

Questions for Beginning Perl

| | |
|---|---|
| Category: Intro - (6 questions) | |
| How do you tell Linux operating system to use Perl when running a script | the top line of the script must start with #! #! path/perl |
| What character is at the end of many Perl statements | semicolon - ; |
| Command to read the Perl documentation | perldoc or man on correctly configured systems |
| How can you run Perl with warnings enabled | -w perl -w program_name |
| Perl: what pragma turns on warnings in scripts with version 5.6 or higher | use warnings; |
| What Perl pragma gives longer explanations of warning messages | use diagnostics; |
| Category: Scalars - (23 questions) | |
| How are all numbers stored internally by Perl | double-precision floating-point values |
| Perl: what character can be used to make large numbers easier to read | underscore character which is ignored 1_320_531 |
| Perl syntax to create octal, hexadecimal and binary based numbers | octal - start with leading zero 0 hexadecimal - start with leading 0x binary - start with leading 0b |
| Perl: What operator is used to get the remainder of a division operations | modulus operator - % |
| Perl operator is used to raise a number to a power | ** |
| Perl: what is different between single quote (') strings and double quote (") strings | single quotes are used as typed in double quotes perform variable substitution and have escape characters such as newline \n |
| How do you insert an octal or hexadecimal value in a double quoted string | octal - \0.. hexadecimal - \x... |
| Perl syntax for concatenate strings together | with the period operator (.) "my new" . "sting" |
| Perl: what is the string repetition operator | "string" x num |
| Perl: how is a scalar variable declared | names starts with \$ \$my_var |

| | |
|---|--|
| Perl: shortcut for assigning variable to itself like: <code>\$x = \$x + 10</code> | <code>\$x += 10</code> this works for +, -, *, / |
| Perl statement to print information to standard output | <code>print "something";</code> |
| Perl: How do you insert variables into a double quoted string | just put them in "I like to eat <code>\$my_favorite_food</code> " |
| Perl: how do you put a real dollar sign into double quoted string | escape with a backslash "my <code>\\$100</code> " |
| Perl: Comparison operator for strings | <code>eq</code> - equal, <code>ne</code> not equal <code>\$string1 eq \$string2</code> |
| Perl equality comparison operator for numbers | <code>==</code> |
| Perl: syntax for if / else statements | <code>if (condition) {.....statements} else {.....statements}</code> |
| Perl: what is considered true and false in numbers | 0 is false every other number is true |
| Perl syntax to grab a line from keyboard | <code>\$my_line = <STDIN></code> |
| Perl function to remove newline from end of a string | <code>chomp(my_string)</code> |
| Perl: syntax to loop until condition is false | <code>until (condition) {.....code}</code> |
| Perl: what value do variables hold until defined | <code>undef</code> |
| Perl: function tests to see if a variable has been assigned | <code>defined</code> |
| Category: Lists & Arrays - (21 questions) | |
| Perl: how do you access an element in an array | <code>use []\$x[3];</code> |
| Perl: What happens when you access an element that is beyond the end of an array | you get <code>undef</code> value |
| Perl: shortcut to grab the index of the last element of any array | <code>\$#var_name</code> |
| Perl: what is the effect of using negative number for an array index | it starts counting from the end of the array backwards |
| What is the Perl syntax for a list literal | <code>(value1, value2,)</code> |
| Perl: Syntax to create a list of numbers from 1 to 1000 | <code>(1..1000)..</code> is the range operator |

| | |
|---|---|
| Perl syntax for a easy way to enter a list of strings without having to surround each entry with quotes | <code>qw(string1 string2)</code> |
| Perl: when using qw (quote word) shortcut what other symbols can be used as the delimiter besides () | <code>anyqw# string1 string2 #qw! string1 string2 !</code> |
| Perl: when using qw how do you include the characters in strings that you are using for qw delimiter | <code>escape them with backslashqw! string\!1 string\!2!</code> |
| Perl: how do you assign a list of values to a list of variables | <code>(\$var1, \$var2, ...) = (value1, value2)</code> |
| Perl: syntax to refer to an entire array instead of one value in the array | <code>prefix with @ @list</code> |
| Perl operators to add new item to end of array and remove end item from array | <code>push(@array, new_value)pop(@array)</code> |
| Perl: operators to add and remove items from the beginning of array | <code>unshift(@array, new_item)shift(@array)</code> |
| Perl: what happens when you interpolate an array in a string "My string with @array in it" | <code>It prints out the individual values of the the array automatically separated by spaces</code> |
| Perl: syntax for looping through the values in a list | <code>foreach \$var @array {</code> |
| Perl: What variable receives the value of a loop iteration if no control variable is givenforeach (1..10) | <code>\$_</code> |
| Perl: operator that returns the values of an array in reverse order | <code>reverse(@array)</code> |
| Perl: operator that returns a sorted list from an array | <code>sort(@array)</code> |
| Perl: operator to force a list into scalar context | <code>scalar @array</code> |
| Perl: how can you read <STDIN> from current position to end of file | <code>assign it to a list@lines = <STDIN>;</code> |
| Standard Perl way of reading in all the remaining lines of input and removing newline characters | <code>chomp(@lines = <STDIN>);</code> |