In the name of who surround me with blessing that I can not count

Perl or Practical Extraction and Report Language

by `Ahmadreza` Anaami

Perl is a programming language developed by Larry Wall, especially designed for **text processing**. It stands for Practical Extraction and Report Language.

however now used for a wide range of tasks including system administration, web development, network programming, GUI development, and more.

Perl is Interpreted

Perl is an interpreted language, which means that your code can be run as is, without a compilation stage that creates a non portable executable program.

Traditional compilers convert programs into machine language. When you run a Perl program, it's first compiled into a byte code, which is then converted into machine instructions.

Getting Perl Installation

Perl Official Website – https://www.perl.org/

Perl Documentation Website – https://perldoc.perl.org

download link : https://cdna.download.ir/dlir-software/ActivePerl-v5.28.1.0000.rar

```
print("Hello, world\n");
print "Hello, world\n";
```

Perl File Extension

As a Perl convention, a Perl file must be saved with a .pl or .PL file extension in order to be recognized as a functioning Perl script. File names can contain numbers, symbols, and letters but must not contain a space. Use an underscore (_) in places of spaces.

comments

```
# This is a comment in perl

=begin comment
This is all part of multiline comment.
You can use as many lines as you like
=cut
```

```
$a = 10;
print "Value of a = $a\n";
print 'Value of a = $a\n';
```

```
Value of a = 10
Value of a = $a\n$
```

Escaping Characters

```
$result = "This is \"number\"";
print "$result\n";
print "\$result\n";
```

```
This is "number"
$result
```

Creating Variables

Scalars

```
$age = 25;             # An integer assignment
$name = "John Paul";   # A string
$salary = 1445.50;     # A floating point
```

Arrays

```
@ages = (25, 30, 40);
@names = ("John Paul", "Lisa", "Kumar");

print "\$ages[0] = $ages[0]\n";
print "\$ages[1] = $ages[1]\n";
print "\$names[0] = $names[0]\n";
print "\$names[2] = $names[2]\n";
```

Arrays

```
@names = ('John Paul', 'Lisa', 'Kumar');

@copy = @names;
$size = @names;

print "Given names are : @copy\n";
print "Number of names are : $size\n";
```

```
Given names are : John Paul Lisa Kumar
Number of names are : 3
```

```
@array = (1, 2, 'Hello');
@array = qw/This is an array/;
```

```perl
@array = (1,2,3);
$array[50] = 4;

$size = @array;
$max_index = $#array;

print "Size:  $size\n";
print "Max Index: $max_index\n";
```

```
Size: 51
Max Index: 50
```

Adding and Removing Elements in Array

```perl
# create a simple array
@coins = ("Quarter","Dime","Nickel");
print "1. \@coins  = @coins\n";

# add one element at the end of the array
push(@coins, "Penny");
print "2. \@coins  = @coins\n";

# add one element at the beginning of the array
unshift(@coins, "Dollar");
print "3. \@coins  = @coins\n";

# remove one element from the last of the array.
pop(@coins);
print "4. \@coins  = @coins\n";

# remove one element from the beginning of the array.
shift(@coins);
print "5. \@coins  = @coins\n";
```

more info about Perl arrays

https://www.tutorialspoint.com/perl/perl_arrays.htm

IF...ELSE

```perl
$a = 1;

if($a == 1){
   print("Welcome to the Perl if tutorial!\n");
   print("another form of the Perl if statement\n");
}
```

```perl
my $a = 1;
my $b = 2;

if($a == $b){
   print("a and b are equal\n");
}elsif($a > $b){
   print("a is greater than b\n");
}else{
   print("a is less than b\n");
}
```

The ? : Operator

```
$name = "Ali";
$age = 10;

$status = ($age > 60 )? "A senior citizen" : "Not a senior citizen";

print "$name is  - $status\n";
```

```
Ali is - Not a senior citizen
```

loops

```
my @a = (1..9);
for(@a){
        print("$_","\n");
}
# same as foreach
foreach $number (@a)
{
    print $number
}
```

```
for ($count = 1 ; $count <= 3 ; $count++)
{
    print "GeeksForGeeks\n"
}
```

while loop

```
$count = 3;
while ($count >= 0)
{
    $count--;
    print("GeeksForGeeks\n");
}
```

do while

```perl
$a = 10;

# do..While loop
do {

    print "$a ";
    $a = $a - 1;
} while ($a > 0);
```

more info

https://www.geeksforgeeks.org/perl-loops-for-foreach-while-do-while-until-nested-loops/