CGI, script PHP ou email) :
<mailto:adresse.email@machine>

ENCTYPE

ENCTYPE spécifie le format des données envoyées, dans le cas où un protocole n'imposerait pas de format précis. Il possède plusieurs valeurs dont :

- "text/plain"
- "multipart/form-data"

Il est fortement conseillé de faire des formulaires auto-soumis (la même page gère l'affichage et le traitement du formulaire) :
`action=<?php htmlspecialchars($_SERVER['PHP_SELF']) ?>"`

Dev Web / Backend : PHP : Les Formulaires

Les éléments de Formulaire

On retrouve 3 classes d'éléments :

- **INPUT** : Champs de saisie de texte et différents types de boutons .
- **SELECT** : Listes (menus déroulants et ascenseurs) .
- **TEXTAREA** : Zone de saisie de texte libre.

<INPUT> :

Type	Syntaxe	Exemple
Ø	<input name="ident">	<input type="text"/>
	<input name="ident" value="Par défaut">	Par défaut
submit	<input type="submit" value="Envoi">	envoi
checkbox	<input type="checkbox" name="pfm" value="linux" checked> Linux 	<input checked="" type="checkbox"/> Linux
	<input type="checkbox" name="pfm" value="macos"> Macos 	<input type="checkbox"/> Macos
	<input type="checkbox" name="pfm" value="win"> Windows	<input type="checkbox"/> Windows

Dev Web / Backend : PHP : Les Formulaires

<INPUT> :

Type	Syntaxe	Exemple
radio	<pre><input type="radio" name="media" value="dvd" checked> DVD
<input type="radio" name="media" value="usb"> USB</pre>	<input checked="" type="radio"/> DVD <input type="radio"/> USB
password	<pre><input type="password" value="pass"></pre>	<input type="password"/> <input type="button" value="Efface"/>
reset	<pre><input type="reset" value="Efface"></pre>	<input type="button" value="Efface"/>
file	<pre><input type="file" name="file"></pre>	<input type="text"/> <input type="button" value="Parcourir..."/>

Dev Web / Backend : PHP : Les Formulaires

<SELECT> :

Syntaxe	Exemple
<pre><select name="menu"> <option> Banane <option> Orange <option> Citron <option selected> Pomme <option> Pêche <option> Poire </select></pre>	
<pre><select name="menu" size=4> ... </select></pre>	
<pre><select name="menu" size=4 multiple> ... </select></pre>	

Dev Web / Backend : PHP : Les Formulaires

<TEXTAREA> :

Syntaxe

```
<textarea name="comm" rows=10 cols=40>  
    Tapez vos commentaires ici  
</textarea>
```

Exemple

Tapez vos commentaires ici

Dev Web / Backend : PHP : Les Formulaires

Exemple 1 :

formulaire.php

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8" />
    <title> Formulaire html </title>
  </head>
  <body>
    <form action="reponse.php" method="GET">
      Votre nom :<input type="text" name="nom">
      Votre âge :<input type="text" name="age">
      <p> <input type=submit value="Envoyer">
    </form>
  </body>
</html>
```



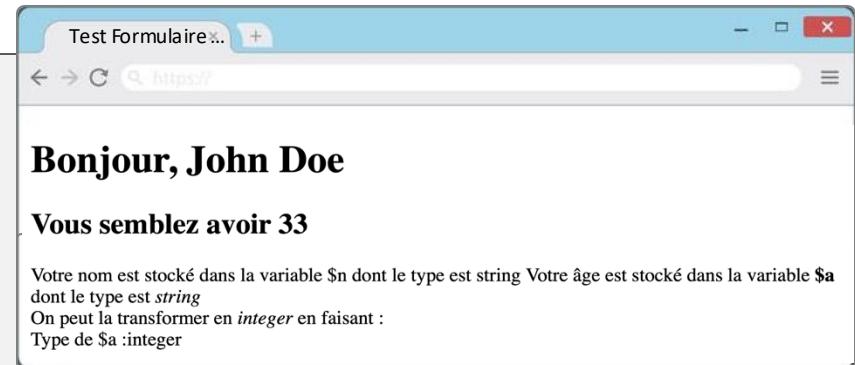
Dev Web / Backend : PHP : Les Formulaires

Exemple 1 :

reponse.php

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8" />
    <title> Test Formulaire PHP </title>
  </head>
  <body>
    <h1>Bonjour, <?php echo $_GET['nom'] ?></h1>
    <h2>Vous semblez avoir <?php echo $_GET['age'] ?></h2>
    <?php
      $n = $_GET['nom'];
      $a = $_GET['age'];
    ?>
    Votre nom est stocké dans la variable $n dont le type est
    <?php echo gettype($n) ?>

    Votre âge est stocké dans la variable <b>$a</b>
    <br/> dont le type est <i><?php echo gettype($a); ?></i>
    <br/> On peut la transformer en <i>integer</i> en faisant :
    <?php settype($a, "integer"); ?>
    <br/> Type de $a :<?php echo gettype($a); ?>
  </body> </html>
```



Dev Web / Backend : PHP : Les Formulaires

Exemple 2 :

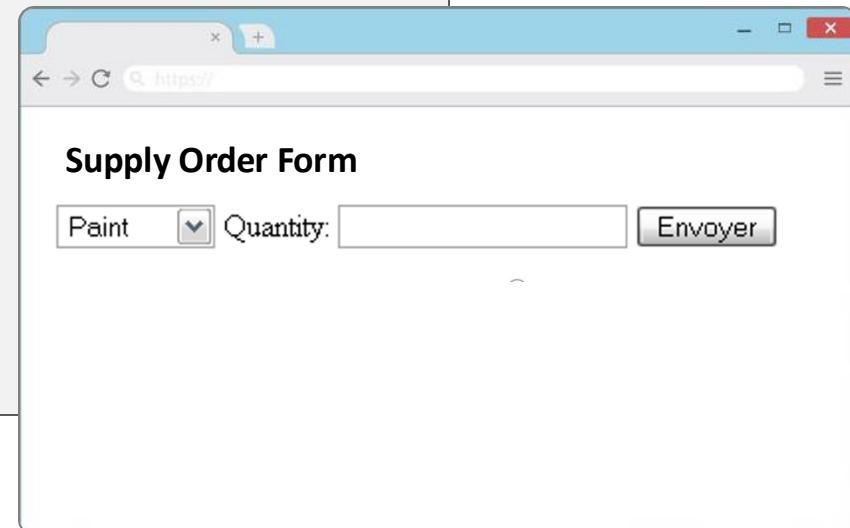
order_form.php

```
<!doctype html>
<html><body>
  <h4>Supply Order Form</h4>
  <form action="<?php htmlspecialchars($_SERVER['PHP_SELF']) ?>" method="post">
    <select name="item">
      <option>Paint</option>
      <option>Brushes</option>
      <option>Erasers</option>
    </select>

    Quantity:
    <input name="quantity" type="text" />

    <input type="submit" />
  </form>

</body></html>
```



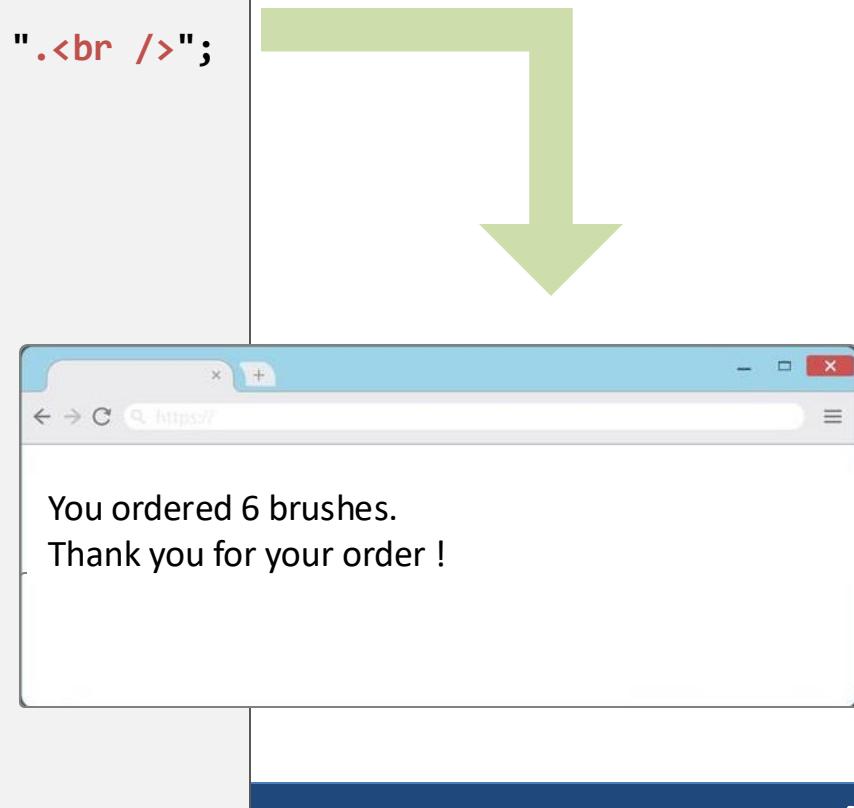
Dev Web / Backend : PHP : Les Formulaires

Exemple 2 :

order_form.php

```
<!doctype html>
<html><body>
<?php
if (isset($_POST["quantity"])){
    $quantity = $_POST['quantity'];
    $item = $_POST['item'];

    echo "You ordered ". $quantity . " " . $item . ".<br />";
    echo "Thank you for your order!";
}
else{
    ?>
    <h4>Supply Order Form</h4>
    <form action="<?php htmlspecialchars($_SERVER['PHP_SELF']) ?>" method="post">
        <select name="item">
            <option>Paint</option>
            <option>Brushes</option>
            <option>Erasers</option>
        </select>
        Quantity:
        <input name="quantity" type="text" />
        <input type="submit" />
    </form>
}
?>
</body></html>
```





Dev Web / Backend : PHP : **Sessions & Cookies**

Dev Web / Backend : PHP : Sessions & Cookies



Les **Sessions** et les **Cookies** permettent de **conserver un état**, de **préserver** des données, relatives au visiteur, **entre plusieurs accès**.

Elles permettent de stocker des types de **données simples** (texte, nombres, ...) mais pas de ressources comme des images ou bdd.

Sessions :

- Stockées côté serveur
- On peut stocker tout objet
- Session close à la fermeture du navigateur

VS

Cookies :

- Stockées côté client
- On peut stocker des strings
- Peuvent être conserver à longue durée

Dev Web / Backend : PHP : Un Langage de Prog.

Les Sessions par l'exemple :

page1.php

```
<?php // page1.php  
  
session_start();  
  
echo 'Welcome to page #1';  
  
$_SESSION['favcolor'] = 'green';  
$_SESSION['animal'] = 'cat';  
$_SESSION['time'] = time();  
  
// Works if session cookie was accepted  
echo '<br /><a href="page2.php">page 2</a>';  
  
// Or maybe pass along the session id, if needed  
echo '<br /><a href="page2.php?' . SID . '">page 2</a>';  
?>
```

Démarrage de la session :

Si un **fichier existe sur le serveur** pour cette session, les variables de sessions seront **récupérées**, si ce n'est pas le cas, **un nouveau fichier sera créé**.

Attention : Cette instruction doit-être placé avant tout code HTML

Création/modification d'une variable de session :

On peut y mettre tout type de variable (même des objets)

Passage du SID :

Si les cookies de session (fichier local) ne sont pas accepté; il faut **propager le SID** de pages en pages

Dev Web / Backend : PHP : Un Langage de Prog.

Les Sessions par l'exemple :

page2.php

```
<?php // page2.php
```

```
session_start();
```

Démarrage de la session :

Utilise le SID (du cookie de session ou passé en paramètre d'url)

```
echo 'Welcome to page #2<br />';
```

```
echo $_SESSION['favcolor']; // green
```

```
echo $_SESSION['animal']; // cat
```

```
echo date('Y m d H:i:s', $_SESSION['time']);
```

Utilisation des variables de session :

On peut y mettre tout type de variable
(même des objets)

```
// You may want to use SID here, like we did in page1.php
```

```
echo '<br /><a href="page1.php">page 1</a>';
```

```
unset($_SESSION['favcolor']);
```

Supprimer une ou toutes les variables de session :

```
session_unset();
```

```
?>
```

Dev Web / Backend : PHP : Un Langage de Prog.

Les Cookies par l'exemple :

page1.php

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

Gestion des variables Cookie

Les cookies doivent être activés sur le navigateur

Attention : Cette instruction doit-être placé avant tout code HTML

page2.php

```
<?php  
// Afficher un cookie  
echo $_COOKIE["TestCookie"];  
  
// Une autre méthode pour afficher tous les cookies  
print_r($_COOKIE);  
?>
```

Affichage des variables Cookie

Dev Web / Backend : PHP : Un Langage de Prog.

WARNING !

```
<?php // page1.php  
  
require('myfunctions.php');  
  
session_start();  
echo 'Welcome to page #1';  
  
$_SESSION['favcolor'] = 'green';  
$_SESSION['animal'] = 'cat';  
$_SESSION['time'] = time();  
  
// Works if session cookie was accepted  
echo '<br/><a href="page2.php">page 2</a>';  
  
?>
```

```
<?php  
  
/**  
 * Useless fiction  
 */  
function function1(){  
    return "test";  
}  
  
?>
```

Welcome to page #1
page 2



Dev Web / Backend : PHP : Un Langage de Prog.

WARNING !

```
<?php // page1.php  
  
require('myfunctions.php');  
  
session_start();  
echo 'Welcome to page #1';  
  
$_SESSION['favcolor'] = 'green';  
$_SESSION['animal'] = 'cat';  
$_SESSION['time'] = time();  
  
// Works if session cookie was accepted  
echo '<br/><a href="page2.php">page 2</a>'>  
?>
```

```
<?php  
  
/**  
 * Useless fiction  
 */  
function function1(){  
    return "test";  
}  
  
?>
```



/bugsession/test1.php on line 5



Warning: session_start(): Session cannot be started
Welcome to page #1
[page 2](#)

Dev Web / Backend : PHP : Un Langage de Prog.

WARNING !

```
<?php // page1.php  
  
require('myfunctions.php');  
  
session_start();  
echo 'Welcome to page #1';  
  
$_SESSION['favcolor'] = 'green';  
$_SESSION['animal'] = 'cat';  
$_SESSION['time'] = time();  
  
// Works if session cookie was accepted  
echo '<br/><a href="page2.php">page 2</a>';  
  
?>
```

```
<?php  
  
/**  
 * Useless fiction  
 */  
function function1(){  
    return "test";  
}  
  
?>
```



Pas Bien
makeabit.com
b92 b160

Warning: session_start(): Session cannot be started after headers have already been sent in /var/www/html/bugsession/test1.php on line 5
Welcome to page #1
[page 2](#)

PHP



PDO



Dev Web / Backend : PHP : PDO



PDO_MYSQL est un pilote qui implémente l'interface de **PHP Data Objects (PDO)** pour autoriser l'accès de PHP aux bases de données **MySQL**.

Connexion aux bases de données depuis PHP avec PDO par l'exemple

Source : <https://www.univ-orleans.fr/iut-orleans/informatique/intra/tuto/php/bd-php-pdo.html>

```
CREATE TABLE `CARNET` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `NOM` varchar(30) DEFAULT NULL,
  `PRENOM` varchar(30) DEFAULT NULL,
  `NAISSANCE` date DEFAULT NULL,
  `VILLE` varchar(30) DEFAULT NULL,
  PRIMARY KEY (`ID`) ) ENGINE=InnoDB AUTO_INCREMENT=13 DEFAULT CHARSET=utf8;
INSERT INTO `CARNET` VALUES
(1, 'SMITH', 'JOHN', '1980-12-17', 'ORLEANS'),
(2, 'DURAND', 'JEAN', '1983-01-13', 'ORLEANS'),
(3, 'GUDULE', 'JEANNE', '1967-11-06', 'TOURS'),
(4, 'ZAPATA', 'EMILIO', '1956-12-01', 'ORLEANS'),
(5, 'JOURDAIN', 'NICOLAS', '2000-09-10', 'TOURS'),
(6, 'DUPUY', 'MARIE', '1986-01-11', 'BLOIS'),
(7, 'ANDREAS', 'LOU', '1861-02-12', 'ST Petersbourg'),
(9, 'Kafka', 'Franz', '1883-07-03', 'Prague'),
(11, 'Dalton', 'Joe', '2003-12-06', 'Dallas');
```

On insère cette table dans MySQL en ligne de commande ou à l'aide de PHPMyAdmin.

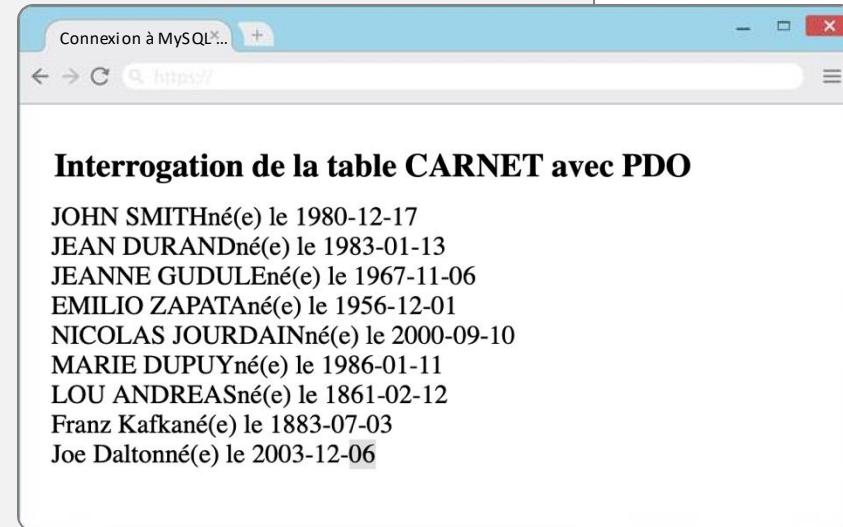
Puis, pour consulter cette table depuis PHP, on utilise le connecteur PDO qui offre une interface de connexion utilisable pour tous les SGBD

Dev Web / Backend : PHP : PDO

Connexion Simple en PHP avec PDO

```
<!doctype html>
<html> <head> <title> Connexion à MySQL avec PDO </title> <meta charset="utf-8"> </head>
<body>
<h1> Interrogation de la table CARNET avec PDO </h1>
<?php
require("connect.php");
$dsn="mysql:dbname=".BASE.";host=".SERVER;
try{
    $connexion=new PDO($dsn,USER,PASSWD);
}
catch(PDOException $e){
    printf("Échec de la connexion : %s\n", $e->getMessage()); exit();
}
$sql="SELECT * from CARNET";
if(!$connexion->query($sql))
    echo "Pb d'accès au CARNET";
else{
    foreach ($connexion->query($sql) as $row)
        echo $row['PRENOM']." ".
              $row['NOM']."né(e) le ".
              $row['NAISSANCE']."<br/>\n";
}
?>
</body> </html>
```

```
<?php
define('USER','scott');
define('PASSWD','tiger');
define('SERVER','localhost');
define('BASE','dbscott');
?>
```

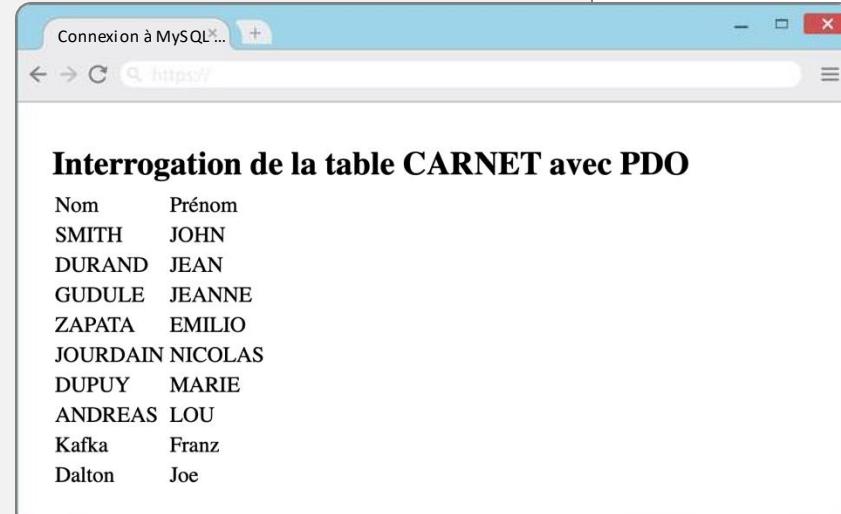


Dev Web / Backend : PHP : PDO

Fabrication d'une table HTML avec les résultats

```
<!doctype html>
<html> <head> <title> Connexion à MySQL avec PDO </title> <meta charset="utf-8"> </head>
<body>
<h1> Interrogation de la table CARNET avec PDO </h1>
<?php
require("connect.php"); ←
$dsn="mysql:dbname=".BASE.";host=".SERVER;
try{
    $connexion=new PDO($dsn,USER,PASSWD);
}
catch(PDOException $e){
    printf("Échec de la connexion : %s\n", $e->getMessage()); exit();
}
$sql="SELECT * from CARNET";
if(!$connexion->query($sql))
    echo "Pb d'accès au CARNET";
else{
    ?> <table> <tr> <td> Nom </td> <td> Prénom </td></tr>
    <?php
        foreach ($connexion->query($sql) as $row)
            echo "<tr><td>".$row['NOM']."</td><td>".$
                $row['PRENOM']."</td></tr>\n";
    ?> </table>
<?php
}
?>
</body> </html>
```

```
<?php
define('USER', "scott");
define('PASSWD', "tiger");
define('SERVER', "localhost");
define('BASE', "dbscott");
?>
```



The screenshot shows a browser window titled "Connexion à MySQL...". The main content area displays a table with the heading "Interrogation de la table CARNET avec PDO". The table lists ten entries, each consisting of a last name ("Nom") and a first name ("Prénom").

Nom	Prénom
SMITH	JOHN
DURAND	JEAN
GUDULE	JEANNE
ZAPATA	EMILIO
JOURDAIN	NICOLAS
DUPUY	MARIE
ANDREAS	LOU
Kafka	Franz
Dalton	Joe

Dev Web / Backend : PHP : PDO

Fabrication d'une table HTML avec les résultats : ++

Refactoring : Fonction de connexion : **connexion.php**

```
<?php
require("connect.php");

function connect_bd(){
    $dsn="mysql:dbname=".BASE.";host=".SERVER;
    try{
        $connexion=new PDO($dsn,USER,PASSWD);
    }
    catch(PDOException $e){
        printf("Échec de la connexion: %s\n", $e->getMessage());
        exit();
    }
    return $connexion;
}
?>
```



Ajout d'une CSS : **tabstyle.css**

```
/* Bordure simple autour
des tableaux */
table, th, td {border: 1px solid grey;}
table{border-collapse:collapse;}
/* Centrage tableau */
table.centre{ margin:auto;}
/* centrage du texte dans
les cellules du tableau */
table.centre td{text-align:center;}

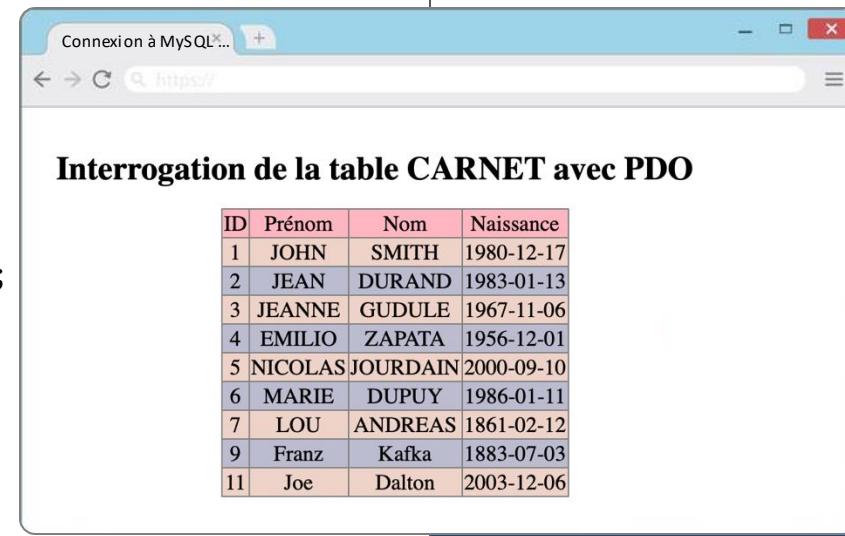
table#jolie tr:first-child{
    background:LightPink;
}
table#jolie tr:nth-child(2n){
    background:#EFD3C9;
}
table#jolie tr:nth-child(2n+3){
    background:#BCBCD0;
}

/* si un tableau a une seule
ligne on l'affiche en rouge */
table tr:only-child{
    background:red;
}
```

Dev Web / Backend : PHP : PDO

Fabrication d'une table HTML avec les résultats : ++

```
<html> <head>
<title> Connexion à MySQL avec PDO </title> <meta charset="utf-8">
<link rel="stylesheet" href="tabstyle.css" />
</head> <body>
<h1> Interrogation de la table CARNET avec PDO </h1>
<?php
require_once('connexion.php');
$connexion=connect_bd();
$sql="SELECT * from CARNET";
if(!$connexion->query($sql))
    echo "Pb d'accès au CARNET";
else{
    ?>
    <table class="centre" id="jolie">
        <tr> <td> ID </td> <td> Prénom </td> <td> Nom </td>
            <td> Naissance </td> </tr>
    <?php
    foreach ($connexion->query($sql) as $row)
        echo "<tr><td>".$row['ID']."</td> <td>".
            $row['PRENOM']."</td> <td>".
            $row['NOM']."</td> <td>".
            $row['NAISSANCE']."</td></tr><br/>\n";
    ?>
    </table>
<?php
}
?>
</body> </html>
```



The screenshot shows a web browser window titled "Connexion à MySQL". The page content is titled "Interrogation de la table CARNET avec PDO". Below the title is a table with the following data:

ID	Prénom	Nom	Naissance
1	JOHN	SMITH	1980-12-17
2	JEAN	DURAND	1983-01-13
3	JEANNE	GUDULE	1967-11-06
4	EMILIO	ZAPATA	1956-12-01
5	NICOLAS	JOURDAIN	2000-09-10
6	MARIE	DUPUY	1986-01-11
7	LOU	ANDREAS	1861-02-12
9	Franz	Kafka	1883-07-03
11	Joe	Dalton	2003-12-06

Dev Web / Backend : PHP : PDO

Création d'une liste déroulante :

```
<!doctype html>
<html> <head> <title> Connexion à MySQL avec PDO </title> <meta charset="utf-8"> </head>
<body>
<h1> Interrogation de la table CARNET avec PDO </h1>
<?php require_once('connexion.php');
$connexion=connect_bd();
$sql="SELECT * from CARNET";
if(!$connexion->query($sql))
    echo "Pb d'accès au CARNET";
else {
    ?>
    <form action="recherche.php" method="GET">
        <select name="ID">
            <?php
                foreach ($connexion->query($sql) as $row)
                    if(!empty($row['NOM']))
                        echo "<option value='".$row['ID']."'>" . $row['PRENOM'] . " " . $row['NOM']. "</option>\n";
            ?>
        </select>
        <input type="submit" value="Rechercher">
    </form>
<?php
}
?>
</body> </html>
```

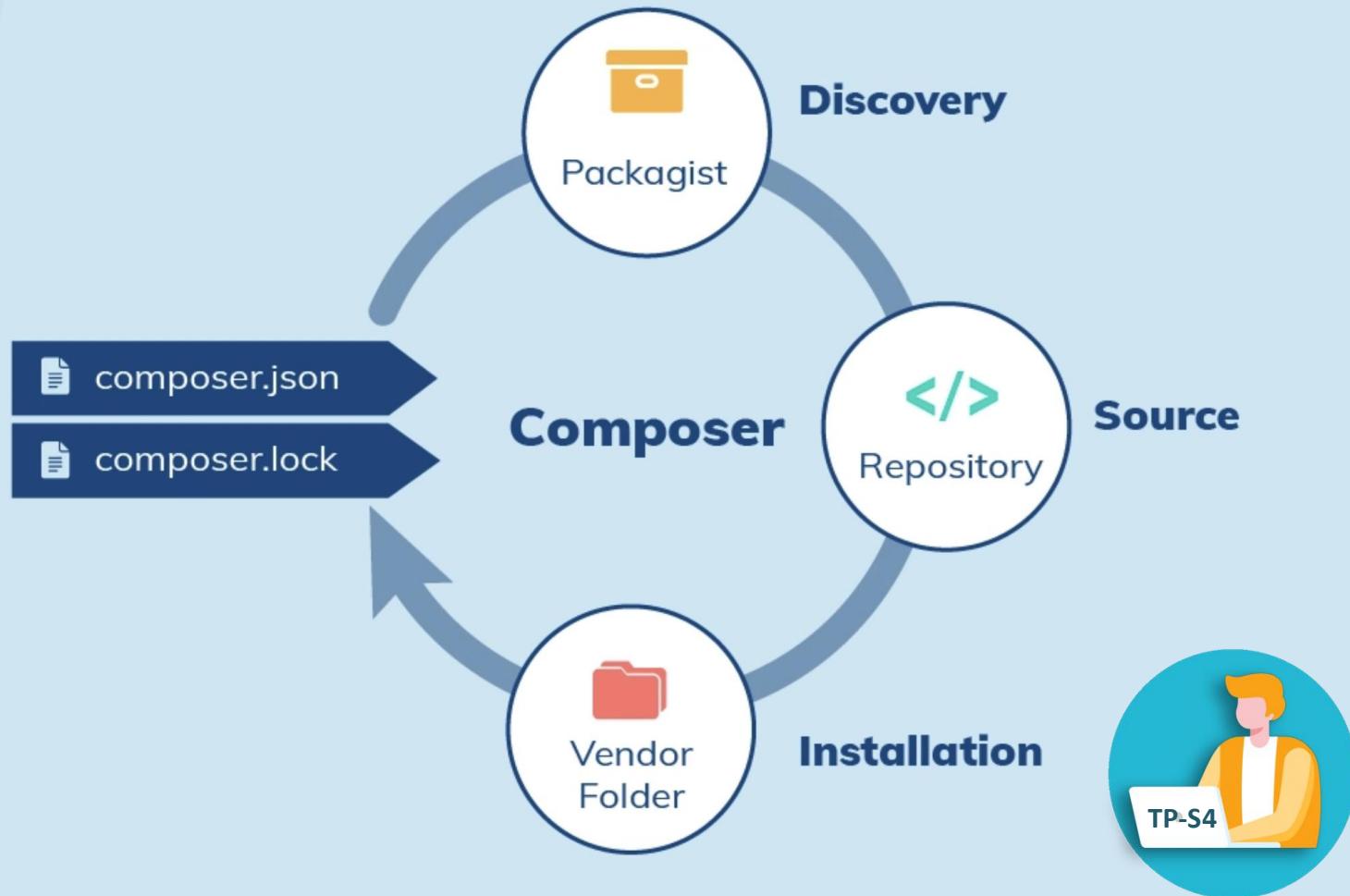




Dev Web / Backend : PHP : **Gestion des paquets**

Dev Web / Backend : PHP : Gestion des paquets

Comment fonctionne Composer ?





Th-Th-Th-That's All Folks

Dev Web / Backend : PHP : **FIN**