

Perl Programming မိတ်ဆက်

Ye Kyaw Thu
@Deep Learning School, Myanmar

What is Perl?

A script programming language တစ်ခု

စာကြောင်းတွေ၊ စာလုံးတွေ၊ text file တွေနဲ့ အလုပ်လုပ်တဲ့ အခါမှာ အဆင်ပြေဖို့အတွက် ရည်ရွယ်ပြီး develop စလုပ်ခဲ့

အခုအခါမှာတော့ အလုပ်မျိုးစုံအတွက် အသုံးပြုနိုင် (system administration, web development, network programming, GUI development ... General Purpose Programming)

“Practical Extraction and Report Language” ရဲ့ အတိုကောက်လို့လည်း ယူဆနိုင်

Stable ဖြစ်တယ်၊ Open Source Software, GPL လိုင်စင်၊ Linux, Mac, Windows စတဲ့ OS platform မျိုးစုံမှာ အသုံးပြုနိုင်

What is Perl?

အမေရိကန် ပရိုဂရမ်မာ Larry Wall က Perl interpreter ကို ဖန်တီးခဲ့

Perl နဲ့ ပတ်သက်တဲ့ စာအုပ်တွေကိုလည်း ရေးခဲ့

HP: <http://www.wall.org/~larry/>

Perl Version 1.0000 ကို ၁၉၈၇ ခုနှစ် မှာ Larry Wall က မိတ်ဆက်ခဲ့ (Linux မတိုင်ခင် ငှက်အလို)

ခရစ်ယန်ဘာသာကို ယုံကြည်ကိုးကွယ်ပြီး တချို့ Perl terminology တွေမှာ bible နဲ့ ပတ်သက်တာမျိုးတွေ တွေ့ရပါလိမ့်မယ် (Perl နာမည်ကိုယ်တိုင် က “pearl of great price” Matthew 13:46, Function နာမည် Bless)



What is Perl?

လေ့လာရတာ လွယ်ကူပြီး၊ Coding လုပ်ရတာလွယ်တယ်

လွတ်လွတ်လပ်လပ် ရေးလို့ရတဲ့ အတွက် တစ်ယောက်ရေးထားတာကို
နောက်တစ်ယောက်က ဖတ်လို့ မလွယ်ကူတဲ့ ပြဿနာမျိုးတွေရှိတတ်တယ်

Text ဖိုင်တွေအပြင်၊ HTML, XML နဲ့ တခြား Mark-up language တွေကို ကောင်းကောင်း
handle လုပ်ပေးနိုင်

C, C⁺⁺, Java တွေကို လေ့လာဖူးပြီးသားသူဆိုရင် Perl ကို လေ့လာရတာက
အရမ်းကိုလွယ်ကူပါတယ်

Procedural လိုလည်းရေးနိုင်၊ OOP (Object-Oriented Programming) ပုံစံမျိုးလည်း
ရေးနိုင်

Scalar Variables

Variable type ကို မသုံးခင်မှာ သတ်မှတ်ပေးစရာ မလိုဘူး

Mathematical operator တွေကို ကြည့်ပြီး Perl interpreter က ဆုံးဖြတ်လိမ့်မယ်

```
$a=10;    # interger variable
```

```
$b=3.14;  # float variable
```

```
$result=$b+12;  # result variable က float ဖြစ်လိမ့်မယ်
```

```
$hv=1D921;  # hv variable က 121121 ရဲ့ Hexadecimal value
```

```
$a++;  # $a variable ကို ၁ပေါင်းထည့်တာ
```

Scalar Variables

```
$sar="မြန်မာစာ" # string variable
```

Single quote နဲ့ double quote သုံးတဲ့အခါမှ သိထားရမဲ့အချက်က

```
$city= 'ပြင်ဦးလွင်';
```

```
print 'မြို့နာမည် $city' # မြို့နာမည် $city ဆိုပြီးပဲ ရိုက်ထုတ်ပြပေးလိမ့်မယ်
```

```
$city='ပြင်ဦးလွင်';
```

```
print "မြို့နာမည် $city" # မြို့နာမည် ပြင်ဦးလွင် ဆိုပြီး ရိုက်ထုတ်ပြပေးလိမ့်မယ်
```

Array and Hash

@ သင်္ကေတ ကို array variable ရဲ့ ရှေ့မှာသုံးတယ်

```
my @names = ("မောင်မောင်", "အောင်အောင်", "မြမြ");
```

```
my @mark = (35, 75, 100);
```

```
my @array = 'A' .. 'Z';
```

```
@1_10 = (1..10);
```

Variable ကို ခေါ်သုံးတဲ့ အခါမှာတော့ variable နာမည်ရှေ့မှာ \$တပ်ပြီးသုံးတယ်

\$names[0] က “မောင်မောင်” ဖြစ်ပြီး၊ \$names[2] က “မြမြ” ဖြစ်တယ်။

\$mark[1] က 75

Array and Hash

array ထဲကို integer, string စတဲ့ မတူတဲ့ ဒေတာတွေကို ရောသိမ်းနိုင်

```
@array1 = (1, 2, 'string');
```

ကွင်းစ၊ ကွင်းပိတ်နဲ့ မဟုတ်ပဲ “qw/ /;” ဆိုတဲ့ ပုံစံနဲ့လည်း ရေးနိုင်

```
@array2 = qw/We are learning perl programming./;
```

```
# checking last index
```

```
print "Last index: ", $#array1, "\n"; # Last index: 2 ဆိုပြီး ရိုက်ပြလိမ့်မယ်
```

```
#print out last index
```

```
print " $array1[-1]"; # string ဆိုတာကို ရိုက်ပြလိမ့်မယ်
```


Push, Unshift, Pop, Shift

```
#!/usr/bin/perl

# create an array
@holidays = ("Sat","Sun");
print "original: \@holidays  = @holidays\n";

# add one more holiday
push(@holidays, "Mon");
print "add: \@holidays  = @holidays\n";

# unshift or add one more holiday at the beginning
unshift(@holidays, "Fri");
print "unshift: \@holidays  = @holidays\n";

# pop or remove one element from the last
pop(@holidays);
print "pop: \@holidays  = @holidays\n";

# shift or remove one element from the beginning
shift(@holidays);
print "shift: \@holidays  = @holidays\n";
```

Array and Hash

% သင်္ကေတ ကို hash variable ရဲ့ ရှေ့မှာသုံးတယ်

```
my %names_data = ('Mg Mg', 35, 'Aung Aung', 75, 'Mya Mya', 40);
```

Hash ကို ပြန်ခေါ်သုံးပုံက အောက်ပါအတိုင်း

```
print "\$names_data{'Mg Mg'} = $names_data{'Mg Mg'}\n";  
print "@{[%names_data]}";
```

=====

```
$names_data{'Mg Mg'} = 35  
Mya Mya 40 Aung Aung 75 Mg Mg 35
```

Array and Hash

Perl module “**Data::Dumper**” ကို သုံးပြီးလည်း hash ကို ရိုက်ထုတ်ပြခိုင်းနိုင်

Module က installation လုပ်ထားပြီးသားဆိုရင် use Data::Dumper; ဆိုပြီး ခေါ်ယူသုံးနိုင်

```
print Dumper(\%names_data);
```

```
=====
```

```
$VAR1 = {  
    'Aung Aung' => 75,  
    'Mya Mya' => 40,  
    'Mg Mg' => 35  
};
```

Array and Hash

Hash ထဲမှာ ရှိမရှိ ကို exists ဆိုတဲ့ function နဲ့ check လုပ်နိုင်

ဥပမာ။ ။

```
%tiny_dictionary = ('one' => 1, 'two' => 2, 'three' => 3);
```

```
if( exists($tiny_dictionary{'two'}) ) {  
    print "two = $tiny_dictionary{'two'}!\n";  
} else {  
    print "I can't find \"two\"\n";  
}
```

Multiline String

Perl syntax မှာက string variable အတွက် စာကြောင်းရှည်တွေကို enter ခေါက်ပြီး ရေးသွားနိုင်

```
#!/usr/bin/perl
```

```
$string = 'This is  
a long/multiline  
string';
```

```
print "$string\n";
```

```
=====
```

Perl ရဲ့ shebang (သို့) hashbang က “**#!/usr/bin/perl**”

How to run perl

> perl program-filename

E.g.

> lar@lar-air:~/ss2018/perl/test\$ perl ./multiline-string.pl

This is
a long/multiline
string

=====

chmod +x ./multiline-string.pl လုပ်ထားရင် “./multiline-string.pl” ဆိုပြီးခေါ် run နိုင်

foreach

```
#!/usr/bin/perl
```

```
#assigning hash key and value
```

```
$prices{"Mont Hin Khar"} = 100.00;
```

```
$prices{"Khauk Swe Thout"} = 150.25;
```

```
$prices{"Rakhine Mont Di"} = 300.00;
```

```
print "print key and value with \"foreach\" :\n";
```

```
foreach $key (keys %prices)
```

```
{
```

```
    $value = $prices{$key};
```

```
    print " $key costs $value\n";
```

```
}
```

Let's see practical usage of perl

Syllable breaking program with perl:

<https://github.com/ye-kyaw-thu/sylbreak/blob/master/perl/sylbreak.pl>

Column to line with perl:

<https://github.com/ye-kyaw-thu/myPOS/blob/master/corpus-draft-ver-1.0/model/crf/col2line.pl>