

# Le Terminal (ou la ligne de commande)

Si vous souhaitez devenir hacker, ou simplement développeur web agérie il est obligatoire d'apprendre à utiliser un terminal (ou ligne de commande) !

## Les commandes de base

Un terminal est ouvert dans un répertoire. Nous pouvons à l'aide de commande manipuler notre ordinateur :

- `pwd` : (Print Working Directory) affiche le répertoire dans lequel vous vous situez
- `ls` : (list) affiche tout les fichiers et dossier du répertoire dans lequel on se trouve. Nous pouvons utiliser les options : `lah` afin d'afficher bien plus d'information ( `ls -lah` )
- `clear` : Efface le contenu du terminal
- `cd` : (Change Directory) permet de changer de répertoire :
  - `cd nomDuRepertoire` : Se déplace dans un dossier
  - `cd ..` : Revenir dans le dossier parent
  - Vous pouvez utiliser la touche TAB pour autocompléter
- `mkdir` : (Make Directory) créer un nouveau dossier
- `touch` : (Toucher) créer un nouveau fichier vide
- `cp` : (Copy) copie / colle un fichier
- `mv` : (Move) déplace un fichier
- `rm` : (Remove) supprime un fichier ou un dossier
  - Pour supprimer un dossier il faut remplir un condition :  
Le dossier doit être vide !
  - Pour s'assurer et obliger `rm` à supprimer un dossier et ce qu'il contient il faut utiliser les options : `'rf'`  
(ex: `rm -rf NomDossier` )

## Les binaires

Nous avons dans notre ordinateurs des programmes que l'on nomme les « binaires ». Ces derniers sont de simple fichiers écrit souvent dans un langage de programmation (JavaScript, Python, C, C++ etc ...).

Ces programmes (ou fichiers) sont rangé dans répertoires spécifique de notre système. Pour connaître ces répertoires il faut taper la commande :

```
echo $PATH
```

## Utiliser php et composer

Lorsque l'on développe un programme dans un langage il faut pouvoir le lancer (ou dit exécuter le programme). Pour cela il existe 2 grandes familles de langage de programmation :

## Les langages compilés :

Ce sont des langages que l'on va « compiler » (transformer en binaire (des 0 et des 1)).

Les avantages principaux :

- Se sont des langages très rapides

Les inconvénients :

- Se sont des langages plus difficile
- Ce n'est pas portable.

Les langages les plus célèbres :

1. C++
2. C
3. Pascal
- (4. Python)
- (5. JavaScript)

## Les langages interprétés:

Ce sont des langages que l'on va « interpréter » (c'est un programme de notre ordinateur qui s'occupe de TOUT !)

Les avantages :

- Se sont des langages portables
- Se sont des langages plus faciles

Les inconvénients :

- Se sont des langages plus lents

Les langages les plus célèbres :

1. Javascript
2. Python
3. PHP

## Utiliser php

Lorsque nous avons écrit un fichier php, il est possible de l'exécuter sur notre ordinateur sans passer par un navigateur. Pour cela il suffit de lancer la commande :

```
php monScript.php
```

Il est possible de lancer un REPL, une console PHP :

```
php -a
```

Vous pouvez appuyer sur CTRL-C pour quitter la console PHP

## Lancer votre propre Serveur !

Il est possible d'utiliser php pour lancer son propre serveur web. C'est un petit programme en ligne (c'est à dire sur une adresse) qui nous permet d'afficher et d'exécuter du php pour le web.

```
php -S localhost:5353 -t .
```

Vous pouvez appuyer sur CTRL-C pour quitter le serveur php

## Et composer ?

Composer c'est un petit outil qui permet de créer des projets php maintenable, en utilisant l'orientée objet et permettant de télécharger des librairie que d'autres utilisateurs php ont créés.

## Initialiser composer

Afin d'utiliser composer et d'autres librairies PHP, il faut tout d'abord l'initialiser. Pour cela :

```
composer init
```

## Mettre en place l'autoload

Afin de commencer à utiliser nos classes il faut pour cela installer l'autoload :

```
composer install
```

## Utiliser l'autoload

Maintenant nous pouvons facilement importer l'autoload de composer depuis n'importe quel dossier. Les classes seront dès lors automatiquement incluses :

```
require __DIR__ . '/../chemin/vers/vendor/autoload.php';
```

## Écrire des classes PHP

Afin d'écrire notre code en Orienté objet, il faut pour cela respecter certaines conventions :

- Le répertoire `src` doit correspondre au namespace `App`
- Chaque classe possède son propre fichier PHP
- Chaque fichier doit être nommé comme le nom de la classe, à l'identique

## Structure d'un projet PHP

Les projets PHP sont structurés en suivant la même logique qu'un système linux. Voici les répertoires les plus courants :

- Le répertoire `bin` : Contient les scripts « exécutable » en ligne de commande
- Le répertoire `public` : Contient notre site internet (les différentes pages ou le frontend controller ...)
- Le répertoire `src` : Contient l'intégralité de nos classes
- Le répertoire `vendor` : Contient l'autoload et les librairies php installées

