

**NAME**

experimental - Experimental features made easy

**VERSION**

version 0.007

**SYNOPSIS**

```
use experimental 'lexical_subs', 'smartmatch';
my sub foo { $_[0] 1 }
```

**DESCRIPTION**

This pragma provides an easy and convenient way to enable or disable experimental features.

Every version of perl has some number of features present but considered “experimental.” For much of the life of Perl 5, this was only a designation found in the documentation. Starting in Perl v5.10.0, and more aggressively in v5.18.0, experimental features were placed behind pragmata used to enable the feature and disable associated warnings.

The `experimental` pragma exists to combine the required incantations into a single interface stable across releases of perl. For every experimental feature, this should enable the feature and silence warnings for the enclosing lexical scope:

```
use experimental 'feature-name';
```

To disable the feature and, if applicable, re-enable any warnings, use:

```
no experimental 'feature-name';
```

The supported features, documented further below, are:

```
array_base - allow the use of $[ to change the starting index of @array
autoderef - allow push, each, keys, and other built-ins on references
lexical_topic - allow the use of lexical $_ via "my $_"
postderef - allow the use of postfix dereferencing expressions, including
in interpolating strings
regex_sets - allow extended bracketed character classes in regexps
signatures - allow subroutine signatures (for named arguments)
smartmatch - allow the use of , given, and when
```

**Disclaimer**

Because of the nature of the features it enables, forward compatibility can not be guaranteed in any way.

**AUTHOR**

Leon Timmermans <leont@cpan.org>

**COPYRIGHT AND LICENSE**

This software is copyright (c) 2013 by Leon Timmermans.

This is free software; you can redistribute it and/or modify it under the same terms as the Perl 5 programming language system itself.