```
## Notes

# Perl - Practical Extraction and Report Language - http://www.perl.org/
# Pathologically Eclectic Rubbish Lister

# Perl is designed to make easy job easy, without making the hard job
# impossible - Programming Perl - Larry Wall

# Used in: CGI, graphics programming, system administration, network
# programming, finance, _bioinformatics_, and...much more.

# 21000 CPAN modules (Comprehensive Perl Archive Network) http://www.cpan.org/
# BioPerl - http://www.bioperl.org/wiki/Main_Page

# references
# http://perldoc.perl.org/perlintro.html


## Basics

# Our first perl script
#!/usr/bin/env perl     # This line tells linux to use perl to interpret
print "Hello World!\n"; # This line prints the message

# How to run a program
chmod +x <prog.pl> # To be typed in the console
perl prog.pl       # To be typed in the console

# Test your snippets
perl -e '<code>'


## Variables

# scalars
$lang="French";
print "I speak $lang\n";

$fav_num=11;
print "My favourite number is: $fav_num\n";

# vectors
@langs=("Italian","English","Spanish","French");

print "My mother tongue is: $langs[0]\n";
print "I would like to speak @langs\n";
print "I would like to speak @langs[0..2] and $langs[$#langs]\n";
print "I would like to speak @langs[0,1,3]\n";

# hashes
%canada=(
    QC=>"Quebec",
    ON=>"Ontario",
    BC=>"British Columbia"
);

print "$canada{QC}\n";


## Condictional constructs

# string comparison (eq, ne)
$psswd="enigma";
if($psswd eq "secret"){
    print "OK!\n";
}
else{
    print "Access denied!\n";
}
```

```
# there is more than one way to do it!
unless($psswd eq "secret"){
    print "Access denied!\n";
}

$psswd="enigma";
print "Access denied\n" if $psswd ne "secret";

# numeric comparison
$my_num=5;
if($my_num<0){
    print "$my_num is lesser than 0\n";
}
elsif($my_num==0){
    print "$my_num is 0\n";
}
else{
    print "$my_num is grater than 0\n";
}


## Loops
# Intentation is not mendatory BUT STRONGLY ADVISED!

# while
$num=1;
while($num<=10){
    print "$num\n";
    $num++;
}


# do-while
$num=1;
do
{
    print "$num\n";
    $num++;
} while($num<=10)


# for
for($num=1;$num<=10;$num++){
    print "$num\n";
}


# until
$num=0;
until($num==10){
    $num++;
    print "$num\n";
}


# foreach
@array=("Québec","Montreal","Cap-Chat");
foreach $town (@array){
    print "$town\n";
}


## I/O (Input/Output)

# read
open(IN, "<my_file");
$first_line=<IN>;
@data=<IN>;
print @data;
close(IN);
```

```
open(IN,"<my_file");
while($line=<IN>){
    print $line;
}
close(IN);

# die
open(IN,"<my_file") or die "ERROER! I cannot read the file!\n";
while($line=<IN>){
    print $line;
}
close(IN);

# write (erase file if it exists, otherwise create it)
$phrase="I like programming!\n";
open(OUT,">file");
print OUT $phrase;
close(OUT);


# append (to the end of a file if it exist, otherwise create it)
$phrase="I like programming!\n";
open(OUT,">>file");
print OUT $phrase;
close(OUT);


# system commands
$r=`date`;
print $r;


# references
%canada=(
    QC=>("Montreal","Quebec","Cap-chat"),
    ON=>("Toronto","Ottawa")
);
$ref=\$canada;
print "$ref\n";
```