```
{
    no warnings 'redefine';
    eval "sub name { ... }";
}
```

Substitution loop

(P) The substitution was looping infinitely. (Obviously, a substitution shouldn't iterate more times than there are characters of input, which is what happened.) See the discussion of substitution in Quote and Quote-like Operators in *perlop*.

Substitution pattern not terminated

(F) The lexer couldn't find the interior delimiter of an s/// or s{}{} construct. Remember that bracketing delimiters count nesting level. Missing the leading $ from variable $s may cause this error.

Substitution replacement not terminated

(F) The lexer couldn't find the final delimiter of an s/// or s{}{} construct. Remember that bracketing delimiters count nesting level. Missing the leading $ from variable $s may cause this error.

substr outside of string

(W substr),(F) You tried to reference a substr() that pointed outside of a string. That is, the absolute value of the offset was larger than the length of the string. See substr in *perlfunc*. This warning is fatal if substr is used in an lvalue context (as the left hand side of an assignment or as a subroutine argument for example).

suidperl is no longer needed since %s

(F) Your Perl was compiled with **-D**SETUID_SCRIPTS_ARE_SECURE_NOW, but a version of the setuid emulator somehow got run anyway.

Switch (?(condition)... contains too many branches in regex; marked by <– HERE in m/%s/

(F) A (?(condition)if-clause|else-clause) construct can have at most two branches (the if-clause and the else-clause). If you want one or both to contain alternation, such as using this|that|other, enclose it in clustering parentheses:

```
(?(condition)(?:this|that|other)|else-clause)
```

The <– HERE shows in the regular expression about where the problem was discovered. See *perlre*.

Switch condition not recognized in regex; marked by <– HERE in m/%s/

(F) If the argument to the (?(...)if-clause|else-clause) construct is a number, it can be only a number. The <– HERE shows in the regular expression about where the problem was discovered. See *perlre*.

switching effective %s is not implemented

(F) While under the use filetest pragma, we cannot switch the real and effective uids or gids.

%s syntax

(F) The final summary message when a perl -c succeeds.

syntax error

(F) Probably means you had a syntax error. Common reasons include:

```
A keyword is misspelled.
A semicolon is missing.
A comma is missing.
An opening or closing parenthesis is missing.
An opening or closing brace is missing.
A closing quote is missing.
```