Use strict and use warnings in Perl

BY CHAITANYA SINGH | FILED UNDER: PERL

You may find the following lines in almost every perl script.

```
use strict;
use warnings;
```

In this article, we will discuss about them one by one.
Note: You may not find these pragmas in some of the scripts available in this website, it is to avoid confusion to the beginners. However you will find them on advanced topics.

Lets start.

use strict

The use strict statement is called pragma and it can be placed at the beginning of the script like this:

```
#!/usr/local/bin/perl
use strict;
```

What does it do?
It forces you to code properly to make your program less error-prone. For example: It forces you to declare variables before you use them. You can declare variable using "my" keyword. "my" keyword restricts the scope of the variable to local. It makes the code more readable and less error prone.
If you don't declare variable using my keyword then the created variable would be global, which you should avoid, reducing the scope of the variable to the place where it is needed is a good programming practice.

Example:

If you use "use strict" but don't declare a variable.

```
#!/usr/local/bin/perl
use strict;
$s = "Hello!\n";
print $s;
```

It would throw this error:

```
Global symbol "$s" requires explicit package name at st.pl line 3.
Global symbol "$s" requires explicit package name at st.pl line 4.
Execution of st.pl aborted due to compilation errors.
```

To avoid the error you must declare the variable using my keyword.

```
#!/usr/local/bin/perl
use strict;
my $s = "Hello!\n";
print $s;
```

Output:

```
Hello!
```

Similarly, you need to declare the arrays and hashes before you use them.

Note: Starting with Perl 5.12, this pragma is implicitly enabled, which means that if you are using Perl 5.12 or later then there is no need to use the line **use strict** as the pragma is enabled by default.

use warnings

This is another pragma, together they are used like this:

```
#!/usr/local/bin/perl
use strict;
use warnings;
```

Note: use warnings pragma got introduced in Perl 5.6 so if you are using Perl 5.6 or later you are good to go. In case you are using older version you can turn on the warning like this: By putting -w on the 'shebang' line.

```
#!/usr/local/bin/perl -w
```

This would work everywhere even on Perl 5.6 or later.

What is the use of "use warnings"?
It helps you find typing mistakes, it warns you whenever it sees something wrong with your program. It would help you find mistakes in your program faster.

Note: The most important point to note here is that "use strict" would abort the execution of program if it finds errors. On the other hand use warnings would only provide you the warnings, it wont abort the execution.

Conclusion:
You should always use these two pragmas in your programs as it is a good programming practice.

❮ Previous Next ❯

Top Related Articles:

1. Perl Tutorial for beginners
2. Java Variables: Declaration, Scope, and Naming Conventions
3. my keyword – Local and global variables in Perl
4. First Perl Program
5. Perl – Strings

About the Author

I have 15 years of experience in the IT industry, working with renowned multinational corporations. Additionally, I have dedicated over a decade to teaching, allowing me to refine my skills in delivering information in a simple and easily understandable manner.

– Chaitanya

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment *

Name *

Email *

POST COMMENT

Perl Tutorial

Perl Strings

Perl Escape Sequences