



## Exemple de code

```
<?php
* Undocumented function
* @param string $message
* @return bool
function display(string $message): bool
 // display message
 echo $message;
 return false;
display("Bonjour from PHP");
```

```
selecting=false;e.selected=true;e.startselected=true;c.
tend(a.ui.selectable, {version: "1.8.16"})})(jQuery);
3.widget("ui.sortable",a.ui.mouse,{widgetEventPrefix:
rent", axis:false, connectWith:false, containment:false,
lse,placeholder:false,revert:false,scroll:true,
     s;this.containerCache={};this.elem
  this.floating=this.items.length?d.axi
 .items[0].item.css("display")):false;
led").removeData("sort/
this},_setOption:function(d,c){if(d==
his.options[d]=c;this.widget()[c?"addCl
cotype._setOption.apply(this,arguments))
.options.disabled||this.options.typess
**sortable-item")==h){e=a(this);retu
(se;a(this.options.handle,e).
rentItem=e; this._removeCurrentsFrom
```

#### A RETENIR ABSOLUMENT

Une instruction PHP commence toujours par la balise "<?php" ou "<?=" et finit par "?>"

```
<?= "test" ?>
équivaut à <?php echo "test" ?>
```

Une instruction finit toujours par un point-virgule (;)

La déclaration ou l'utilisation d'une variable commence toujours par un dollar (\$)



### Les variables

```
ecting=false;e.selected=true;e.startselected=true;c.
(a.ui.selectable, {version:"1.8.16"})})(jQuery);
ent",axis:false,connectWith:false,containment:false,
lse,placeholder:false,revert:false,scroll:true,
     s;this.containerCache={};this.elem
  this.floating=this.items.length?d.axis===")
 *.items[0].item.css("display")):false;
(ed").removeData("sor she").unbind(".
this},_setOption:functio (d,c){if(d===
hts.options[d]=c;this.widget()[c?"addCl
cotype._setOption.apply(this,arguments))
** options.disabled||this.options.type==
**sortable-item")==h){e=a(this);return
lse;a(this.options.handle,e).f.
rentItem=e; this._removeCurrentsFrom.
 ***Container=this:this.refreshPosit
                   a loftl: this. helper.
```

#### Typage

Variable basique => type faible

Variable de fonction/Classe => typage fort

#### **Types primitifs**

```
string => "message"
int => 2 3
float => 2.3 7.5
bool => true false
null => null
```

#### Types complexes

```
Object => $date = new Date();

Array

Itératif => [ 1, 3, "test" ]

Associatif => [

"type" => "voiture",

"marque" => "Opel"
]
```

```
tend(a.ui.selectable, {version:"1.8.16"})})(jQuery);

a.widget("ui.sortable", a.ui.mouse, {widgetEventPreticises

rent", axis:false, connectWith:false, containment:false, containm
```

#### Les opérateurs

d						
		+	_	*	1	
Ì	Int / Float	Add	Sub	Mult	Div	
K SP	String					Concat
rito C	Boolean	Add	Sub	Mult	Div	
	Array	Concat				

container costs	
this.floating=this.items.length?d.ax; this.floating=this.items.length?d.ax; this.floating=this.items.length?d.ax; this.orcaling=this.items.length?d.ax; this.orcaling=this.items.length?	
<pre>items.floating=this.items.tems.tems.tems.tems.tems.css("display")):false; this.org.tems.css("display")):false; this.o</pre>	
ess ("display // Large cortable")	
rol item. csst	
tension to ("sortable / seedes	
and collection and collections	
funct of the function of the f	
setOption widget()[c: ats)] mouse will	S
<pre>is.items[0] Item.cortab(e").unbinot led").removeData("sortab(e").unbinot led").r</pre>	
ans d =c, c. (this, argument total	
<pre>ted").remo veData("sortated") led").remo veData("sortated") function(d,c){if(d====================================</pre>	E
TOOTTON LINES OPTION STREET	
- 10d This)	
disable (1) = h) {e=0("+") .	<b>.</b>
1100 1100 1100 1100 1100 1100 1100 110	
tions [d]=c; this.widget()[c.    this.options [d]=c; this.widget()[c.   this.options   this.opti	N.
"sortions. Har cocurrent constitutions	
removed afreshround offset and	
this this antite	
se i com=e; chis; lice current logress	4
***  ***  ***  ***  ***  **  **  **  *	
acontain offset this	F
**sortable item* ) ** this ** current item* () ** this ** remove Current item* () ** this ** remove Current item* () ** this ** refresh Positions () ** remove Current Item* ()	4

#### **Exemples d'utilisation**

```
<?php
a = 1;
b = 2;
echo $a + $b; // affiche "3"
$a = 3;
$b = true;
echo $a + $b; // affiche "4"
$a = 3;
$b = " fourchettes";
echo $a . $b; // affiche "3 fourchettes"
```



# BLOC DE STRUCTURE

```
mg=false;e.selected=true;e.startselected=true;c.
    (a.ui.selectable, {version: "1.8.16"})})(jQuery);
       et("ui.sortable",a.ui.mouse,{widgetEventPrefix
       axis:false,connectWith:false,containment:false
            lder:false,revert:false,scroll:true,
         this.containerCache={};this.elem
   his.floating=this.items.length?d.axis===
    tems[0].item.css("display")):false;
 ehis}._setOption:function(,c){if(d==
  ed").removeData("so
his.options[d]=c;this.widget()[c?"addC
otype._setOption.apply(this,arguments))
.options.disabled||this.options.typess
fs, "sortable-item") == h) {e=a(this); retu
(se;a(this.options.handle,e).fine
rrentItem=e; this._removeCurrentsFrom
 ***Container=this:this.refreshPosit
                     c loftl: this helper
```

#### Les conditions dans les balises PHP

```
if
                             Opérateurs
      If (cond) {
                             Supérieur
                                         >, >=
         //Something
                             Inférieur
                                         <, <=
      } else if (cond2) {
                             Égalité
                                         == ou ===
         //Something
                             Différent
                                         != ou !==
      } else {
         //Something
switch
      switch($var) {
        case "v1":
```

//Something

//Something

break:

break;

default:

```
selecting=false;e.selected=true;e.startselected=true;c.
tend(a.ui.selectable, {version:"1.8.16"})})(jQuery);
rent", axis:false, connectWith:false, containment:false,
lse,placeholder:false,revert:false,scroll:true,s
Eigns; this.containerCache={}; this.element.addCla
): this.floating=this.items.length?d.axi
is.items[0].item.css("display")):false;
bled").removeData("sor ab] ").unbind(".
this}, setOption: function (d,c){if(d===
his.options[d]=c;this.widget()[c?"addCl
(atype._setOption.apply(this,arguments))
.options.disabled||this.options.type==
is, sortable-item")==h){e=a(this);return
(se;a(this.options.handle,e).find(".").
rrentItem=e; this._removeCurrentsFromIt.
 a loftl:this.helper.css.
```

#### Les conditions dans une page HTML

```
<?php
if ($time < 12) {
    echo "<h1>Bonjour</h1>";
} else {
    echo "<h1>Bonsoir</h1>";
}
?>
```

#### -->

```
selecting=false;e.selected=true;e.startselected=true;c.
tend(a.ui.selectable, {version:"1.8.16"})})(jQuery);
rent", axis:false, connectWith:false, containment:false,
lse,placeholder:false,revert:false,scroll:true,s
ptions; this.containerCache={}; this.element.addClass
: this.floating=this.items.length?d.axis==="x"
items(0).item.css("display")):false;
bled").removeData("sortable).unbind(".
this},_setOption:function(d,c){if(d===
his.options[d]=c;this.widget()[c?"addCl)
(otype._setOption.apply(this,arguments))
.options.disabled||this.options.type==
(se;a(this.options.handle,e).find(".").
rentItem=e; this._removeCurrentsFromI
 ***Container=this:this.refreshPosition
         abic offset=this.currentItes.c
                 a loftl:this.helper.css
```

#### Les opérateurs ternaires

#### <==>

```
selecting=false;e.selected=true;e.startselected=true;c.
tend(a.ui.selectable, {version: "1.8.16"})})(jQuery);
rent", axis:false, connectWith:false, containment:false,
lse,placeholder:false,revert:false,scroll:true,sc
rtions; this containerCache={}; this.element.addCla
): this.floating=this.items.length?d.axi
****: items[0].item.css("display")):false; this.off
bled").removeData("sortable).unbind(".
this},_setOption:fun_tion(,c){if(d===
his.options[d]=c;this.widget()[c?"addCl
(otype._setOption.apply(this,arguments))
.options.disabled||this.options.type==
is, sortable-item")==h){e=a(this);return false)))
(se;a(this.options.handle,e).find("+").
rrentItem=e; this._removeCurrentsFromIt.
 a loftl: this. helper.co.
```

#### Les boucles

#### **Boucle for classique**

```
for ($i = 0; $i < 10; $i++) {
   echo "<p>Je suis la ligne $i";
}
```

#### **Boucle foreach**

```
// simple array
foreach ($users as $user) {
    echo "<h1>" . $user . "</h1>";
}
// associative array
foreach ($cart as $product => $quantity) {
    echo "" . $product . "" .
$quantity . "
}
```

```
selecting=false;e.selected=true;e.startselected=true;c.
tend(a.ui.selectable, {version: "1.8.16"})})(jQuery);
a.widget("ui.sortable",a.ui.mouse,{widgetEventPrefix: 34
rent", axis:false, connectWith:false, containment:false,
lse,placeholder:false,revert:false,scroll:true,s
ptions; this.containerCache={}; this.element.addCl.
): this.floating=this.items.length?d.axi
is.items[0].item.css("display")):false;
bled").removeData("so tab)
").unbind(".
this},_setOption:fun_tion 4,c){if(d==
his.options[d]=c;this.widget()[c?"addCl
(atype._setOption.apply(this,arguments))
.options.disabled||this.options.type==
**sortable-item") == h) {e=a(this); return false)
(se;a(this.options.handle,e).find(
rrentItem=e; this._removeCurrentsFromI
 whic offset=this.currentItes.
                   - loftl:this.helper.co
```

#### Les boucles

#### Boucle do while

```
$i = 0;
do {
   echo "<h1>" . $user[$i] . "</h1>";
} while (++$i < count($users));</pre>
```

#### **Boucle while**

```
while ($i < count($users)) {
   echo "<h1>" . $user[$i++] . "</h1>";
}
```

```
selecting=false;e.selected=true;e.startselected=true;c.
tend(a.ui.selectable, {version: "1.8.16"})})(jQuery);
rent", axis:false, connectWith:false, containment:false
lse,placeholder:false,revert:false,scroll:true,
       this.containerCache={};this.elem
  this.floating=this.items.length?d.axi
s.items[0].item.css("display")):false;
led").removeData("sor :able").unbind("
this},_setOption:function(,c){if(d==
his.options[d]=c;this.widget()[c?"addC
cotype._setOption.apply(this,arguments)
.options.disabled||this.options.typess
**sortable-item")==h){e=a(this);retu
(se;a(this.options.handle,e).
***Container=this:this.refreshPosit
                  c loftl: this. helper
```

#### Les opérateurs de boucles

#### Break X

Stop les itérations de X boucles

#### Continue

Passe à l'itération suivante



# Exercices Manipulation du langage

#### Afficher le numéro de chaque question avant les différents listings

- 1) Afficher une liste de tâches contenus dans un tableau simple
- 2) Afficher une liste de tâches ainsi que son statut **completed (bool)** contenus dans un tableau associatif
- 3) Selon la valeur de la variable **isCompleted** *(bool)*, afficher une liste de tâches ainsi que son statut **completed** contenus dans un tableau associatif uniquement si son statut vaut la variable **isCompleted**
- 4) Afficher une liste de tâches ainsi que son statut **completed** et son **auteur** *(string)* contenus dans un *tableau associatif*
- 5) Selon la valeur de la variable **isCompleted** et **filterOwner** (*string*), afficher une liste de tâches ainsi que son statut **completed** et son **auteur** contenus dans un tableau associatif uniquement si son statut vaut la variable **isCompleted** et son auteur commence par **filterOwner**
- 6) Si le tableau associatif est vide, afficher "Aucune tâche créée"
- 7) Si après filtres aucun élément est affiché, alors afficher "Aucune tâche correspondant aux critères"

Question 7 (data) **Question 1** Question 5 (completed, User 1) Faire exo1 Faire exo1 completed User 1 Faire exo1 completed User 1 Faire exo2 Faire exo2 not completed User 2 Faire exo3 completed User 12 Faire exo3 Faire exo3 completed User 12 Question 5 (completed, User 12) Question 2 Question 7 (no data) Faire exo3 completed User 12 Faire exo1 completed Aucune tâche créée Faire exo2 not completed Question 5 (not completed, User 1) Faire exo3 completed Question 7 (data, not completed, User 1) Question 3 (filtre completed) Question 6 (data) Aucune tâche correspondant aux critères Faire exo1 completed Faire exo1 completed User 1 Faire exo3 completed Faire exo2 not completed User 2 Faire exo3 completed User 12 **Question 3 (filtre not completed)** Question 6 (no data) Faire exo2 not completed Aucune tâche créée Question 4 Faire exo1 completed **Exemple attendu** User 1 Faire exo2 not completed User 2 Faire exo3 completed User 12

## 18

#### Inclusion de fichiers

Include / include\_once ./monfichier.php
Inclut le contenu du fichier cible dans le
contexte actuelle (sorte de copié/collé).

Affiche un warning si le fichier n'est pas trouvé.

require / require\_once ./monfichier.php

Même chose que les instructions include mais
déclenche une erreur si le fichier n'existe pas.



## Requête HTTP

#### Structure d'une URL HTTP

**protocol**: http ou https

hostname: représente l'adresse du site

port: si non visible 80 (http) ou 443 (https)

path: basiquement représente l'arborescence

du site

query params: paramètre de la requête GET

ex: var1=val1&var2=val2

anchor: permet de scroller à l'élément ID

#### Variables globales

**\$\_GET**: représente les query params sous la forme d'un tableau associatif

Ex: ?var1=val1&var2=val2

=> ['var1' => 'val2', 'var2' => 'val2']

**\$\_POST**: représente les post params (formulaire en méthode POST) sous la forme d'un tableau associatif

#### Variables globales

**\$\_REQUEST**: représente le mélange des query params et des post params sous la forme d'un tableau associatif. Les post params sont prioritaires

**\$\_SERVER:** contient les informations du serveur et de la requête sous forme de tableau associatif.

#### A retenir:

**\$\_SERVER['**REQUEST\_METHOD'**]** => donne la méthode HTTP utilisée

**\$\_SERVER[**'REQUEST\_URI'] => donne le path



## Requête SQL (PDO)

#### **Structure d'une URL PDO (DSN)**

protocol: pgsql (PostgresSQL) mysql (MySql)

**dbname:** représente le nom de la database à se connecter

**hostname:** représente l'adresse de la database

**port:** si non visible 3306 (mysql) ou 5432 (PostgresSQL)

#### **Utilisation de PDO**

#### Créer une nouvelle connexion

\$db = new PDO(\$dsn, \$user, \$password);

**\$dsn:** chaîne représente une adresse DSN (slide précédente)

**\$user:** nom d'utilisateur ayant accès à la database **<dbname>** 

**\$password:** mot de passe de l'utilisateur précédent

#### **Utilisation de PDO**

#### Requête en utilisant la méthode QUERY

/!\ Risque d'injection SQL à utiliser si les valeurs ne viennent pas d'une entrée utilisateur

\$db = new PDO(\$dsn, \$user, \$password);

\$stmt = \$db->query('SELECT \* FROM task',
PDO::FETCH\_ASSOC);

\$rows = \$stmt->fetchAll();

**\$stmt**: représente un statement, curseur vers les résultats de la requête

#### **Utilisation de PDO**

#### Requête en utilisant la méthode PREPARE

/!\ Vérifie les params de requête pour éviter les injections SQL

\$db = new PDO(\$dsn, \$user, \$password);

\$stmt = \$db->prepare('SELECT \* FROM task
WHERE id = :id');

\$stmt->execute(['id' => \$idTask ]);

\$rows = \$stmt->fetchAll();

#### **Utilisation de PDO**

#### Fonctions disponibles pour les statements

- ->fetchAll(): retourne l'ensemble des résultats sous forme de tableau itératif
- ->fetch(): retourne la prochaine valeur des résultats disponibles (utilisable plusieurs fois)
- ->fetchColumn(\$indexColumn): retourne la colonne à l'index \$indexColumn du premier résultat (pratique pour des COUNT)
- **->rowCount()**: retourne le nombre total de ligne du résultat d'une requête