



In return for the given project, I chose the Perl programming language and tried to transfer it with my own comments.





Introduction

Before anything else, reason the name is spelled short is because it is written in lowercase letters to shorten anything :)

By the way, 'Perl' has this name just because it sounds good.

It also has humorous names.

Although some sources are also known as Practical Extraction and Reporting Language, they were not affected by anything.

History

- Perl was first created by Larry Wall on December 18, 1987, when he worked at a company called Unisys and then started working as a System Administrator at NASA. Someone who knows the C ++ language can continue without any difficulty.

Why should we use Perl?

1. Complex structures support and use, that means; For example, Perl programmers have the option to use references, class-based technical reference, word-covered variables, and packages.
2. One of the things Perl advocates is that it's very easy. Think and code your problems and things that will speed you up. The other thing is that this world is not just for computer scientists or anyone interested in it, anyone can do it, write it.
3. Perl is designed to speed you up so much that some of your code you have thought and written has led to a different event by saying option. Perl has a slogan, 'There is more than one way to try.' It offers you the one that is faster and more likely than you think.
4. Adding a database, Perl DBI supports most databases and is used. (Example MySQL, Oracle)
5. Its use in critical areas, in public, important places and personal areas is great.
6. In fact, more can be written here, just enough to make it incompatible with other languages.

How can we write in Perl?

First of all, there are ways to run it in Perl as in general in every programming language.

These are using online converters (IDE)

The second is with command lines.

Download instructions available on Perl's own site.

Perl runs on over 100 platforms!

We recommend that you always run the latest stable version, currently 5.32.1. If you're running a version older than 5.8.3, you may find that the latest version of CPAN modules will not work.

Unix/Linux



Included
(may not be latest)



GET STARTED

macOS



Included
(may not be latest)



GET STARTED

Windows



Strawberry Perl
&
ActiveState Perl



GET STARTED

I think every computer comes installed with Windows, Linux, Macintosh regardless of it.

To check

To Terminal or Console screen

perl -v

it is enough to write.

Example output on my device:

```
yasinozelci — -bash — 91x26
Yasin-MacBook-Pro:~ yasinozelci$ perl -v

This is perl 5, version 18, subversion 4 (v5.18.4) built for darwin-thread-multi-2level
(with 2 registered patches, see perl -V for more detail)

Copyright 1987-2013, 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.org/, the Perl Home Page.

Yasin-MacBook-Pro:~ yasinozelci$
```

Perl Variables

There are 3 different variables in Perl. Scalars, lists and hashes.

Because Perl is a very well (relax)-written language, it does not require you to specify a type when using it. The programmer must choose his own genre.

Scalar

The variable known as scalar must have a \$ sign in front of it. Scalar can take on more than one thing. These are numbers, characters etc.

```
$string="Yasin";  
$string5="Ozel";  
$number=183025  
$number3=7.3;
```

Lists

Arrays (lists) must precede them with the @ sign. The numerical index starting with 0 is an ordered list of scans.

```
$string = "Try" . " Perl" . " ---->";  
print "Result = $string";
```

```
Result = Try Perl ---->
```

Hashes

The % sign must be used mandatory. Irregular key-value group.

```
%data = ('YSN', 99.9);  
  
print "\$data{'YSN'} = $data{'YSN'}\n";  
  
$data{'YSN'} = 99.9
```

String Literals

String literals are strings of characters. They have different meanings.

Popular:

\t	Tab
\\	Backslash
\n	Newline
\r	Return
\b	Backspace
\a	Alarm (bell)
\e	Escape
\l	Lowercase next char
\u	Uppercase next char
\'	Single quote

```

$string = "String \t Literals";
print "$string\n";

$string = "String \n Literals";
print "$string\n";

$string = "String \a Literals"; #Alarm..By the way comment line
print "$string\n";

$string = "String \lLiterals";
print "$string\n";

$string = "String \L\iterals";
print "$string\n";

```

```

String  Literals
String
  Literals
String  Literals
String literals
String  LiTerals

```

Operators

Operators used in numbers in general(Arithmetic):

+ (Addition)

- (Subtraction)

* (Multiplication)

/ (Division)

% (Moduls)

Simply:

\$x=5; \$y=10;

\$a+\$b

Output : 15

Equality Operators:

== (Equal to)

!= (Not Equal to)

> (Greater)

< (Less than)

String Operators

eq

ne

gt

lt

Array

The Array subject is used to maintain the list of the group entered in Perl. Each group is given a separate name, each can be a separate type.

```
@letters = (a..f);  
@numbers = (1..6);  
  
print "@letters\n";  
print "@numbers\n";
```

```
a b c d e f  
1 2 3 4 5 6
```

List Functions

3 kinds of sorting functions are used. By the way it always starts from 0. To start, we need parentheses and commas.

join()

reverse()

sort()

join()

Separating the elements in a list with the join () function is written to put a distance.

```
@list = (a..f);  
print join(" ", @list);
```

```
a b c d e f
```

reverse()

The reverse function is a function that reverses the elements in the list. Same the strrev().

```
@list = (a..f);  
print join(" ", @list);  
@chang = reverse(@list);  
print "\n@chang";
```

```
a b c d e f  
f e d c b a
```

sort()

Using the sort () function in the Perl programming language is used to sort a list from beginning to end. For example, a list of letters is written in a complicated way, the sort () function is useful for editing.

```
@phones = ('Samsung ', ' LG ', 'Apple ', ' Nokia ');  
print sort @phones;
```

```
LG  Nokia Apple Samsung
```

String functions

There are 4 different string functions most used.

length()

lc()

uc()

index()

The `length()` perform is employed to calculate the amount of characters within the list.

```
$phones=('SamsungLGAppleNokia');  
print(length($phones), "\n");
```

19

The `lc()` perform converts the characters within the list to minuscular values.

```
$phones=('SamsungLGAppleNokia');  
print(lc($phones), "\n");
```

samsunglgapplenokia

The `uc()` function The opposite of `lc()` converts characters to uppercase letters.

The `index()` perform is employed to see its position within the list written higher than.

Class-Constructor

It is an inspiration for compatible categories that are the same as one another. The laptop is an object, but the building stages of the laptop is a class. When this process is over, we can produce as many laptops as we want.

```
package TryClass;

sub new {
    return bless {}, shift;
}

1;
```

The simplest use may be like this.

It records to the file named (name).pm, calls it.

Shift uses the task here to transfer the file @name with the bless function.