

FAQ 4.22 How do I expand function calls in a string?

Source: <http://coding.derkeiler.com/Archive/Perl/comp.lang.perl.misc/2005-08/msg01351.html>

- *From:* PerlFAQ Server <comdog@xxxxxxxxxx>
 - *Date:* Mon, 22 Aug 2005 04:03:01 +0000 (UTC)
-

This message is one of several periodic postings to comp.lang.perl.misc intended to make it easier for perl programmers to find answers to common questions. The core of this message represents an excerpt from the documentation provided with Perl.

4.22: How do I expand function calls in a string?

(contributed by brian d foy)

This is documented in perlref, and although it's not the easiest thing to read, it does work. In each of these examples, we call the function inside the braces of used to dereference a reference. If we have a more than one return value, we can construct and dereference an anonymous array. In this case, we call the function in list context.

```
print "The time values are @ { [localtime] }.\n";
```

If we want to call the function in scalar context, we have to do a bit more work. We can really have any code we like inside the braces, so we simply have to end with the scalar reference, although how you do that is up to you, and you can use code inside the braces.

```
print "The time is ${\(scalar localtime)}.\n"
```

```
print "The time is ${ my $x = localtime; \$x }.\n";
```

If your function already returns a reference, you don't need to create the reference yourself.

```
sub timestamp { my $t = localtime; $t }
```

```
print "The time is ${ timestamp() }.\n";
```

In most cases, it is probably easier to simply use string concatenation, which also forces scalar context.

FAQ 4.22 How do I expand function calls in a string?

```
print "The time is " . localtime . ".\n";
```

Documents such as this have been called "Answers to Frequently Asked Questions" or FAQ for short. They represent an important part of the Usenet tradition. They serve to reduce the volume of redundant traffic on a news group by providing quality answers to questions that keep coming up.

If you are some how irritated by seeing these postings you are free to ignore them or add the sender to your killfile. If you find errors or other problems with these postings please send corrections or comments to the posting email address or to the maintainers as directed in the perlfq manual page.

Note that the FAQ text posted by this server may have been modified from that distributed in the stable Perl release. It may have been edited to reflect the additions, changes and corrections provided by respondents, reviewers, and critics to previous postings of these FAQ. Complete text of these FAQ are available on request.

The perlfq manual page contains the following copyright notice.

AUTHOR AND COPYRIGHT

Copyright (c) 1997–2002 Tom Christiansen and Nathan Torkington, and other contributors as noted. All rights reserved.

This posting is provided in the hope that it will be useful but does not represent a commitment or contract of any kind on the part of the contributors, authors or their agents.

.

- Prev by Date: [*Re: Use of uninitialized value Error*](#)
- Next by Date: [*Re: Using Crypt::DSA*](#)
- Previous by thread: [*Using Crypt::DSA*](#)
- Next by thread: [*Pater matching and backreference*](#)
- Index(es):
 - ◆ [*Date*](#)
 - ◆ [*Thread*](#)