# Perl基础

#!/usr/bin/perl

$hv = "Hello World!/n";

print $hv;

=================================

1. Perl 数据类型非常简单，只有3种类型：Scalar标量、Array数组和Hash哈希。

Scalar类型即标量，是单个的值，它基本上是除了Array和Hash类型的所有一切。

$myfirst = 123;　 #数字123

$mysecond = "123"; #字符串123

1. Array类型是一个数组，可以包含不同类型的元素，比如它可以同时包含整数元素和字符串元素的数组。

@arr = (1,2,3)

Hash类型与Java语言的HashMap类相似。

%h = ('a'=>1,'b'=>2);

#!/usr/bin/perl -w

my %hash;

$hash{'author'}="Young";

my %food=('fruit',"apple",'drink',"Coco");

my %fruit=(apple=>"fruit",banana=>'fruit');

print "$hash{'author'}\n";

foreach $key (keys%food)

{

print"$key=>$food{$key}\n";

}

my @key=keys%fruit;

my @value=values%fruit;

print@key;

print@value;

print"\n";

print"author is exists\n" if(exists$hash{'author'});

delete$food{'drink'};

print"after delete some keyword print \%food\n";

while((my$key,my$value)=each%food)

{

print"$key=>$value\n";

}

=================================

#claim a hash and assign some values

my %aHash;

$aHash{'a'}=0;

$aHash{'b'}=1;

$aHash{'c'}=2;

$aHash{'d'}=3;

$aHash{'e'}=4;

#put all keys to an array

my @anArray = keys (%aHash);

#loop array and output each scalar

foreach my $aScalar (@anArray){

print $aScalar."\n";

}

如果想对数组排序，可以简单的使用sort函数，如下：

foreach my $aScalar (sort @anArray){

print $aScalar."\n";

}

1. Perl语言的条件语句和循环语句有if、while、for、foreach等关键字，
2. 文件读/写

下面的例子说明了如何从文件中读取内容，以及如何向文件写入内容。

< 是读 >是写

要注意“>”和“>>”的不同，前者是创建新文件，后者是向文件添加内容。

#read from a file

my $file = "input.txt";

open(my $fh, "<", $file) or die "cannot open < $file!";

while ( my $aline = <$fh> ) {

#chomp so no new line character 删除换行符，<>表示一行一行读

chomp($aline);

print $aline;

}

close $fh;

# write to a file

my $output = "output.txt";

open (my $fhOutput, ">", $output) or die("Error: Cannot open $output file!");

print $fhOutput "something";

close $fhOutput;

1. 正则表达式

匹配：m/<regexp>/ （或 /<regexp>/ ，略去 m）

替换：s/<pattern>/<replacement>/

转化：tr/<pattern>/<replacemnt>/

这三种形式一般都和 =~ 或 !~ 搭配使用（其中 =~表示相匹配，在整条语句中读作 does，!~表示不匹配，在整条语句中读作doesn't）

$str = "I love Perl";

$str =~ m/Perl/; #表示如果在$str中发现"Perl"字符串，则返回"1"否则返回"0"。

print $str =~ m/Perl/;

$str =~ s/Perl/BASH/; #表示将变量$str中的"Perl"字符串替换为"BASH"，匹配成功返回" 1"，否则返回"0"。

print $str;

1. 值传递

sub getparameter

{

(my $var,my @arr)=@\_;

print "It's the 1 parameter:$var\n";

print "It's the 2 parameter:@arr\n";

}

my @array=("this","is","a","test");

my $variable="this is another test";

getparameter($variable,@array);

引用传递

sub getparameter

{

(my $arr1,my $arr2)=@\_;

print "It's the 1 parameter:@$arr1\n";

print "It's the 2 parameter:@$arr2\n";

@$arr2 = ("this","is","a","pig");

}

my @array1=("this","is","a","test");

my @array2=qw/this is another test/;

getparameter(\@array1,\@array2);

print @array2;

1. 其他小技巧

1）通过字符串的每个字符进行循环

$aline = "thisisafefe";

my @lineCharArray = split('',$aline);

foreach my $character (@lineCharArray){

print $character."\n";

}

2）创建一个包含26个字母的数组

my @charArray = ('a'..'z' );

my @twoCharArray = ('aa'..'zz');