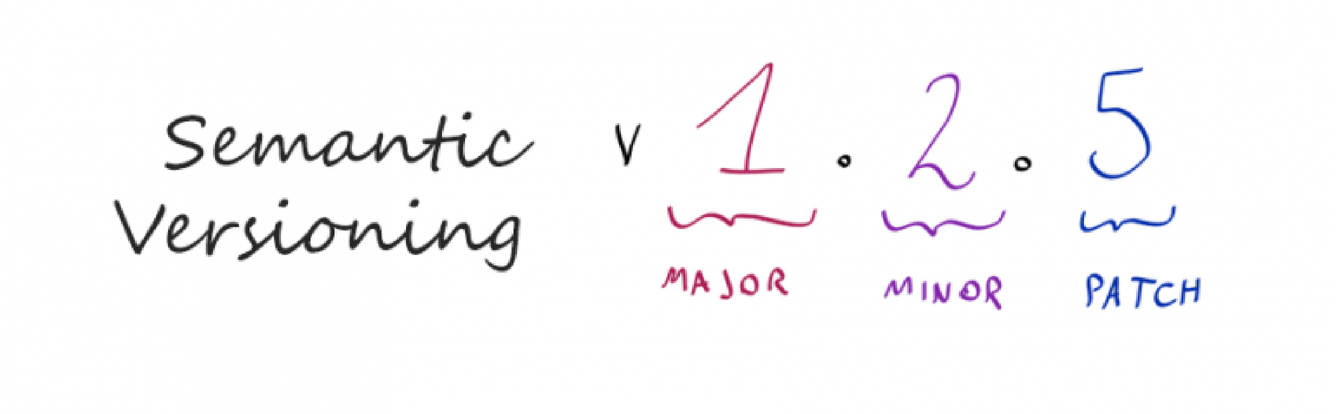
PERL HACKS

Just another Perl Hacker's blog

POSTS

13 DECEMBER, 2016

Version Numbers



Last week I mentioned how I had uploaded a new version of Symbol::Approx::Sub. Because there were pretty major changes to the inner workings of the module (although the interface still looked the same) I decided that I would move it from version 2.07 to version 3. At the same time, I decided that I would switch to a semantic versioning scheme.

Later in the week, I released minor updates to a few more of my modules. And I decided to apply semantic versioning to those as well. But as I was only making minor packaging fixes to these modules, I didn't increment the major version number. For example, Array::Compare went from 12.2 to 12.2.1.

It turns out that was a mistake.

Well, I don't really think it was a mistake. I think it was the right thing to do. But it appears that my opinion is at odds with what some parts of the Perl toolchain think.

Last night I got this bug report. It seems that by switching to three-part semantic versions, the version number can (in some quite common circumstances) appear to decrease.

To my mind, a version number is a dot-separated sequence of numbers. So 12.2 is smaller than 12.2.1. Any sane version number comparison will separate the two strings on dots and compare the individual components. Any missing components (12.2 is, for example, one component shorter than 12.2.1) should be assumed to be zero.

But that's not what the Perl toolchain does. Observe:

```
$ perl -Mversion -E"say version->parse('2.12')->normal"
v2.120.0
```

```
$ perl -Mversion -E"say version->parse('2.12.1')->normal"
v2.12.1
$ perl -Mversion -E"say version->parse('2.12') <=> version->parse('2.12.1')"
1
```

When the version number with two components (2.12) is split into components, the second component is bizarrely treated as a three-digit number so it becomes 12o instead of 12 and when it is compared with the second component of the three-component version, 120 is obviously larger than 12 and any tool which relies on this behaviour to work out which version of a module is the most recent will get the wrong answer.

This leads to other “interesting” effects. In my head, versions 1.1, 1.01 and 1.001 are all the same version. The leading zeroes mean nothing. But under this scheme, they are very different version numbers.

I know that versioning isn’t as easy as it should be and I know that some people use bizarre versioning systems. And I’m pretty sure that no matter how bizarre a versioning system is, you’ll almost certainly find an example of it on CPAN. So I suppose that this behaviour was a “least worse” scenario that was chosen to make the most sense given CPAN’s wide range of versioning schemes.

Personally, I see it as a bug in version.pm. But I’m not going to report it as such as I’m sure the Perl toolchain gang know what they’re doing and have very good reasons for adopting this seemingly broken behaviour.

I just need to remember to be more careful when switching my modules to semantic versioning. Using a minor or patch level version change when switching to semantic versioning is likely to lead to confusion and bug reports. Only a major level change (as I did with Symbol::Approx::Sub) is guaranteed to work.

And, I suppose, I’ll need to release Array::Compare 3.0.0 to CPAN pretty soon.

Share this:

6 DECEMBER, 2016

Hacking Symbol::Approx::Sub

Hacktoberfest
Support and celebrate open source. Earn a limited edition T-shirt.

In October, for (I think) the second year, Digital Ocean ran Hacktoberfest – a campaign encouraging people to submit pull requests to Github repos in exchange for free t-shirts.

A few of us thought that this might be a good way to do a small bit of easy Perl advocacy, so we tagged some issues on Perl repos with “hacktoberfest” and waited to see what would happen.

I created a few issues on some of my repos. But the one I concentrated on most was symbol-approx-sub. This is a very silly CPAN module that allows you to make errors in the names of your subroutines. I wrote it many years ago and there’s an article I wrote for *The Perl Journal* explaining why (and how) I did it.

Long-time readers might remember that in 2014 I wrote an article for the Perl Advent Calendar about Perl::Metrics::Simple. I used Symbol::Approx::Sub as the example module in the article and it showed me that the module had some depressingly high complexity scores and I planned to get round to doing something about that. Of course, real life got in the way and Symbol::Approx::Sub isn’t exactly high on my list of things to do, so nothing happened. Until this October.

Over the month, a lot of changes were made to the module. I probably did about half of it and the rest was pull requests from other people. The fixes include:

- Better tests (and better test coverage – it’s now at 100%)

- Using Module::Load to load module
- Using real exceptions to report errors
- Updating the code to remove unnecessary ampersands on subroutine calls
- Fixed a couple of long-term bugs (that were found by the improved tests)
- Breaking monolithic subroutines down

And I'm pretty happy with how it all went. The work was mostly completed in October and this morning I finally got round to doing the last couple of admin-y bits and version 3.0.0 of Symbol::Approx::Sub is now on the way to CPAN. You still shouldn't use it in production code though!

Thanks to everyone who submitted a pull request. I hope you did enough to earn a free t-shirt.

If you want to get involved in fixing or improving other people's code, there's the 24 Pull Request Challenge taking place over Advent. Or for more Perl-specific code, there's the CPAN Pull Request Challenge.

p.s. In the Advert Calendar article, I linked to the HTML version of the results. For comparison, I've also put the new results online. It's a pretty good improvement.

Share this:

Tweet Share 0 submit More

12 NOVEMBER, 2016

The Fragility of Contracting

I’ve been rather quiet for a few months. That’s because I’ve been working for a large investment bank in Canary Wharf. It’s no so much that the work takes up more of my time than other contracts I’ve had, but more that the incredibly restrictive firewalls banks have around their networks have meant that I have far less ability to keep in touch with things during the working day. I understand security is so important to them but, wow, it’s hard having to live with it.

Working in the finance sector is lucrative, but not much fun (which, I suppose, might explain why they make it so lucrative).

But all that is about to change. On Wednesday, the project leader told me that the bank were letting all of the contractors in the group go. It had come as a complete surprise to him too – he had just received an email telling him to let us know. That’s the way things work in the banking sector.

I’m not sure if it was a slow reaction to Brexit or an extremely quick reaction to Trump or something else completely. But we’ll all be leaving at the end of this month.

Which means that I’m looking for a new contract. So if you’re reading this and you know of a team who are looking for a contractor then please let me know and we might be able to work something out.

Because of way this was timed, I think I'll probably be looking for something to start at the beginning of next year. I'm going to South Africa for a couple of weeks at the end of the year and it seems pretty pointless to do a couple of weeks at a new job before going away for a while and forgetting everything I've learned.

But it would be nice if I didn't spend all of the first two weeks of December watching Netflix. So there are a few possibilities I'm considering:

- Could I find magazines or web sites that would pay me to write articles for them?
- Could I go into a company for a few days of consultancy (perhaps an architectural review or something like that)?
- Could I do a code review for some of your companies codebase?

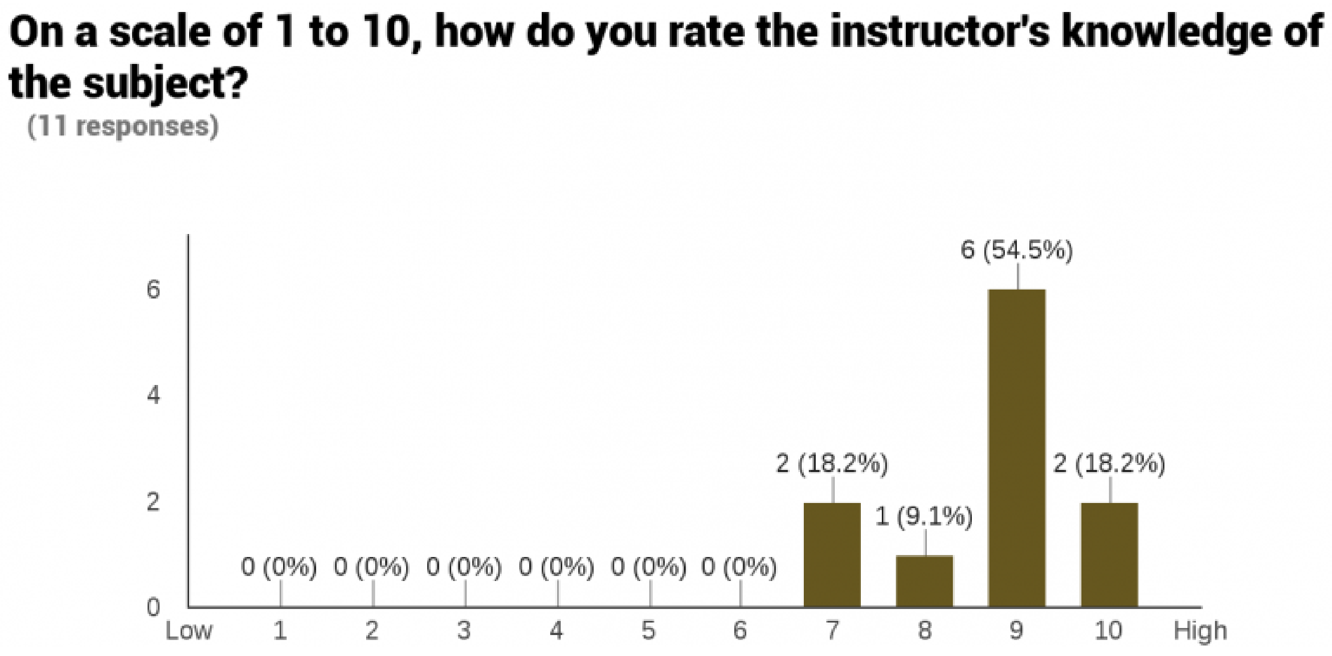
Or, the most likely option:

- Do you have colleagues who could benefit from a few days Perl training? Have you been vaguely thinking "you, know it might be nice to get Dave in to run some in-house training"? If that's the case, then the first couple of weeks of December would be a great time to get more serious about this.

In fact, if there's any way that you think I could be of use to your company for a few days in December or on a longer-term basis from January, then please get in touch.

25 SEPTEMBER, 2016

Feedback



During the week, Barbie sent out the results from the feedback survey that he ran after YAPC Europe. The general results will be published later, but all of the speakers will have received an email containing the feedback from their talks. That feedback is private, but I'm happy to share mine with the world.

The feedback survey takes the form of five questions. People are asked to answer these questions with a rating from 1 to 10. The questions are:

- Q1: Your prior knowledge of subject?
- Q2: Speaker's knowledge of subject?
- Q3: Speaker's presentation of subject?
- Q4: Quality of presentation materials?
- Q5: Overall presentation rating?

There is also an opportunity for people to write in more detailed comments if they want.

I gave two talks at the conference. A lightning talk called "Medium Perl" which introduced the idea of the Cultured Perl blog and a longer talk called "Error(s) Free Programming" which talked about Damian Conway's module Lingua::EN::Inflexion.

Eight people gave feedback on "Medium Perl".

Qu	1	2	3	4	5	6	7	8	9	10	Avg
Q1	1	1	–	2	1	2	–	1	–	–	4.5

Q2	–	–	–	–	–	–	1	1	1	5	9.25
Q3	–	–	–	–	–	1	1	1	–	5	8.875
Q4	–	–	–	–	–	1	1	1	–	5	8.875
Q5	–	–	–	–	–	–	2	1	1	4	8.875

What aspects of the tutorial or presentation worked really well?

- I always enjoyed Dave's humor.
- History and goal are clear
- Excellent presentation, as you always do. Funny and surprising.

How could the tutorial or presentation be improved?

- Make Medium use a readable font or have them stop forcing me to use serif fonts. As long as the articles are presented as they are, I won't read them at all. Period. (let alone open the possibility that I would post any material myself)

I'm not really sure how I'm supposed to make Medium change their fonts. I suppose I could suggest that they make other fonts available as an option. But then, so could the person who made that comment.

Four people gave feedback on "Error(s) Free Programming".

Qu	1	2	3	4	5	6	7	8	9	10	Avg
Q1	1	–	–	–	–	1	–	1	1	–	4
Q2	–	–	–	–	–	–	–	–	3	1	9.25
Q3	–	–	–	–	–	–	–	–	2	2	9.5
Q4	–	–	–	–	–	–	–	–	1	3	9.75
Q5	–	–	–	–	–	–	–	–	2	2	9.5

What aspects of the tutorial or presentation worked really well?

- Just about everything, an excellent presentation. Congrats.

- Damianware!

  How could the tutorial or presentation be improved?

- I misunderstood the topic, and I thought it was a talk about programming without errors instead of how to solve localization of messages.

  I can only suggest that the last people reads the talk description, not just the title in future.

  I also got feedback about the “Modern Web Programming with Perl and Dancer” course that I ran before the conference. The feedback here is in a slightly different format as it’s a form that I made up myself. I got feedback from 11 people.

1	2	3	4	5	6	7	8	9	10	Avg
On a scale of 1 to 10, how do you rate your Perl ability?										
–	–	–	1	1	2	2	3	1	1	7.09
On a scale of 1 to 10, how useful did you find the course?										
–	–	–	2	–	–	1	6	1	1	7.45
On a scale of 1 to 10, how much did you enjoy the course?										
–	–	–	–	–	1	–	5	2	3	8.54
On a scale of 1 to 10, how do you rate the instructor’s knowledge of the subject?										
–	–	–	–	–	2	1	6	2	–	8.72
On a scale of 1 to 10, how well did the instructor teach the subject matter?										
–	–	–	–	–	1	2	2	4	2	8.36
On a scale of 1 to 10, please rate the amount of material covered										
1	–	–	1	3	1	1	2	–	2	6.27

That last question is always tricky. The form is clear that if you think it was just right, to score 5. But I always get some people choosing 10 and I think I’d know if people thought I was covering stuff far too quickly. That 1 is a bit of a worry though.

So, all in all, not bad scores. And generally people saying nice things. Which is always nice to see.

Now I need to start thinking about the London Perl Workshop.

Share this:

Tweet Share 0 submit More

3 SEPTEMBER, 2016

YAPC Europe 2016

I've been back from Cluj-Napoca for almost a week, so I should really write down what I remember about YAPC Europe before it's all forgotten.

Day -1

I arrived in Cluj-Napoca on Sunday evening and got to my hotel quickly. There was just time for a quick meal before bed.

Monday was the day that I was going to get to grips with the city. After meeting a few Perl Mongers at breakfast, my wife and I set off to explore. My first target was to find Cluj Hub, the venue where I was running a training course the following day. That was simple and took less than fifteen minutes. We then explored both the Orthodox and Catholic cathedrals before settling into a bar on the main square called "Guevara" for a coffee. After that we decided that we needed to pick up some supplies and whilst on that hunt we bumped into Max Maischein who recommended a visit to the botanical gardens.

On returning to the hotel with our supplies, we met Curtis Poe and invited him to join us for lunch. Wandering at random we found a really good restaurant called Livada and enjoyed a very pleasant meal.

After lunch we spent a very enjoyable couple of hours in the botanical gardens and only just failed to get back to the hotel before it really started raining. That evening we ate in restaurant really close to the hotel called the Crying Monkey (but in Romanian).

Day 0

Tuesday was my "Modern Web Development with Perl and Dancer" training course. This was by far the most successful training course that I've ever run at a YAPC. I'll write more about it when I get the feedback results, but I think that the attendees enjoyed it. I know I had great fun giving it. Cluj Hub was a great venue and Andra Gligor and her small team looked after us all really well.

That evening, the traditional pre-conference meet-up was held on the roof of Evozon's offices. As always, it was lovely to catch-up with old friends that I only get to see once or twice a year.

Day 1

On Wednesday, I set off in plenty of time to find the venue. It turned out that our hotel was really well located for both sight-seeing and the conference and I got there in ten minutes or so. The registration queues seemed shorter than usual and before long I had my name-tag and bag of conference swag.

As always, there were far too many good talks and it was impossible to see everything. I'll just talk about the talks that I saw. Everything was videoed, so it will all be online soon.

The day began with Amalia welcoming us to the conference. Then the YAPC Europe Foundation announced that next year's conference will be in Amsterdam. This is the first time that the conference has returned to a previous city (the second YAPC Europe was held in Amsterdam back in 2001) and I'm looking forward to going.

The first day's keynote was from Curtis Poe. It was a wide-ranging talk covering the history and future of both Perl and the Perl community. After that I went into one of the small rooms to see H. Merijn Brand talking about his recent improvements to Perl's CSV parser followed by Alex Muntada on how the Debian project packages CPAN modules. I then went back to the main room to see Mickey Nasriachi talking about PONAPI, which is a Perl implementation of JSONAPI.

Lunch suffered slightly from the inevitable queues, but it was worth the wait as the quality of the food (as it was throughout the conference) was very high.

After lunch I saw Lee Johnson giving some Git tips, Sawyer talking about the XS guts of Ref::Util and Jose Luis Martinez talking about PAWS (the Perl interface to Amazon Web Services). I saw Jose Luis talking about PAWS last year in Granada but really wanted to see how the project is progressing. I think this has the potential to be a great advocacy tool for Perl.

A quick coffee break and then I saw Thomas Klausner give his opinions on writing API endpoints and Tina Müller talking about App::Spec which looks like a great tool for easily writing command line applications.

Then it was was lightning talks. They were the usual combination of the useful, the banal and the ridiculous. I think the highlight for me was Curtis Poe announcing more details of his online game (which is now officially called Tau Station). This was the point at which I announced *Cultured Perl* – which seems to be going well so far.

That evening was the conference dinner. Which was a buffet party held in the open-air quadrangle at the centre of the Banffy Palace (Cluj's major art museum). A great time was had by all.

Day 2

Another day, another keynote. This time it was Sawyer X with "The State of the Velociraptor" – an annual round-up of what's going on in the Perl 5 world. This year Sawyer found a number of volunteers who all gave short talks about their part of the Perl community. This was a great idea which was only slightly marred by the fact that the projector wasn't at all happy changing laptops – so the switches between presenters weren't as smooth as they could have been.

After that I saw Max talking about how he uses ElasticSearch on his laptop to give himself a local search engine and Job van Achterberg talking about making web sites more accessible. This was a great talk – particularly the sections where he showed just how bad most web sites appear to screen readers.

Another queue for another great lunch. And also many interesting conversations.

After lunch I saw a former colleague, Mirela Iclodean, talking about how her company have managed to shoe-horn many modern tools and practices into their working day – while still maintaining a nasty monolithic code-base which they are slowly chipping away at. It was a great talk and it made me miss working on that project. I'm hoping that she will repeat this talk at the London Perl Workshop.

Later that afternoon, I gave my "Error(s) Free Programming" talk – in a slot where every speaker was a London.pm leader. The talk seemed to go down well, but somehow I ran considerably short.

After that I saw Albert Hilazo talk about his first few months as a Perl programmer. I found this really interesting as Albert talked in some detail about things that other language communities provide but he found hard to find for Perl. In particular, he would like to see more "war story" blog posts showing how people have solved particular problems using Perl tools.

Then it was Matt Trout celebrating ten years in the Perl community by explaining how his career was largely a series of happy accidents and that a lot of the responsibilities he has taken on were just through being in the wrong place at the wrong time – or something like that.

One talk I couldn't miss was Andrew Yates talking about the work that his team do at the European Bioinformatics Institute. I couldn't miss it as I was at least partly responsible for Andrew proposing the talk. I ran some training at the EBI earlier this year and during our email conversation YAPC was mentioned and Andrew asked if people might be interested in hearing about their work. I replied "hell, yes!" and sent him a link to the talk proposal web page.

And then, of course, there another ten or so lightning talks to close the day entertainingly.

Day 3

The keynote speaker on the last day was Larry Wall. His topic was "Strange Consistency". If you've seen Larry speak before, you'll know what it was like.

I followed that by watching Jason Clifford talk about how his team had written a major new toolset in Perl despite management pressure to use other technologies. The project, of course, ended up being very successful.

One of the most interesting talks was Nicholas Clark's view of an alternative universe where Jon Orwant never threw those mugs in 2000 and the Perl 6 project was never started. The main lesson appeared to be "what goes around, comes around" and his fictional universe didn't end up too far away from where we are now.

The afternoon had a curious combination of some time slots where I wanted to see every talk and others where I didn't really want to see anything. So in some cases I'm eagerly awaiting the videos going online and in others I sat in the back of the room only half-concentrating while giving most of my attention to Twitter or Facebook.

I really enjoyed Sawyer talking about the things that were added in Perl 5.24 (and very carefully not talking about the things that were added in previous versions) and also Jose

Luis Perez talking about what he has got out of doing the CPAN Pull Request Challenge.

The final lightning talks were as much fun as they always are. The projectors were still giving the speakers plenty of technical difficulties which led to lots of time for “lightning adverts” between the talks. I think that towards the end the differences between the two rather broke down and on the video at one point I expect you’ll hear Geoff Avery saying “I seem to have lost control of this”.

The conference ended, as it always does, with a brief presentation from the organisers of next year’s conference, a final thank-you to all of the speakers and sponsors and a standing ovation for the organisers.

This was one of the best-organised YAPCs I’ve been to for a very long time. And Cluj-Napoca is a city I would never have considered visiting if it wasn’t for the Perl community there. And already I’m considering a return visit. I had a lovely time in the city and returned to London completely recharged and reinvigorated.

See you all in Amsterdam next year.

Share this:

Tweet Share 0 submit More