

PHP

<https://www.codepile.net/pile/dwm12php>

Nécessité d'avoir un serveur pour afficher PHP (via la virtualisation, terminal pour les cours)

Possibilité de mettre du HTML dans le fichier PHP, le contraire n'est pas possible

Bosser sous LAMP signifie, lancer le vagrant init etc (retenir l'adresse IP)

L/M/W : Linux (l'OS, notre environnement virtuel) M=>Mac, W=>Window

A : Apache (notre serveur, afin d'afficher et d'intégrer notre serveur)

M : MySQL (notre système de base de données)

P : PHP (pour pouvoir utiliser notre langage)

Mettre à jour les paquets à installer : `sudo apt update`

Pour installer un paquet : `sudo apt install XXX`

Paquet à installer :

- apache 2 `sudo apt install apache2` (ceci a créé automatiquement un fichier html dans le dossier data (où se situe le Vagrantfile), pour vérifier qu'apache2 est bien installé, ouvrir l'explorateur et entrer l'adresse IP)
- `sudo apt install mysql-server` (psw : 0000)
- spécifique au locaux IT, `sudo apt-add-repository ppa:ondrej/php` forcer l'installation de PHP.3, et faire `sudo apt update` derrière
- `sudo apt install php7.3 -y` (pour éviter de valider avec Y à chaque fois)
- `sudo apt install php7.3-mysql` (pour le pdo et lier php à mysql)

Paquet pour mieux interpréter le `var_dump`

`sudo apt install php-xdebug -y`

Activer les erreurs en PHP : **!! À TOUJOURS APPLIQUER !!**

- `sudo nano /etc/php/7.3/apache2/php.ini`
- CTRL + W pour chercher "display_errors"
- CTRL + W puis taper la touche "entrée" pour aller au mot suivant
- Modifier les lignes "display_errors = Off" et "display_startup_errors = Off" en On
- Sauvegarder (CTRL + O, puis CTRL + X)
- **Commande magique** `sudo service apache2 restart`, relance le serveur Apache

Extensions à installer pour aider le codage :

- dans l'IDE
- extensions
- php => php intellisense et bracket 2.0

PHP : Création du fichier dans l'IDE avec l'extension .php

<code><?php ?></code>	signifie que je vais faire du php pour fermer et faire derrière du html
<code>\$fruit="XXX"; (fraise)</code>	pour créer une variable
<code>echo \$fruit;</code>	pour afficher le contenu de la variable
EX: <code>echo "j'aime la \$fruit";</code>	=j'aime la fraise
<code>echo gettype(\$XXX); \$myString ="navet"; \$myNumber =30; \$myFloat =3.5; \$myBool = true 1;false 0 \$myArray =[];</code>	pour afficher le type de la variable
<code>var_dump(\$myArray);</code> Résultat : <code>\$myArray =["Tomatoes", "Potatoes", "Salad", "Carrot"];</code>	pour se debugger et savoir où j'en suis dans mon code, équivalent du console.log avec JS
<code>echo \$myArray[3];</code> Résultat : pour afficher "Carrot" array (size=4) 0 => string 'Tomatoes' (length=8) 1 => string 'Potatoes' (length=8) 2 => string 'Salad' (length=5) 3 => string 'Carrot' (length=6)	afficher un élément spécifique
<code>echo \$myArray[3];</code>	Array numérique, car à l'affiche, tout est listé numériquement
<code>\$myArray = ["name" => "John", "age" => "54",</code>	Array associatif, on associe une clé à une valeur en utilisant => name, age, city, favorite_chocolat = clé/key

<pre> "city" => "Lyon", "favorite_chocolate" => "Dark",]; </pre>	
<pre> \$myNumber = 40; if (\$myNumber == 30) { echo "Exact"; } else { echo "Faux"; }; </pre>	<p>Nombre = 40 SI ma variable égal 30 echo EXACT OU echo FAUX</p>
<pre> if (isset(\$myNumber)) { echo "C'est bien 5"; } else { echo "Ce n'est pas 5"; }; </pre>	<p>vérifie l'existence d'une variable</p>
<pre> if (!empty(\$myNumber)) { echo "C'est bien 5"; } else { echo "Ce n'est pas 5"; }; </pre>	<p>vérifie que le contenu d'une variable n'est pas vide avec le !</p>
<pre> if (isset(\$myNumber) && !empty(\$myNumber)) { echo "C'est bien 5"; } else { echo "Ce n'est pas 5"; }; </pre>	<p>vérifie l'existence d'une variable ET qu'elle ne soit pas vide</p>
<pre> while(true){ echo "boucle infini"; } </pre>	<p>boucle infini</p>
<pre> echo \$informations["name"]; EX:\$informations = ["name" => "Naruto", "age" => 18, "favorite_food" => "Ramen",]; </pre>	<p>commande pour afficher une information dans une clé ex : affichera Naruto</p>

<pre> "favorite_color" => "Orange", "height" => 180, "is_alive" => true]; </pre>	
<p style="text-align: center;"><u>BOUCLE WHILE</u></p> <pre> \$number=0; while(\$number<10){ \$number++; echo "Mon nombre vaut désormais : \$number
"; } echo "On est sortit de la boucle"; </pre>	<p>Mon nombre vaut désormais : 1 Mon nombre vaut désormais : 2 Mon nombre vaut désormais : 3 Mon nombre vaut désormais : 4 Mon nombre vaut désormais : 5 Mon nombre vaut désormais : 6 Mon nombre vaut désormais : 7 Mon nombre vaut désormais : 8 Mon nombre vaut désormais : 9 Mon nombre vaut désormais : 10 On est sortit de la boucle</p>
<pre>die;</pre>	<p>Stop le reste de PHP</p>
<p style="text-align: center;"><u>BOUCLE FOREACH</u></p> <pre> \$aliments=["Tomatoes", "Potatoes", "Chips", "Water", "Orange", "Saint-Moret"]; echo "Voici ma liste de course :"; foreach(\$aliments as \$aliment){ var_dump(\$aliment); die; }; </pre>	<p>Mon tableau s'appelle aliments, pour afficher les éléments de mon tableau, la liste s'arrête à TOMATOES car j'ai utilisé le DIE pour stopper ma boucle</p> <p>Voici ma liste de course : /var/www/html/index.php:91:string 'Tomatoes' (length=8)</p>
<p style="text-align: center;"><u>BOUCLE FOREACH</u></p> <pre> \$aliments=["Tomatoes", "Potatoes", "Chips", "Water", "Orange", "Saint-Moret"]; echo "Voici ma liste de course :
"; </pre>	<p>Pour afficher le contenu de mon tableau sans utiliser les [xxx]</p> <p>Voici ma liste de course : Tomatoes Potatoes Chips Water Orange Saint-Moret</p>

<pre>foreach(\$saliments as \$saliment){ echo \$saliment; };</pre>	
<p style="text-align: center;"><u>FUNCTION</u></p> <pre>function sayHello(){ echo "Hello"; };</pre>	
<pre>sayHello();</pre>	<p>permet d'afficher la function</p>
<p><u>FUNCTION AVEC VARIABLE (SCOPE)</u></p> <pre>function sayHello(){ \$message="Hello"; echo \$message; }; sayHello();</pre>	<p>utiliser une variable dans une function, si je veux l'afficher, je dois uniquement mettre le echo dans ma function</p>
<p><u>FUNCTION AVEC PLUSIEURS PARAMETRES</u></p> <pre>\$exemple1="ciel"; \$exemple2="nuage"; function nature(\$exemple1, \$exemple2){ echo "Voici le \$exemple1 et ici les \$exemple2"; }; nature(\$exemple1, \$exemple2);</pre>	

TEMPLATING

1. créer un dossier `Template` dans l'IDE
2. créer des fichiers `header.php` et `footer.php`
 - a. `header` : DOCTYPE HTML + Link CSS
 - b. `footer` : SANS DOCTYPE HTML + script
3. dans le fichier `index.php`, ajoute les fichiers `header.php` et `footer.php` avec :
 - i. `include"template/header.php";`
 - ii. `include"template/footer.php";`
 1. possibilité d'utiliser `REQUIRE` sauf qu'il déclenche une `FATAL ERROR` qui stop le chargement de la page contrairement à l'`INCLUDE` qui déclenche un `WARNING`

Démarrer une session :

`session_start(xxx);` À AJOUTER DANS MON FICHIER INDEX TOUT EN HAUT
pour vérifier : inspector navigateur => application => cookie (name, value etc)

```
$_SESSION["username"]="Halloween";  
echo "Bonjour, $_SESSION[username]";s
```

```
session_id("Toto");
```

À compléter FORMULAIRE => METHOD : GET (information s'affiche dans l'url, non secure) & POST (envoyé dans une base de donnée)

CTRL + MAJ + R pour effacer les caches