

# Chapitre 1

## Introduction

### 1.1 Qu'est-ce que PERL

PERL est l'acronyme de "Practical Extraction and Report Language" ou "Pathologically Eclectic Rubbish Lister", son inventeur Larry Wall n'ayant pas encore tranché entre ces deux possibilités. Ce langage a été conçu à l'origine pour écrire des programmes (si possible réutilisables) allant au-delà de certaines limites des scripts shell UNIX. Cette image de super langage shell est une assez bonne définition de ce qu'est PERL. Il permet d'écrire rapidement de petits programmes qui officient là où les manipulations manuelles seraient inenvisageables parce que trop répétitives ou ardues.

Une des forces majeures de PERL est sa capacité à pouvoir manipuler fichiers et chaînes de caractères, comme nous le verrons au fil de ce cours. Ceux qui se sont risqués à renommer huit cents fichiers à la main via un explorateur quelconque savent à quel point un tel langage est salvateur.

Pourquoi PERL ? Il existe bien sûr d'autres langages pouvant remplir un office comparable à PERL à commencer par le shell UNIX. Nous citerons aussi PYTHON ou RUBY. Alors pourquoi PERL plutôt qu'un de ces autres langages ? Contrairement aux scripts UNIX, PERL est portable et fonctionne sur de nombreuses plateformes. Il n'est pas à la base orienté objet (si vous ne savez pas ce que cela signifie, vous pouvez le voir comme une conception avancée d'un langage de programmation) simplifiant ainsi son apprentissage. Avouons aussi que le fait que le rédacteur de ces notes soit un adepte de ce langage n'est pas sans rapport avec ce choix.

### 1.2 Comment installer PERL

Si vous travaillez sur un système UNIX, BSD ou MAC OS X, il y a de fortes chances que PERL soit déjà présent. Ouvrez un terminal et tapez la commande `perl -v`. Si `perl` est présent sur votre système, vous devriez obtenir un message du style

*This is perl, v5.8.4 built for x86\_64-linux*

*Copyright 1987-2004, Larry Wall*

*Perl may be copied only under the terms of either the Artistic License or the GNU General Public License, which may be found in the Perl 5 source kit.*

*Complete documentation for Perl, including FAQ lists, should be found on this system using 'man perl' or 'perldoc perl'. If you have access to the Internet, point your browser at <http://www.perl.com/>, the Perl Home Page.*

Dans le cas contraire (ce qui signifie que ce n'est pas votre jour de chance, vous êtes tombés sur le seul système UNIX-BSD où PERL n'est pas fourni en standard), je vous invite à consulter l'aide à l'installation de logiciels de votre système.

Pour toutes les autres plates-formes, il existe un site web<sup>1</sup> indiquant où obtenir PERL pour chacune d'entre elles.

Comme quelques irréductibles continuent d'utiliser Windows, nous allons détailler l'installation de PERL sur ce système. Il vous faudra tout d'abord récupérer la dernière version en date sur le site d'Active Perl<sup>2</sup>. La version dite MSI possède un assistant d'installation, les options par défaut suffisent. Profitez de l'occasion pour télécharger Open-PERL-IDE<sup>3</sup>, un environnement de développement PERL, qui consiste en un fichier zip contenant le fichier exécutable qui est le programme lui-même (décompressez l'archive là où bon vous semble).

Si vous ne souhaitez pas installer PERL, sachez qu'il est disponible en démarrant sur un CD de KNOPPIX<sup>4</sup>. Vous pourrez ainsi apprendre à utiliser ce langage sans avoir à l'installer.

### 1.3 La documentation

Il existe deux livres incontournables pour PERL. Le premier est *Introduction à PERL* de R. Schwartz et T. Christiansen aux éditions O'Reilly. Cet ouvrage est ce qui se fait de mieux dans ce domaine et sa qualité d'écriture en fait sûrement le meilleur ouvrage informatique (voire littéraire pour tout bon geek qui se respecte). Il est aussi suffisant pour la majorité des utilisateurs de base. Le deuxième livre est le livre de référence de Larry Wall, *Programmation en PERL*, chez O'Reilly. Cet ouvrage est la Bible de PERL, mais je déconseille son usage à tout débutant sous peine d'allergie définitive à ce langage (pour les initiés, ce serait comme commencer la mécanique du point avec le Landau ou la théorie des champs avec le Zinn-Justin).

Sous UNIX et BSD, les pages man sont d'un grand secours, en particulier *man perl* et *man perlfunc*. Dans la même veine, il existe *man faq* disponible aussi sur le web<sup>5</sup> qui répondra à une majorité des questions simples. Sous Windows, Open-PERL-IDE contient une aide en ligne très complète (issue des pages man).

Le web regorge de documentations et autres introductions à PERL. En faire une liste même non exhaustive serait trop long (une petite recherche sur votre Google favori vous en convaincra). Je vous renvoie en particulier à [www.perl.com](http://www.perl.com)<sup>6</sup> qui contient l'équivalent des pages man, ce qui est un atout important lorsque vous cherchez une fonction ou une syntaxe précise.

### 1.4 Hello World !

Vous devez disposer maintenant de PERL sur votre ordinateur. Vous devez sentir comme un besoin impétueux d'écrire votre premier programme. Pour taper un programme en PERL, vous pouvez prendre n'importe quel éditeur de texte ASCII comme (x)emacs, vi(m), nedit, jed, nano, pico, kwriter... pour le monde UNIX, BSD, MAC OS X. Sachez cependant que certains éditeurs permettent de colorer la syntaxe du code, le rendant de ce fait plus lisible. Sous Windows, Notepad, winedt ou write feront dans un premier temps l'affaire. Si vous le souhaitez, vous pouvez utiliser le logiciel libre Open-PERL-IDE qui offre un véritable environnement de développement PERL sous Windows.

---

<sup>1</sup><http://www.cpan.org/ports>

<sup>2</sup><http://www.activestate.com/Products/Download/Download.plex?id=ActivePerl>

<sup>3</sup><http://open-perl-ide.sourceforge.net>

<sup>4</sup><http://www.knoppix.org>

<sup>5</sup><http://faq.perl.org>

<sup>6</sup><http://www.perl.com/pub/q/documentation>

Nous commencerons par le traditionnel et indémodable "Hello World" :

```
1.  #!/usr/bin/perl
2.
3.  # premier exemple : Hello World!
4.  print ("Hello World!\n");
```

Les numéros de lignes ne sont pas à recopier, ils ne servent que de repère pour les explications qui suivent. Prenez garde à la casse en recopiant ce programme. Majuscules et minuscules ne sont pas équivalentes en PERL. Sauvegardez ce fichier sous **helloworld.pl** (l'extension .pl désigne en général un script PERL). Pour exécuter ce script, vous avez plusieurs possibilités :

- Sous UNIX-BSD, ouvrez un terminal et placez vous dans le répertoire contenant le fichier **helloworld.pl** et tapez la commande `perl helloworld.pl`. Une autre technique consiste à rendre ce script exécutable en tapant `chmod u+x helloworld.pl`. Pour exécuter ce script, il vous suffira alors d'un simple `./helloworld.pl`. Il se peut que vous obteniez un message d'erreur du type `sh : ./helloworld.pl : /usr/bin/perl : bad interpreter : No such file or directory`. Dans ce cas, vérifiez le chemin pour accéder à la commande `perl`. Ceci peut être fait grâce à la commande `whereis perl`, le premier chemin indiqué par cette commande devant être utilisé en lieu et place de `/usr/bin` indiqué à la première ligne du script.
- Sous Windows, si vous avez tapez ce programme à l'aide de OpenPERL-IDE, il vous suffit de cliquer sur run. Pour les intrépides, vous pouvez aussi double cliquer sur le fichier. Mais en général, Windows ouvre et ferme immédiatement la fenêtre dans laquelle il a exécuté votre script. il est aussi possible de passer par la ligne de commande. Placez vous alors dans répertoire contenant votre script et tapez `perl helloworld.pl`.

Vous venez d'afficher un joli Hello World à l'écran. Essayons de comprendre le fonctionnement de ce programme très simple. Tout ce qui se situe après un dièse est considéré par PERL comme un commentaire. Les lignes 1 et 3 ne sont donc pas traitées. De même PERL ne tient pas compte des lignes vides et des tabulations, des espaces (qui ne sont pas inclus dans une chaîne de caractères bien sûr). Vous pouvez en user et abuser pour rendre votre code plus lisible. Pour être tout à fait rigoureux, la première ligne est utilisée dans le monde UNIX/BSD si vous souhaitez rendre votre script exécutable. Elle correspond à l'invocation de PERL pour interpréter le reste du script.

La quatrième ligne est la seule ligne non triviale. La commande `print` permet d'afficher la chaîne de caractères entre guillemets. Notez que la ligne se termine par un point virgule comme dans d'autres langages (comme le C et le Java, ..). Ce caractère est impératif, c'est lui qui indique véritablement la fin de la ligne de commande et non le saut de ligne. Vous pouvez d'ailleurs concaténer plusieurs lignes de commandes sur la même ligne de texte, du moment que chacune d'entre elle est séparée par un point virgule (`print ("toto"); print ("tata");`). Les parenthèses qui entourent la chaîne de caractères ne sont pas obligatoires, PERL est relativement souple sur sa syntaxe. Pour ceux qui viennent du monde C, Java, ... ou qui souhaitent y passer un jour, mieux vaut ajouter ces parenthèses (elles sont obligatoires dans ces autres langages).

Concernant le texte lui-même, il ne présente rien de particulier hormis le caractère `\n` qui signifie le saut de ligne. PERL utilise la même convention que le C concernant ce type de symbole. En particulier, pour obtenir un caractère `\` dans une chaîne de caractères, il faut écrire `\\`.

A partir de maintenant, vous devez être en mesure de recopier et d'exécuter n'importe quel programme écrit en PERL. Il ne vous reste plus qu'à étudier le reste de ces notes pour maîtriser ce langage.