

Survive Your First Day of Perl

by José Castro
cog@cpan.org



Welcome to Perl! Here's my advice on ways to find answers to the questions and problems that show up with your first hours in Perl. This isn't a complete guide, but it should be enough to get you started. If you already survived your first day of Perl, this guide should help you help your friends and co-workers survive their first day of Perl.

■ A tiny bit of Perl culture.-----

Some people write it "PERL", some people write it "Perl", and some people write it "perl". Contrary to popular belief, Perl is not an acronym—it's actually a backronym. That being said, "Perl" is the generally accepted way of writing the name of the language, while "perl" is the generally accepted way of writing the name of the interpreter.

Why is this important, you might ask? Well, it's just a small part of the Perl culture. When you get into a website full of Perl adepts and write it "PERL", they assume you don't know what you're talking about. They won't generally be mad at you, but if you want to blend in from day one, learn to tell the difference between the two.

The moral here is that while you can't judge people's Perl expertise by the way they spell Perl, they will still judge you. The lesson, on the other hand, is that there's lots of tiny cultural aspects of Perl that you'll only learn about with time.

■ Making sense out of sigils -----

Sigils are those funny characters in front of Perl variable names. Most programmers that have experience with other sigil-free languages find the small peculiarity of Perl sigils rather annoying.

Larry Wall, Perl's creator, intended from day-one to create a language that would be able to evolve. That meant that he couldn't risk the possibility of new built-ins clashing with variable names someone would use in the meantime.

Larry made his design decisions based on what he'd seen in the evolution of other languages. Think of all the C programmers who had a variable named `new` and that one day tried to compile their code in C++. You might think "but who'd name a variable `new`?" Well, back then they were thinking "but who'd name a built-in `new`?" Just to play it safe, Perl variables include the sigils in front of them. They'll never conflict with a Perl keyword.

And now, you need to be able to make sense out of sigils. They mean more than just the variable type, which is why some people are confused by them. The `$` sigil means single item and the `@` sigil means multiple items, also known as a list.

A scalar is preceded with a `$` because it's a single item. If you look closely, `$` is kind of similar to `S`, the first letter in "scalar". An array is preceded with a `@` sign because it represents multiple items. If you look closely, `@` is kind of similar to `a`, the first letter in "array".

A hash table is preceded with a `%` sign. If you look closely at your keyboard, nothing else looked like an `H`. It also doesn't quite fit into the single-multiple distinction. It's a fancier sort of variable, so it gets a fancier sigil.

Now, the trick is to realize that you always use the sigil corresponding to the value you're accessing, not necessarily the value type. When you declare an array, you use the `@` because there are multiple values involved:

```
my @prime_numbers = ( 2, 3, 5, 7, 11, 13 );
```

However, you use `$` when you're accessing a single value from within the array:

```
my $third_prime_number = $prime_numbers[2];
```

You still know that you're dealing with the `@prime_numbers` array because you see the `[]` indexing brackets after the variable name. You only see those with an array.

When you want to access a sublist of values from the array as a slice, you use the `@` sign, corresponding to what you're retrieving: a list:

```
my @first_five = @prime_numbers[ 0 .. 4 ];
```

That's the same for hash slices too, because you're still accessing a list:

```
my @values = @last_names{ @first_names };
```

As before, you know that you have a hash because you see the `{}` indexing brackets after the variable name, and you'll only see that with a hash.

■ Perl hygiene -----

As you probably know, some people don't wash their hands afterwards. Now, those people might argue that they don't need to, while those of us who do might argue that they're more susceptible to catch diseases. Washing your hands, even if you don't think you need to, helps everyone.