

PHP – COURS 01 - Introduction à la programmation

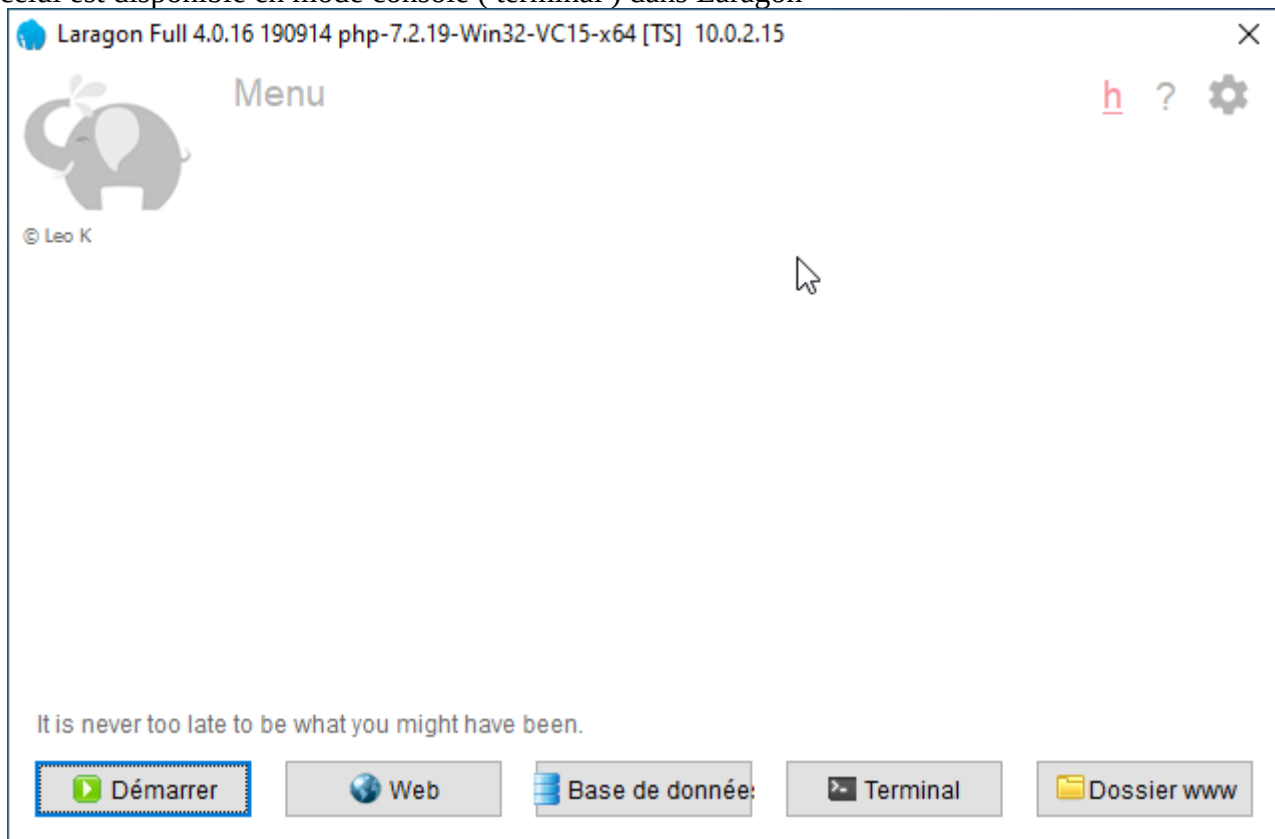
INTRODUCTION

PHP (Hypertext Preprocessor) est avant tout un langage dédié à la génération de contenu pour le Web. Le client (un navigateur par exemple) demande l'exécution d'une "page PHP" (en fournissant d'éventuels paramètres) à un serveur, ce dernier sous-traite la demande à un processeur PHP qui interprète le code et retourne un résultat au serveur qui le renvoie au client.

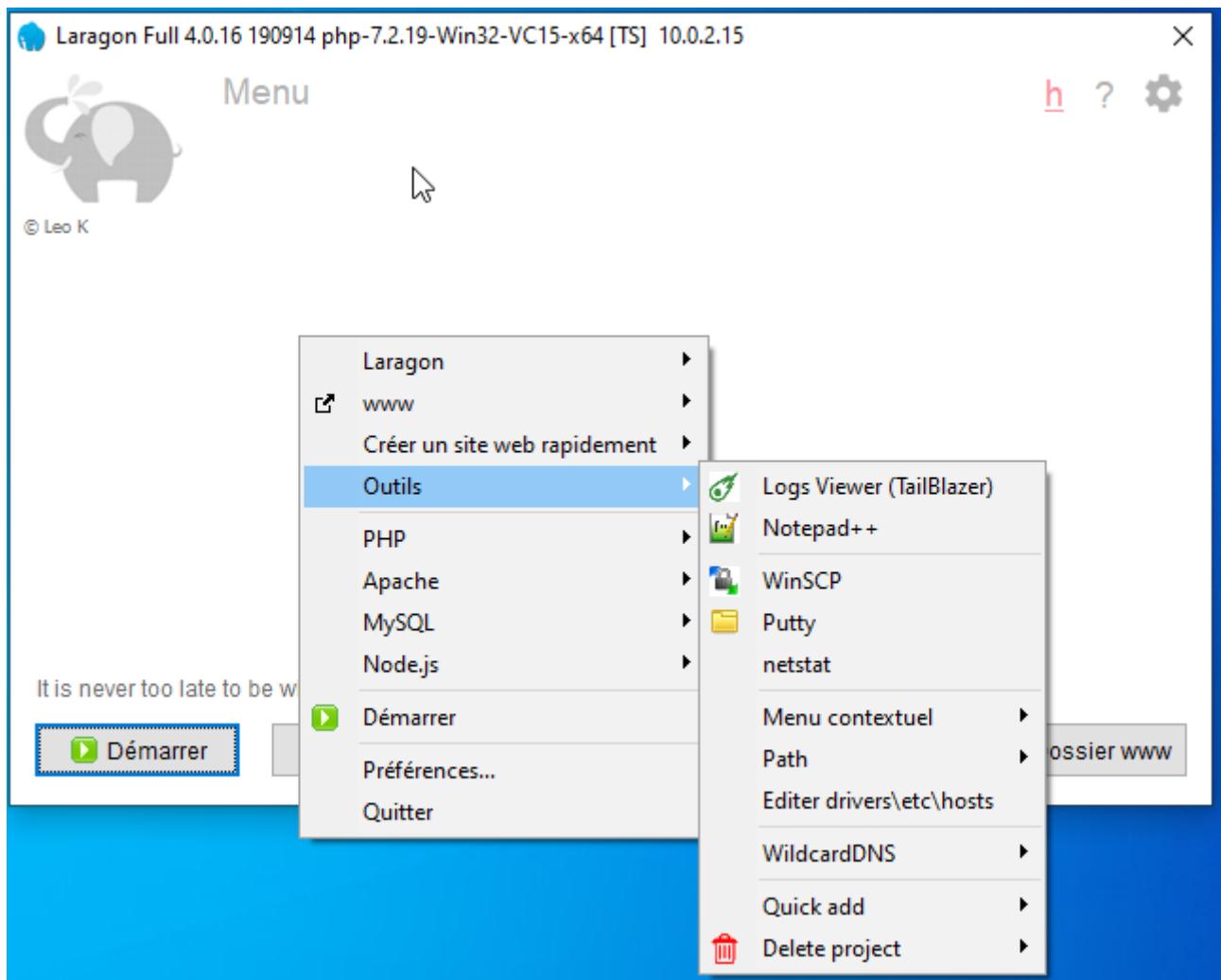
Mais dans ce cours, nous l'utiliserons de manière à faire des programmes autonomes, dans le but d'avoir une initiation aux principes de base de la programmation. Les programmes seront exécutés depuis une console sous Linux ou invite de commande sous WINDOWS.

OUTILS NÉCESSAIRES

Pour écrire et exécuter un programme, vous aurez besoin d'un éditeur de texte, sous Windows vous pouvez utiliser **NotePad++**, un éditeur de texte disponible gratuitement. Vous avez également besoin l'interpréteur PHP, qui servira à exécuter vos programmes, au lycée celui est disponible en mode console (terminal) dans Laragon

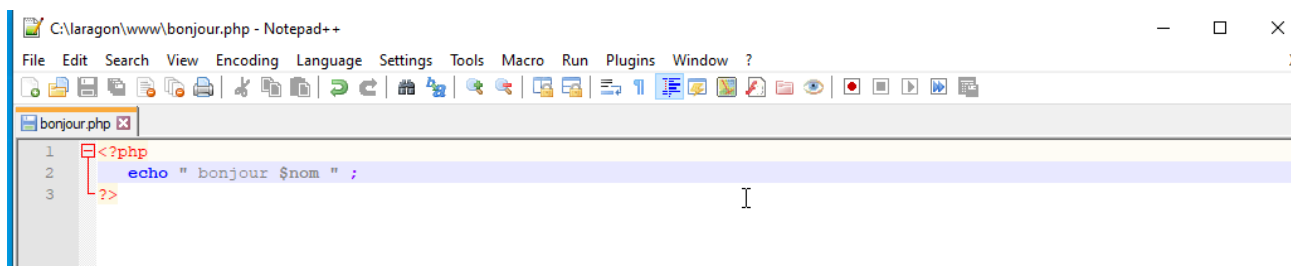


Pour NotePad++ dans Laragon clic droit :

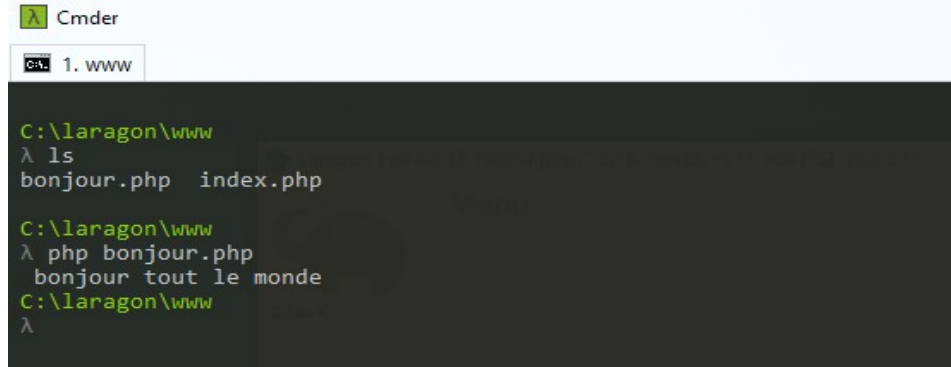


UN EXEMPLE DE PROGRAMME

bonjour.php



Pour exécuter un script PHP, il faut ouvrir une console (terminal) puis déclencher l'interpréteur PHP en lui passant en paramètre le nom du script. Exemple: php bonjour.php



```
C:\laragon\www
λ ls
bonjour.php  index.php

C:\laragon\www
λ php bonjour.php
bonjour tout le monde

C:\laragon\www
λ
```

Si votre programme ne s'exécute pas vous devez :

- vérifier que l'interpréteur PHP est installé : php -v
- vérifier que le programme se situe bien à l'endroit où vous souhaitez l'exécuter
- vérifier la syntaxe du programme

QUELQUES RÈGLES DE BASE

1- Un script PHP doit être placé à l'intérieur d'une paire de balises, il existe plusieurs styles de balises, celles que nous utiliserons s'écrivent ainsi :

```
<?php
instructions PHP;
?>
```

2 – Les instructions PHP doivent se terminer par un point virgule.

3 - Pour afficher une information, vous pouvez utiliser 2 instructions: print ou echo.

Syntaxe :

```
print (expression );
echo (expression );
```

4 – Un fichier contenant des instructions PHP doit avoir l'extension « .php »

LES VARIABLES

Le rôle d'une variable est de stocker une information temporairement dans la mémoire vive le temps de l'exécution d'un programme. Les noms des variables doivent obligatoirement commencer par le caractère dollar (\$). Ils sont sensibles à la case : \$prix et \$Prix ne désignent pas la même variable. Par convention le nom d'une variable commence par une minuscule et une majuscule au début de chaque nouveau terme.

exemple :

- \$prix=100
- \$nombreArticle=5

RÉCUPÉRER UNE INFORMATION SAISIE AU CLAVIER

PHP n'étant pas conçu pour des programmes autonomes, les données sont normalement saisies dans des formulaires WEB. Cependant, il existe une fonction appelée `readline()` qui permet à un programme PHP de récupérer les données saisies au clavier.

Pour interagir avec l'utilisateur, voici le code PHP :

```
<?php
$nom = readline(" entrer votre nom : ");
echo " Bonjour $nom ";
?>
```

Lorsque PHP atteindra cette ligne, le script se "bloquera". L'utilisateur pourra alors entrer du texte dans la console, et pourra l'envoyer dans `$nom` à l'aide de la touche **Entrée**. le script se "débloquera" alors.

INITIALISER OU METTRE À JOUR UNE VARIABLE

l'instruction :

```
$prix = 10 ;
```

se lit de **droite à gauche** : la valeur 10 est affectée à la variable `$prix`.

Modifions légèrement le programme :

```
<?php
$prix = 10;
$cout = $prix;
echo ($cout);
?>
```

La deuxième instruction se lit donc : la valeur de la variable `$prix` est affectée à la variable `$cout`. La variable `$prix` est de type numérique entier. D'autres types de données sont disponibles en PHP. Le type d'une variable est déterminé automatiquement par PHP lors de l'affectation.

Attention comme beaucoup de langages de programmation, PHP note l'affectation par le symbole traditionnellement réservé à l'égalité (=). A ne pas confondre avec le symbole de test d'égalité (==).

CALCULER - COMPARER

Opérateurs mathématiques

<code>\$a+\$b</code>	addition
<code>\$a-\$b</code>	soustraction
<code>\$a/\$b</code>	division
<code>\$a*\$b</code>	multiplication
<code>\$a%\$b</code>	modulo ->reste entier de la division

Opérateurs de comparaison

<code>\$a>\$b</code>	supérieur
<code>\$a>=\$b</code>	supérieur ou égal
<code>\$a<\$b</code>	inférieur
<code>\$a<=\$b</code>	inférieur ou égal
<code>\$a==\$b</code>	égal
<code>\$a!=\$b</code>	différent

LISTE DES CARACTÈRES SPÉCIAUX

L'antislash (\) est un caractère spécial, on peut l'utiliser dans les cas suivants :

Un saut de ligne	\n
Une tabulation horizontale	\t
Un antislash \	\\
Un caractère \$	\\$

La principale difficulté au début sera de vous familiariser avec ces outils et de travailler en mode console. Pour cela, vous devez connaître quelques commandes de base notamment savoir comment vous déplacer dans l'arborescence de votre espace de travail.

Command's Purpose	MS-DOS	Linux	Basic Linux Example
Copies files	copy	cp	cp <i>thisfile.txt</i> /home/ <i>thisdirectory</i>
Moves files	move	mv	mv <i>thisfile.txt</i> /home/ <i>thisdirectory</i>
Lists files	dir	ls	ls
Clears screen	cls	clear	clear
Closes prompt window	exit	exit	exit
Displays or sets date	date	date	date
Deletes files	del	rm	rm <i>thisfile.txt</i>
"Echoes" output on the screen	echo	echo	echo <i>this message</i>
Edits files with simple text editor	edit	pico [a]	pico <i>thisfile.txt</i>
Compares the contents of files	fc	diff	diff <i>file1 file2</i>
Finds a string of text in a file	find	grep	grep <i>this word or phrase</i> <i>thisfile.txt</i>
Formats a floppy	format a: (if floppy's in A:)	mke2fs (or mformat [b])	/sbin/mke2fs /dev/fd0 (/dev/fd0 is the Linux equivalent of A:)
Displays command help	command /?	man [c]	man <i>command</i>
Creates a directory	mkdir	mkdir	mkdir <i>directory</i>
Screens through a file	more	less [d]	less <i>thisfile.txt</i>
Renames a file	ren	mv	mv <i>thisfile.txt</i> <i>thatfile.txt</i> [e]
Shows your location in the file system	chdir	pwd	pwd
Changes directories with a specified path (<i>absolute path</i>)	cd <i>pathname</i>	cd <i>pathname</i>	cd / <i>directory/directory</i>
Changes directories with a <i>relative path</i>	cd ..	cd ..	cd ..
Displays the time	time	date	date
Shows amount of RAM and use	mem	free	procinfo