

Padre::Document::Perl::Beginner(3pm) User Contributed Perl Documentation Padre::Document::Perl::Beginner(3pm)

NAME

Padre::Document::Perl::Beginner – naive implementation of some beginner specific error checking

SYNOPSIS

```
use Padre::Document::Perl::Beginner;
my $beginner = Padre::Document::Perl::Beginner->new;
if (not $beginner->check($data)) {
    warn $beginner->error;
}
```

DESCRIPTION

This is a naive implementation. It needs to be replaced by one using PPI.

In Perl 5 there are lots of pitfalls the unaware, especially the beginner can easily fall in. While some might expect the Perl compiler itself would catch those it does not (yet ?) do it. So we took the initiative and added a beginners mode to Padre in which these extra issues are checked. Some are real problems that would trigger an error anyway we just make them a special case with a more specific error message. (e.g. `use warning;` without the trailing `s`) Others are valid code that can be useful in the hands of a master but that are poisonous when written by mistake by someone who does not understand them. (e.g. `if ($x = /value/) { }`).

This module provides a method called `check` that can check a Perl script (provided as parameter as a single string) and recognize problematic code.

Examples

See <http://padre.perlide.org/ticket/52> and http://www.perlmonks.org/?node_id=728569

Cases

-

```
split /,, , @data;
```

Here `@data` is in scalar context returning the number of elements. Spotted in this form:

```
split /,, , @ARGV;
```

-

```
use warning;
```

`s` is missing at the end.

-

```
map { $_; } (@items), $extra_item;
```

is the same as

```
map { $_; } (@items, $extra_item);
```

but you usually want

```
(map { $_; } (@items)), $extra_item;
```

which means: map all `@items` and then add `$extra_item` without mapping it.

- Warn about Perl-standard package names being reused

```
package DB;
```

-

```
$x = chomp $y;
```

```
print chomp $y;
```

-



```
map { s/foo/bar/i } (@items);
```

This returns an array containing true or false values (`s///` – return value).

Use

```
map { s/foo/bar/i $_; } (@items);
```

to actually change the array via `s///`.

-

```
<@X>
```

-

```
if ($x = /bla/) {
}
```

- Pipe `|` in `open()` not at the end or the beginning.

-

```
open($ph, "| something |");
```

- Regular expression starting with a quantifier such as

```
/+.../
```

-

```
} else if {
```

-

```
} elsif {
```

-

```
close;
```

HOW TO ADD ANOTHER ONE

Please feel free to add as many checks as you like. This is done in three steps:

Add the test

Add one (or more) tests for this case to `t/75-perl-beginner.t`

The test should be successful when your supplied sample fails the check and returns the correct error message. As texts of error messages may change, try to match a good part which allows identification of the message but don't match the very exact text.

Tests could use either one-liners written as strings within the test file or external support files. There are samples for both ways in the test script.

Add the check

Add the check to the check-sub of this file (*Document/Perl/Beginner.pm*). There are plenty samples here. Remember to add a sample (and maybe short description) what would fail the test.

Run the test script to match your test case(s) to the new check.

Add the configuration option

Go to *Config.pm*, look for the beginner error checks configuration and add a new setting for your new check there. It defaults to 1 (run the check), but a user could turn it off by setting this to 0 within the Padre configuration file.

COPYRIGHT & LICENSE

Copyright 2008–2010 The Padre development team as listed in *Padre.pm*.



Padre::Document::Perl::Beginner(3pm) User Contributed Perl Documentation Padre::Document::Perl::Beginner(3pm)

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

The full text of the license can be found in the LICENSE file included with this module.

