

Perl Interview questions and Answers

What is Perl one-liner?

There are two ways a Perl script can be run:

a) from a command line, called one-liner, that means you type and execute immediately on the command line. You'll need the `-e` option to start like `"perl -e "print "Hello";"`. One-liner doesn't mean one Perl statement. One-liner may contain many statements in one line.

b) from a script file, called Perl program.

Assuming both a `local($var)` and a `my($var)` exist, what's the difference between `${var}` and `${"var"}`?

`${var}` is the lexical variable `$var`, and `${"var"}` is the dynamic variable `$var`.

Note that because the second is a symbol table lookup, it is disallowed under ``use strict "refs"`. The words `global`, `local`, `package`, `symbol table`, and `dynamic` all refer to the kind of variables that `local()` affects, whereas the other sort, those governed by `my()`, are variously known as `private`, `lexical`, or `scoped variable`.

Name all the prefix dereferencer in perl?

The symbol that starts all scalar variables is called a prefix dereferencer. The different types of dereferencer are.

(i) `$`-Scalar variables

(ii) %-Hash variables

(iii) @-arrays

(iv) &-subroutines

(v) Type globs-*myvar stands for @myvar, %myvar.

What exactly is grooving and shortening of the array?

You can change the number of elements in an array simply by changing the value of the last index of/in the array \$#array. In fact, if you simply refer to a non existent element in an array perl extends the array as needed, creating new elements. It also includes new elements in its array.

What are the three ways to empty an array?

The three different ways to empty an array are as follows

1. You can empty an array by setting its length to a negative number.
2. Another way of emptying an array is to assign the null list ().
3. Try to clear an array by setting it to undef, but be aware when you set to undef.

How do you work with array slices

An array slice is a section of an array that acts like a list, and you indicate what elements to put into the slice by using multiple array indexes in square brackets. By specifying the range operator you can also specify a slice.

What is meant by splicing arrays explain in context of list and scalar.

Splicing an array means adding elements from a list to that array, possibly replacing elements now in the array. In list context, the splice function returns the elements removed from the array. In scalar context, the splice function returns the last element removed.

What are the different types of perl operators?

Please refer following URL...[Perl Operators](#)

What are the two different types of data perl handles?

Perl handles two types of data they are

- (i) Scalar Variables and
- (ii) Lists

Scalar variables hold a single data item whereas lists hold multiple data items.

What are scalar variables?

Scalar variables are what many programming languages refer to as simple variables. They hold a single data item, a number, a string, or a perl reference. Scalars are called scalars to differentiate them from constructs that can hold more than one item, like arrays.

Explain about lists?

A list is a construct that associates data elements together and you can specify a list by enclosing those elements in parenthesis and separating them with commas. They could themselves be arrays, hashes or even other lists. Lists do not have a specific list data type.

Name all the prefix dereferencer in perl?

The symbol that starts all scalar variables is called a prefix dereferencer. The different types of dereferencer are.

- (i) \$-Scalar variables
- (ii) %-Hash variables
- (iii) @-arrays
- (iv) &-subroutines
- (v) Type globs-*myvar stands for @myvar, %myvar.

Explain about an ivalue?

An ivalue is an item that can serve as the target of an assignment. The term l value originally meant a "left value", which is to say a value that appears on the left. An ivalue usually represents a data space in memory and you can store data using the ivalues name. Any variable can serve as an ivalue.

How does a “grep” function perform?

Grep returns the number of lines the expression is true. Grep returns a sublist of a list for which a specific criterion is true. This function often involves pattern matching. It modifies the elements in the original list.

Explain about Typeglobs?

Type globs are another integral type in perl. A typeglob's prefix derefencer is *, which is also the wild card character because you can use typeglobs to create an alias for all types associated with a particular name. All kinds of manipulations are possible with typeglobs.

Is there any way to add two arrays together?

Of course you can add two arrays together by using push function. The push function adds a value or values to the end of an array. The push function pushes the values of list onto the end of the array. Length of an array can be increased by the length of list.

How to use the command shift?

Shift array function shifts off the first value of the array and returns it, thereby shortening the array by one element and moving everything from one place to the left. If you don't specify an array to shift, shift uses @ ARGV, the array of command line arguments passed to the script or the array named @-.

Which has the highest precedence, List or Terms? Explain?

Terms have the highest precedence in perl. Terms include variables, quotes, expressions in parenthesis etc. List operators have the same level of precedence as terms. Specifically, these operators have very strong left word precedence.

What is a short circuit operator?

The C-Style operator, ||, performs a logical (or) operation and you can use it to tie logical clauses together, returning an overall value of true if either clause is true. This operator is called a short-

circuit operator because if the left operand is true the right operand is not checked or evaluated.

What are the different forms of goto in perl? Explain?

The three forms of goto are as follows. They are

- (i) Goto label
- (ii) Goto name
- (iii) Goto expr

The first form, goto LABEL, transfers execution to the statement labeled LABEL. The second form, goto EXPR, expects EXPR to evaluate to a label. The last form goto &name is used with subroutines. This goto statement is used only when there is a necessity as it can create havoc in a program.

What are the different types of eval statements?

There are two different types of eval statements they are eval EXPR and eval BLOCK. Eval EXPR executes an expression and eval BLOCK executes BLOCK. Eval Block executes an entire block, BLOCK. First one is used when you want your code passed in the expression and the second one is used to parse the code in the block.

Explain about returning values from subroutines (functions)?

The return value of the subroutine is the value of the last expression evaluated or you can explicitly use a return statement to exit the subroutine specifying the return value. That return value is evaluated in the appropriate content depending on the content of the subroutine call.

What is sed?

sed is stream editor, a Unix tool for working with streams of text data. See the awful truth about sed.

How do you substitute strings with sed?

Use 's/old/new' command, so sed 's/hello/goodbye/' would substitute the occurrence of the word hello to goodbye.

How do you inject text with sed?

& in the substitution string defines the pattern found in the search string. As an example, here's us trying to find a word 'hello' and replacing it with 'hello and how are you':

echo 'hello there' | sed 's/^hello/& and how are you/'

Can I find several patterns and refer to them in the replacement string?

Yes, use (pattern) and then refer to your patterns as \1, \2, \3 and so on.

If the string is 'old old old' and I run 's/old/new', I get 'new old old' as the result. I need 'new new new'.

You forgot the global modifier, which would replace every occurrence of the pattern with the substitution. 's/old/new/g' will work.

But I want 'old old new' from the previous example. - Just use the numeric modifier saying you want the third occurrence to be replaced. 's/old/new/3' will work.

I wrote a rather complex sed script. How do I save and run it?

Assuming that your file is named myscript1.sed, you can invoke sed -f myscript1.sed.

How do I delete trailing whitespaces from each line?

sed 's/[\t]*\$//' Here we're replacing any occurrence of a space or a tab with nothing. Check sed one-liners for more examples.

How do you print just a few first lines of the file? - sed 1q will give you just the first line, sed 10q the first 10 lines.

How do you replace a pattern only if it's found, so that it's executed faster? - Nest the replacement statement: sed '/old/ s/old/new/g' file.txt

How do you know the reference of a variable whether it is a reference, scalar, hash or array?

There is a 'ref' function that lets you know

What is the difference between 'use' and 'require' function?

Use: 1. the method is used only for modules (only to include .pm type file) 2. the included object are verified at the time of compilation. 3. No Need to give file extension. Require: 1. The method is used for both libraries (package) and modules 2. The include objects are verified at the run time. 3. Need to give file Extension.

What is the use of 'chomp' ? what is the difference between 'chomp' and 'chop'?

Ans. 'chop' function only removes the last character completely from the scalar, whereas 'chomp' function only removes the last character if it is a newline. By default, 'chomp' only removes what is currently defined as the \$INPUT_RECORD_SEPARATOR. Whenever you call 'chomp', it checks the value of a special variable '\$/'. Whatever the value of '\$/' is eliminated from the scalar. By default the value of '\$/' is '\n'.

Print this array @arr in reversed case-insensitive order

Ans> @solution = sort {lc \$a cmp lc \$b} @arr.

What is '->' in Perl?

Ans. It is a symbolic link to link one file name to a new name. So let's say we do it like file1-> file2, if we read file1, we end up reading file2.

How do you check the return code of system call?

System calls "traditionally" return 0 when successful and 1 when it fails. `system (cmd) or die "Error in command";`

#create directory if not there

```
if (! -s "$temp/engl_2/wf"){ System "mkdir -p $temp/engl_2/wf"; } if (! -s "$temp/backup_basedir"){
system "mkdir -p $temp/backup_basedir"; } ${pack_2} = -M
"$temp/engl_2/wf/${wf_package_name}.data"; ${new_pack} = -M "${pack}/package.data";
```

What is the use of -M and -s in the above script?

Ans. -s means is filename a non-empty file -M how long since filename modified?

How to substitute a particular string in a file containing million of record?

Ans. `perl -p -i bak -e 's/search_str/replace_str/g' filename`

I have a variable named \$objref which is defined in main package. I want to make it as a

Object of class XYZ. how could I do it?

Ans. use XYZ my \$objref =XYZ -> new() OR, bless \$objref, 'XYZ';

what is meant by a 'pack' in perl?

Pack Converts a list into a binary representation. Takes an array or list of values and packs it into a binary structure, returning the string containing the structure It takes a LIST of values and converts it into a string. The string contains a concatenation of the converted values. Typically, each converted value looks like its machine-level representation. for example, on 32-bit machines a converted integer may be represented by a sequence of 4 bytes.

how to implement stack in Perl?

through push() and shift() function. push adds the element at the last of array and shift() removes from the beginning of an array.

What is Grep used for in Perl?

Grep is used with regular expression to check if a particular value exists in an array. it returns 0 if the value does not exist, 1 otherwise.

How to code in Perl to implement the tail function in unix?

And. You have to maintain a structure to store the line number and the size of the file at that time eg. 1-10bytes, 2-18bytes.. you have a counter to increase the number of lines to find out the number of lines in the file. once you are through the file, you will know the size of the file at any nth line, use 'sysseek' to move the file pointer back to that position (last 10) and then start reading till the end.

15. How do you navigate through an XML documents?

Ans. You can use the XML::DOM navigation methods to navigate through an XML::DOM node tree and use the getnodevalue to recover the data. DOM Parser is used when it is needed to do node operation. Instead we may use SAX parser if you require simple processing of the xml structure.

Why do you use Perl?

- * Perl is a powerful free interpreter.
- * Perl is portable, flexible and easy to learn.

How do I set environment variables in Perl programs?

You can just do something like this:

```
$ENV{'PATH'} = '...';
```

As you may remember, "%ENV" is a special hash in Perl that contains the value of all your environment variables.

Because %ENV is a hash, you can set environment variables just as you'd set the value of any Perl hash variable. Here's how you can set your PATH variable to make sure the following four directories are in your path::

```
$ENV{'PATH'} = '/bin:/usr/bin:/usr/local/bin:/home/yourname/bin';
```

Which of these is a difference between C++ and Perl?

Perl can have objects whose data cannot be accessed outside its class, but C++ cannot.

Perl can use closures with unreachable private data as objects, and C++ doesn't support closures. Furthermore, C++ does support pointer arithmetic via `int *ip = (int*)&object`, allowing you to look all over the object. Perl doesn't have pointer arithmetic. It also doesn't allow `#define private public` to change access rights to foreign objects. On the other hand, once you start poking around in `/dev/mem`, no one is safe.

How to open and read data files with Perl

Data files are opened in Perl using the `open()` function. When you open a data file, all you have to do is specify (a) a file handle and (b) the name of the file you want to read from.

As an example, suppose you need to read some data from a file named "checkbook.txt". Here's a simple open statement that opens the checkbook file for read access: `open (CHECKBOOK, "checkbook.txt");` In this example, the name "CHECKBOOK" is the file handle that you'll use later when reading from the checkbook.txt data file. Any time you want to read data from the checkbook

file, just use the file handle named "CHECKBOOK".

Now that we've opened the checkbook file, we'd like to be able to read what's in it. Here's how to read one line of data from the checkbook file:

```
$record = < CHECKBOOK > ;
```

After this statement is executed, the variable \$record contains the contents of the first line of the checkbook file. The "<>" symbol is called the line reading operator.

To print every record of information from the checkbook file

```
open (CHECKBOOK, "checkbook.txt") || die "couldn't open the file!";  
while ($record = < CHECKBOOK >) {  
    print $record;  
}  
close(CHECKBOOK);
```

How do I do fill_in_the_blank for each file in a directory?

Answer:Here's code that just prints a listing of every file in the current directory:

```
#!/usr/bin/perl -w  
opendir(DIR, ".");  
@files = readdir(DIR);  
closedir(DIR);  
foreach $file (@files) {  
    print "$file\n";  
}
```

How do I generate a list of all .html files in a directory?

Answer:

Here's a snippet of code that just prints a listing of every file in the current directory that ends with the extension .html:

```
#!/usr/bin/perl -w  
opendir(DIR, ".");  
@files = grep(/\..html$/,readdir(DIR));  
closedir(DIR);
```

```
foreach $file (@files) {  
    print "$file\n";  
}
```

What happens when you return a reference to a private variable?

Perl keeps track of your variables, whether dynamic or otherwise, and doesn't free things before you're done using them.

How to turn on Perl warnings? Why is that important?

Perl is very forgiving of strange and sometimes wrong code, which can mean hours spent searching for bugs and weird results. Turning on warnings helps uncover common mistakes and strange places and save a lot of debugging time in the long run. There are various ways of turning on Perl warnings:

- * For Perl one-liner, use -w option on the command line.
- * On Unix or Windows, use the -w option in the shebang line (The first # line in the script). Note: Windows Perl interpreter may not require it.
- * For other systems, choose compiler warnings, or check compiler documentation.

What are scalar data and scalar variables?

Perl has a flexible concept of data types. Scalar means a single thing, like a number or string. So the Java concept of int, float, double and string equals to Perl's scalar in concept and the numbers and strings are exchangeable. Scalar variable is a Perl variable that is used to store scalar data. It uses a dollar sign \$ and followed by one or more alphanumeric characters or underscores. It is case sensitive.

Why should I use the -w argument with my Perl programs?

Many Perl developers use the -w option of the interpreter, especially during the development stages of an application. This warning option turns on many warning messages that can help you understand and debug your applications.

To use this option on Unix systems, just include it on the first line of the program, like this:

```
#!/usr/bin/perl -w
```

If you develop Perl apps on a DOS/Windows computer, and you're creating a program named myApp.pl, you can turn on the warning messages when you run your program like this:

```
perl -w myApp.pl
```

I want users send data by formmail but when they send nothing or call it from web site they will see error. codes in PHP like this: if (isset(\$HTTP_POST_VARS)){ } else{ echo ("error lalalalal") } How it will look in perl?

In php it will be like

```
if (isset($HTTP_POST_VARS)){
```

```
....
```

```
}
```

In perl, tried this.

```
if ($ENV{'REQUEST_METHOD'} eq 'POST'){
```

```
.....
```

```
}
```

What is the output of the following Perl program?

```
1 $p1 = "prog1.java";
```

```
2 $p1 =~ s/(.*)\.java/$1.cpp/;
```

```
3 print "$p1\n";
```

Answer:What is the output of the following Perl program?

```
1 $p1 = "prog1.java";
```

```
2 $p1 =~ s/(.*)\.java/$1.cpp/;
```

```
3 print "$p1\n";
```

```
prog1.cpp
```

Why are not Perls patterns regular expressions?

Because Perl patterns have backreferences.

A regular expression by definition must be able to determine the next state in the finite automaton without requiring any extra memory to keep around previous state. A pattern `/([ab]+)c\1/` requires the state machine to remember old states, and thus disqualifies such patterns as being regular expressions in the classic sense of the term.

What does Perl do if you try to exploit the `execve(2)` race involving `setuid` scripts?

Sends mail to root and exits.

It has been said that all programs advance to the point of being able to automatically read mail.

While not quite at that point (well, without having a module loaded), Perl does at least automatically send it.

How do I do < fill-in-the-blank > for each element in a hash?

Here's a simple technique to process each element in a hash:

```
#!/usr/bin/perl -w
```

```
%days = (  
'Sun' => 'Sunday',  
'Mon' => 'Monday',  
'Tue' => 'Tuesday',  
'Wed' => 'Wednesday',  
'Thu' => 'Thursday',  
'Fri' => 'Friday',  
'Sat' => 'Saturday' );
```

```
foreach $key (sort keys %days) {  
    print "The long name for $key is $days{$key}.\n";  
}
```

How do I sort a hash by the hash key?

Suppose we have a class of five students.

Their names are kim, al, rocky, chrisy, and jane.

Here's a test program that prints the contents of the grades hash, sorted by student name:

```
#!/usr/bin/perl -w

%grades = (
    kim => 96,
    al => 63,
    rocky => 87,
    chrisy => 96,
    jane => 79,
);

print "\n\tGRADES SORTED BY STUDENT NAME:\n";
foreach $key (sort (keys(%grades))) {
    print "\t\t$key \t\t$grades{$key}\n";
}
```

The output of this program looks like this:

GRADES SORTED BY STUDENT NAME:

```
al 63
chrisy 96
jane 79
kim 96
rocky 87

}
```

How do you print out the next line from a filehandle with all its bytes reversed?

```
print scalar reverse scalar <FH>
```

Surprisingly enough, you have to put both the reverse and the <FH> into scalar context separately for this to work.

How do I send e-mail from a Perl/CGI program on a Unix system?

Sending e-mail from a Perl/CGI program on a Unix computer system is usually pretty simple. Most Perl programs directly invoke the Unix sendmail program. We'll go through a quick example here. Assuming that you've already have e-mail information you need, such as the send-to address and subject, you can use these next steps to generate and send the e-mail message:

```
# the rest of your program is up here ...  
open(MAIL, "|/usr/lib/sendmail -t");  
print MAIL "To: $sendToAddress\n";  
print MAIL "From: $myEmailAddress\n";  
print MAIL "Subject: $subject\n";  
print MAIL "This is the message body.\n";  
print MAIL "Put your message here in the body.\n";  
close (MAIL);
```

How to read from a pipeline with Perl

Example 1:

To run the date command from a Perl program, and read the output of the command, all you need are a few lines of code like this:

```
open(DATE, "date|");  
$theDate = <DATE>;  
close(DATE);
```

The open() function runs the external date command, then opens a file handle DATE to the output of the date command.

Next, the output of the date command is read into the variable \$theDate through the file handle DATE.

Example 2:

The following code runs the "ps -f" command, and reads the output:

```

open(PS_F, "ps -f|");
while (<PS_F) {
($uid,$pid,$ppid,$restOfLine) = split;
# do whatever I want with the variables here ...
}
close(PS_F);

```

Why is it hard to call this function: sub y { "because" }

Because y is a kind of quoting operator.

The y/// operator is the sed-savvy synonym for tr///. That means y(3) would be like tr(), which would be looking for a second string, as in tr/a-z/A-Z/, tr(a-z)(A-Z), or tr[a-z][A-Z].

What does \$result = f() .. g() really return?

False so long as f() returns false, after which it returns true until g() returns true, and then starts the cycle again.

This is scalar not list context, so we have the bistable flip-flop range operator famous in parsing of mail messages, as in ``$in_body = /^$/ .. eof()'`. Except for the first time f() returns true, g() is entirely ignored, and f() will be ignored while g() later when g() is evaluated. Double dot is the inclusive range operator, f() and g() will both be evaluated on the same record. If you don't want that to happen, the exclusive range operator, triple dots, can be used instead. For extra credit, describe this:

```
$bingo = ( a() .. b() ) ... ( c() .. d() );
```

Why does Perl not have overloaded functions?

Because you can inspect the argument count, return context, and object types all by yourself.

In Perl, the number of arguments is trivially available to a function via the scalar sense of `@_`, the return context via `wantarray()`, and the types of the arguments via `ref()` if they're references and

simple pattern matching like `/^\d+$/` otherwise. In languages like C++ where you can't do this, you simply must resort to overloading of functions.

What does `read()` return at end of file?

0

A defined (but false) 0 value is the proper indication of the end of file for `read()` and `sysread()`.

What does `new $cur->{LINK}` do? (Assume the current package has no `new()` function of its own.)

```
$cur->new()->{LINK}
```

The indirect object syntax only has a single token lookahead. That means if `new()` is a method, it only grabs the very next token, not the entire following expression.

This is why ``new $obj[23] arg'` doesn't work, as well as why ``print $fh[23] "stuff\n"` doesn't work.

Mixing notations between the OO and IO notations is perilous. If you always use arrow syntax for method calls, and nothing else, you'll not be surprised.

How do I sort a hash by the hash value?

Here's a program that prints the contents of the grades hash, sorted numerically by the hash value:

```
#!/usr/bin/perl -w
```

```
# Help sort a hash by the hash 'value', not the 'key'.  
to highest).
```

```
sub hashValueAscendingNum {  
    $grades{$a} <=> $grades{$b};  
}
```

```
# Help sort a hash by the hash 'value', not the 'key'.
```

```
# Values are returned in descending numeric order
```

(highest to lowest).

```
sub hashValueDescendingNum {
    $grades{$b} <=> $grades{$a};
}
```

```
%grades = (
    student1 => 90,
    student2 => 75,
    student3 => 96,
    student4 => 55,
    student5 => 76,
);
```

```
print "\n\tGRADES IN ASCENDING NUMERIC ORDER:\n";
foreach $key (sort hashValueAscendingNum (keys(%grades))) {
    print "\t\t$grades{$key} \t\t $key\n";
}
```

```
print "\n\tGRADES IN DESCENDING NUMERIC ORDER:\n";
foreach $key (sort hashValueDescendingNum (keys(%grades))) {
    print "\t\t$grades{$key} \t\t $key\n";
}
```

How to read file into hash array ?

```
open(IN, "<name_file")
or die "Couldn't open file for processing: $!";
while (<IN>) {
    chomp;
    $hash_table{$_} = 0;
}
close IN;

print "$_ = $hash_table{$_}\n" foreach keys %hash_table;
```

How do you find the length of an array?

```
$@array
```

What value is returned by a lone return; statement?

The undefined value in scalar context, and the empty list value () in list context.

This way functions that wish to return failure can just use a simple return without worrying about the context in which they were called.

What is the difference between /^Foo/s and /^Foo/?

The second would match Foo other than at the start of the record if \$* were set.

The deprecated \$* flag does double duty, filling the roles of both /s and /m. By using /s, you suppress any settings of that spooky variable, and force your carets and dollars to match only at the ends of the string and not at ends of line as well -- just as they would if \$* weren't set at all.

Does Perl have reference type?

Yes. Perl can make a scalar or hash type reference by using backslash operator.

For example

```
$str = "here we go"; # a scalar variable
```

```
$strref = \$str; # a reference to a scalar
```

```
@array = (1..10); # an array
```

```
$arrayref = \@array; # a reference to an array
```

Note that the reference itself is a scalar.

How to dereference a reference?

There are a number of ways to dereference a reference.

Using two dollar signs to dereference a scalar.

```
$original = $$strref;
```

Using @ sign to dereference an array.

```
@list = @$arrayref;
```

Similar for hashes.

What does `length(%HASH)` produce if you have thirty-seven random keys in a newly created hash?

5

`length()` is a built-in prototyped as `sub length($)`, and a scalar prototype silently changes aggregates into radically different forms. The scalar sense of a hash is false (0) if it's empty, otherwise it's a string representing the fullness of the buckets, like "18/32" or "39/64". The length of that string is likely to be 5. Likewise, `'length(@a)'` would be 2 if there were 37 elements in `@a`.

If `EXPR` is an arbitrary expression, what is the difference between `$Foo::{EXPR}` and `{"Foo::".EXPR}`?

The second is disallowed under ``use strict "refs"`.

Dereferencing a string with `*{"STR"}` is disallowed under the `refs` stricture, although `*{STR}` would not be. This is similar in spirit to the way `${STR}` is always the symbol table variable, while `$_{STR}` may be the lexical variable. If it's not a bareword, you're playing with the symbol table in a particular dynamic fashion.

How do I do < fill-in-the-blank > for each element in an array?

```
#!/usr/bin/perl -w

@homeRunHitters = ('McGwire', 'Sosa', 'Maris', 'Ruth');

foreach (@homeRunHitters) {
    print "$_ hit a lot of home runs in one year\n";
}
```

How do I replace every <TAB> character in a file with a comma?

```
perl -pi.bak -e 's/\t/,/g' myfile.txt
```

What is the easiest way to download the contents of a URL with Perl?

Answer:

Once you have the libwww-perl library, LWP.pm installed, the code is this:

```
#!/usr/bin/perl

use LWP::Simple;

$url = get 'http://www.websitename.com/';
```

How to concatenate strings with Perl?

Method #1 - using Perl's dot operator:

```
$name = 'checkboxbook';

$filename = "/tmp/" . $name . ".tmp";
```

Method #2 - using Perl's join function

```
$name = "checkboxbook";

$filename = join "", "/tmp/", $name, ".tmp";
```

Method #3 - usual way of concatenating strings

```
$filename = "/tmp/${name}.tmp";
```

How do I read command-line arguments with Perl?

With Perl, command-line arguments are stored in the array named @ARGV.

\$ARGV[0] contains the first argument, \$ARGV[1] contains the second argument, etc.

\$#ARGV is the subscript of the last element of the @ARGV array, so the number of arguments on the command line is \$#ARGV + 1.

Here's a simple program:

```
#!/usr/bin/perl

$numArgs = $#ARGV + 1;

print "thanks, you gave me $numArgs command-line arguments.\n";

foreach $argnum (0 .. $#ARGV) {
    print "$ARGV[$argnum]\n";
}
```

When would local \$_ in a function ruin your day?

When your caller was in the middle for a while(m//g) loop

The /g state on a global variable is not protected by running local on it. That'll teach you to stop using locals. Too bad \$_ can't be the target of a my() -- yet.

What happens to objects lost in "unreachable" memory, such as the object returned by Ob->new() in '{ my \$ap; \$ap = [Ob->new(), \ \$ap]; }' ?

Their destructors are called when that interpreter thread shuts down.

When the interpreter exits, it first does an exhaustive search looking for anything that it allocated. This allows Perl to be used in embedded and multithreaded applications safely, and furthermore guarantees correctness of object code.

Assume that \$ref refers to a scalar, an array, a hash or to some nested data structure.

Explain the following statements:

Answer:

`$$ref; # returns a scalar`

`$$ref[0]; # returns the first element of that array`

`$ref->[0]; # returns the first element of that array`

`@$ref; # returns the contents of that array, or number of elements, in scalar context`

`$&$ref; # returns the last index in that array`

`$ref->[0][5]; # returns the sixth element in the first row`

`@{$ref->{key}} # returns the contents of the array that is the value of the key "key"`

How do you match one letter in the current locale?

Answer:

`/[^\W_\d]/`

We don't have full POSIX regexps, so you can't get at the isalpha() <ctype.h> macro save indirectly. You ask for one byte which is neither a non-alphanumeric, nor an under, nor a numeric. That leaves just the alphas, which is what you want.

How do I print the entire contents of an array with Perl?

Answer:

To answer this question, we first need a sample array. Let's assume that you have an array that contains the name of baseball teams, like this:

```
@teams = ('cubs', 'reds', 'yankees', 'dodgers');
```

If you just want to print the array with the array members separated by blank spaces, you can just print the array like this:

```
@teams = ('cubs', 'reds', 'yankees', 'dodgers');  
print "@teams\n";
```

But that's not usually the case. More often, you want each element printed on a separate line. To achieve this, you can use this code:

```
@teams = ('cubs', 'reds', 'yankees', 'dodgers');  
foreach (@teams) {  
    print "$_ \n";  
}
```

How many ways can we express string in Perl?

Many. For example 'this is a string' can be expressed in:

"this is a string"

qq/this is a string like double-quoted string/

qq^this is a string like double-quoted string^

q/this is a string/

q&this is a string&

q(this is a string)

How do you give functions private variables that retain their values between calls?

Create a scope surrounding that sub that contains lexicals.

Only lexical variables are truly private, and they will persist even when their block exits if something still cares about them. Thus:

```
{ my $i = 0; sub next_i { $i++ } sub last_i { --$i } }
```

creates two functions that share a private variable. The \$i variable will not be deallocated when its block goes away because next_i and last_i need to be able to access it.

What is the Use of Symbolic Reference in PERL?

How to find type of variable?

What is Grep used for in Perl?

grep (EXPR, LIST) extract any elements from LIST for which EXPR is TRUE.

Return Value

In scalar context - Number of times the expression returned true

In list context - List of elements that matched the expression

Example

Try out following example:

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

```
@list = (1,"Test", 0, "foo", 20 );
```

```
@has_digit = grep ( /\d/, @list );
```

```
print "@has_digit\n";
```

It produces following result

```
1 0 20
```

What is eval in perl?

Evaluates EXPR at execution time as if EXPR were a separate Perl script. This allows you to use a

separate, perhaps user-supplied, piece of Perl script within your program. An eval EXPR statement is evaluated separately each time the function is called.

The second form evaluates BLOCK when the rest of the script is parsed (before execution).

How to substitute a particular string in a file containing millions of records?

```
$_ = s/ string1 / string2 /g
```

How we can navigate the XML documents?

What is the difference between die and exit in perl?

Die is explained below;

exit Evaluates EXPR, exits the Perl interpreter, and returns the value as the exit value. Always runs all END{} blocks defined in the script (and imported packages) before exiting. If EXPR is omitted, then the interpreter exits with a value of 0. Should not be used to exit from a subroutine; either use eval and die or use return.

Following are the usage...

```
$ans = <STDIN>;  
exit 0 if $ans =~ /^[Xx]/
```

What is hash in perl?

A hash represents a set of key/value pairs. Actually hash are type of arrays with the exception that hash index could be a number or string. They are prefixed by % sign as follows:

```
my %fruit_color = ("apple", "red", "banana", "yellow");
```

what is meant 'die' in a perl program?

Raise an exception or bail out.

Following are the usage...

```
die "Can't cd to spool: $!\n" unless chdir '/usr/spool/news';
```

or

```
chdir '/usr/spool/news' or die "Can't cd to spool: $!\n"
```

or

```
open( FILEH, ">/tmp/test" ) || die( "Can't open file test $!\n" );
```

what's is the use of 'require' and what does this mean?

Load in external functions from a library at runtime

what does this mean '\$^O'? tell briefly plse..

Contains the name of the operating system that the current Perl binary was compiled for.

what does this mean : '\$_' ?

\$_ is a default special variable in perl which store the inputs for further pattern searching.

what are regular expressions?

A regular expression, also referred to as regex or regexp, provides a concise and flexible means for matching strings of text, such as particular characters, words, or patterns of characters. A regular expression is written in a formal language that can be interpreted by a regular expression processor, a program that either serves as a parser generator or examines text and identifies parts that match the provided specification.

Difference between for and foreach?

Using for you can initialize variable, condition check and increment the variables like 'C' or like some other languages.

foreach is a function having in perl using you can iterates over the list and you cannot initialize or condition check or increment like 'for' in a single line.

We can use both for or foreach for any scenario but Perl author suggest that you can use for where

you are counting a number of iteration like conventional for loops. foreach for list based operation where you are access the list from array.

Difference between exec and system in perl?

<http://www.scmgalaxy.com/forum/perl-script/t-521.html#522>

What is use strict. Explain with example.

strict - Perl pragma to restrict unsafe constructs.

Example:

```
use strict;  
use strict "vars";  
use strict "refs";  
use strict "subs";  
use strict;  
no strict "vars";
```

```
use strict "subs";
```

The use function allows you to use modules located in the standard library. When the strict pragma takes subs as an argument, it will catch any barewords found in the program while it is being internally compiled. If a bareword is found, the program will be aborted with an error message.

```
use strict "vars"
```

This generates a compile-time error if you access a variable that was neither explicitly declared (using any of my, our, state, or use vars) nor fully qualified.

```
use strict "refs"
```

This generates a runtime error if you use symbolic references.

More –

<http://perldoc.perl.org/strict.html>

<http://szabgab.com/barewords-in-perl.html>

What is CPAN?

CPAN is the Comprehensive Perl Archive Network, a large collection of Perl software and documentation. You can begin exploring from either <http://www.cpan.org/> or any of the mirrors listed at <http://www.cpan.org/SITES.html>.

CPAN is also the name of a Perl module, CPAN.pm, which is used to download and install Perl software from the CPAN archive. you may find the documentation for it by using perldoc CPAN via the command line

More: -

http://www.cpan.org/misc/cpan-faq.html#What_is_CPAN

How to install modules using CPAN or PPM (Perl Package Manager)

Install cpanm to make installing other modules easier

> cpan App::cpanminus (Mac OS X, Win32, X Windows/Linux)

Now install any module you can find in <http://www.cpan.org/modules/index.html>

cpanm Module::Name

such as to install local::lib

> cpanm local::lib

ActiveState maintains repositories of precompiled CPAN packages that can be installed with either the PPM command-line client, or the PPM GUI application. To install the DBD-mysql database driver all you have to do is run the ppm install command:

> ppm install DBD-mysql

Alternatively you can run ppm without any parameters to invoke the GUI, search for DBD-mysql, select the DBD-mysql package, and perform the installation:

> ppm

How do you open a file for writing?

open(FH, "> \$path") or die \$!;

Difference between printf and sprintf in perl ? || print vs sprintf || Difference between print

vs sprintf

The printf and sprintf is built-in function. One difference is printf prints it's output and sprintf returns it's value.

Prints the value of LIST interpreted via the format specified by FORMAT to the current output filehandle, or to the one specified by FILEHANDLE.

Syntax

```
printf FILEHANDLE FORMAT, LIST
```

```
printf FORMAT, LIST
```

Return Value

0 on failure

1 on success

Effectively equivalent to `print FILEHANDLE sprintf(FORMAT, LIST)`

Try out following example:

```
#!/usr/bin/perl -w
```

```
printf "%d\n", 3.1415126;
```

```
printf "The cost is \${%6.2f}\n",499;
```

```
printf "Perl's version is v%vd\n",%^V;
```

```
printf "%04d\n", 20;
```

It will produce following results: Try more options yourself.

3

The cost is \$499.00

Perl's version is v

0020

On the other hand, The sprintf function uses FORMAT to return a formatted string based on the values in LIST. Essentially identical to printf, but the formatted string is returned instead of being printed.

Syntax

`sprintf` FORMAT, LIST

Return Value

SCALAR, a formatted text string.

Example

Try out following example:

```
#!/usr/bin/perl -w

$text = sprintf("%0d \n", 9);
print "Formatted string $text\n";
```

It will produce following result:

Formatted string 9

What arguments do you frequently use for the Perl interpreter and what do they mean?

- w to show warning
- d to debug
- c to compile only not run
- e Which executes

You can check to see if your script has compiled successfully by using the `-c` switch. This used to check for syntax errors.

```
perl -c scriptname.pl
```

To Run on command line:executes the program given as an argument

```
perl -e 'print "hello world\n";' unix/linux
```

With Warning Pragma

```
Perl -w <scriptname>
```

To start perl debugger on the file name specified as an argument

```
Perl -d <scriptname>
```

What do the symbols \$ @ and % mean when prefixing a variable?

\$ - To Define Scalar Variable

@ - To Define an Array

% - To define a hash

What elements of the Perl language could you use to structure your code to allow for maximum re-use and maximum readability?

It could be Subroutine, Module, Class, Package or objects. Questions should be more clear.

Why do you program in Perl?

Practice Report and Extraction Language

Interpreted Language optimized for scanning arbitrary test files, extracting information from them, and printing based on that information.

Very Powerful String handling features

Available for all platforms

Speed of Development

You can enter the program in a text file and just run it. It is an interpretive language; no compiler is needed

It is Powerful

The Regular expression of Perl are extremely powerful

Uses sophisticated pattern matching techniques to scan large amounts of data quickly.

Portability

Perl is a standard language and is available on all platforms.

Free version are available on the internet

Editing Perl programs

No sophisticated editing tool is needed

Any simple text editor like Notepad or vi will do

Flexibility

Perl Does not limit the size of your data

If Memory is available, perl can handle the whole file as a single string.

Allow one to write simple programs to perform complex task

What's your favorite module and why?

```
use DBI;  
use Getopt::Long;  
use Time::Local;  
use List::Compare;  
use Log::Log4perl;  
use CGI;  
use HTML::Template;  
use Data::Dumper;  
use LWP;  
use File::Find;
```

What is a hash?

A hash can be initialized with a list, where elements of the list are key and value pairs:

```
my %food_color = ( "Apple" => "red", "Banana" => "yellow", "Lemon" => "yellow");
```

```
%age = ( "Nat", 24, "Jules", 25, "Josh", 17 );
```

is equivalent to

```
$age{"Nat"} = 24; $age{"Jules"} = 25; $age{"Josh"} = 17;
```

```
$hash{"other"}
```

Only use the % when referring to the hash as a whole such as %boss

Hashes is that their elements are stored in an internal order convenient for efficient retrieval. it will come out in an unpredictable disorder.

When Referring to the value associated with a particular key, that's a single scalar value, so is \$ is called for.

When referring to one element of an array, you also use a \$.

You can store only scalar value in a hash. References, however, are scalars. This solves the problem of storing multiple values for one key by making \$hash{\$key} a reference to an array

containing values of \$key.

What purpose does each of the following serve: -w?

Perl -w <scriptname>

Or

#!/usr/bin/perl -w

Or

use warnings; perl version 5.6 and later

The -w switch prints warnings about ambiguous identifiers, such as variables that have been used only once, improper conversion of strings and numbers etc

What is the difference between 'use' and 'require' function?

Use:

1. the method is used only for modules (only to include .pm type file)
2. the included object are verified at the time of compilation.
3. No Need to give file extension.

Require:

1. The method is used for both libraries (package) and modules
2. The include objects are verified at the run time.
3. Need to give file Extension.

Explain the difference between my and local.

Both of them are used to declare local variables.

The variables declared with "my" can live only within the block it was defined and cannot get its visibility inherited functions called within that block, but one defined with "local" can live within the block and have its visibility in the functions called within that block.

In the example below, \$::a refers to \$a in the 'global' namespace.

'local' temporarily changes the value of the variable, but only within the scope it exists in.

```
$a = 3.14159;
```

```
{
```

```
    local $a = 3;

    print "In block, \$a = $a\n";

    print "In block, \$::a = \$::a\n";

}
```

```
print "Outside block, \$a = $a\n";

print "Outside block, \$::a = \$::a\n";
```

This outputs In block,

This outputs

```
In block, $a = 3
In block, $::a = 3
Outside block, $a = 3.14159
Outside block, $::a = 3.14159
```

'my' has no effect on the global \$a, even inside the block.

```
$a = 3.14159;

{
my $a = 3;

print "In block, \$a = $a\n";

print "In block, \$::a = \$::a\n";

}

print "Outside block, \$a = $a\n";

print "Outside block, \$::a = \$::a\n";
```

This outputs

```
In block, $a = 3
In block, $::a = 3.14159
Outside block, $a = 3.14159
Outside block, $::a = 3.14159
```

What is the difference between for & foreach?

What is the difference between exec & system?

What is the difference between map & grep?

What is the difference between sysread & read?

What is the difference between chop & chomp?

What is the difference between qq,qw & qx?

What is the difference between rand & srand?

What is the difference between exists and defined?

What is the difference between use and require?

What is the difference between eval & study?