

# Learning and Understanding Perl

- [Introduction](#)
- [Perl Online References](#)
- [Perl Basics](#)
  - [Perl Data Types](#)
  - [Use it!](#)
  - [Perl if statement](#)
  - [Perl Loops -> While](#)
  - [Perl Loops -> for](#)
  - [Perl Loops -> foreach](#)
  - [Open a file and loop through](#)

## Introduction

You DO not need to learn Perl to use NMIS, Perl is however a powerful, elegant and thoughtful language for solving real problems. You can do low level things like C with the ease of scripts like BASH, and everything in between. BUT the real power of Perl is [CPAN](#), the Comprehensive Perl Archive Network, someone somewhere has probably already solved your problem and published something on CPAN.

## Perl Online References

Some great web pages to learn about Perl

- [Perl Tutorial for Beginners: Learn in 1 Day](#)
- [Learn Perl in about 2 hours 30 minutes](#)
- [Introduction to Perl](#)
- [Perl Tutorial Videos - Online Training](#)
- [Perl 5 Tutorials | Essential Training](#)
- [Beginning Perl \(free book\) - www.perl.org](#)

Please comment if you know some more good ones.

## Perl Basics

### Perl Data Types

- Scalar -> \$variable
- Array -> @array
- Associative Array (hash) -> %hash
- Combinations to make complex types easily (looks confusing but very powerful)
- Array of hashes \$array[\$i]->{\$key}
- Hash of hashes \$hash{\$key}{\$var}
- Multi-dimensional \$var->{\$key}->[0]->{\$thing}

### Use it!

```
use strict;
```

### Perl if statement

```

use strict;
my $string = "this is a string";
if ( $string eq "string" ) {
    print "$string is the same as \"string\"\n";
}
elsif ( $string =~ /string/ ) {
    print "regex match $string has \"string\" in it\n";
}
elsif ( $string == 100 ) {
    print "$string is the number 100\n";
}
else {
    print "Else Nothing\n";
}

```

## Perl Loops -> While

```

while (condition) {
    # do something
}

```

## Perl Loops -> for

```

use strict;
my @array = (1, 2, 3, 4, 5, 6);
for ( my $i = 0; $i <= $#array; ++$i ) {
    print "i=$array[$i]\n";
}

```

## Perl Loops -> foreach

```

use strict;
my @array = (1, 2, 3, 4, 5, 6);
foreach my $i ( @array ) {
    print "i=$i\n";
}

```

## Open a file and loop through

```

use strict;
my $match = "blah";
my $file = "textfile.txt";
my $lines = 0;
open (DATA, $file) or die "ERROR with $file. $!";
while (<DATA>) {
    chomp; # not necessary but gets rid of trailing spaces and newlines.
    if ( $_ =~ /$match/ ) {
        print "$lines: $_\n";
    }
    ++$lines;
}

```