For LOOP with ARRAY

`````````````````````````````````

#!/usr/bin/perl -w

use strict;

my @arr1 = ("one", "two", "three");

for my $i (@arr1){

print "$i\n";

}

# perl script.pl

-------------------------------------------------------------------------

The for loop is one builtin which assigns to the $\_ by default. If you don’t specify a variable name, $\_ is assumed:

#!/usr/bin/perl -w

use strict;

my @arr1 = ("one", "two", "three");

for (@arr1){

print "$\_\n";

}

-------------------------------------------------------------------------

$#arr is the total length of the array ‘arr’

#!/usr/bin/perl -w

use strict;

@arr1 = (1..10);

for (my $i=0; $i<=$#arr1; $i++){

print "$arr1[$i]\n";

}

# perl script.pl

---------------------------------------------

---------------------------------------------

exit with exit status

``````````````````````````String Comparison

#!/usr/bin/perl

use strict;

use warnings;

print "Love is part of life not life! Agree? (y/n): ";

chomp(my $a = <>);

if ($a eq "y"){

print "Just kidding bruhhhh\n";

}elsif($a eq "n"){

print "Kill yourself\n";

}else{

exit;

}

With exit code/status:

``````````````````````

#!/usr/bin/perl

use strict;

use warnings;

print "Love is part of life not life! Agree? (y/n): ";

my $a = <>;

chomp($a);

if ($a eq "y"){

print "Just kidding bruhhhh\n";

}elsif($a eq "n"){

print "Kill yourself\n";

}else{

exit 127;

}

----------------------------------------

---- For printing exit status ----

echo $?

+++++++++++++++++++++++++++++++++++++++

while loop:

```````````

while (condition is true){

statement

}

+++++++

# vim script.pl

#!/usr/bin/perl -w

use strict;

my $a = "true";

while ($a) ## Loop goes into the infinite condition.

{

print "Using true\n";

}

+++++++

Diamond operator (<>).

````````````````````````

#!/usr/bin/perl

while(<>) {

...

}

++++++++++++++++++++++++++++++++++++

#!/usr/bin/perl

while(<>) {

print $\_;

}