Tip 8: Use here scripts

Ben Klemens

15 October 2011

level: intermediate POSIX
purpose: fewer temp files floating around

I'll relate this to C in a few episodes, but this is a general feature of POSIX-compliant shells that you can use for Python, Perl, or whatever else. In fact, if you want to have a multilingual script, this is an easy way to do it. Do some parsing in Perl, do the math in C, then have R produce the pretty pictures, and have it all in one text file.

Here's a Python example. Normally, you'd tell Python to run a script via

```
python your_script.py
```

You can give the file name `'-'` to use standard in as the input file:

```
echo "print 'hi.'" | python '-'
```

[Subtip: We need `'-'` and not just - to indicate that this is plain text and not introducing a switch like the *c* in `python -c "print 'Hi'"`. Many programs follow a custom that two dashes indicate that they should stop reading switches and read subsequent inputs plain. Thus
`echo "print 'hi.'" | python -- -`
also works, but is the sort of thing that scares people.]

You could, in theory, put some lengthy scripts on the command line via `echo`, but you'll quickly see that there are a lot of little undesired parsings going on—you might need `\"hi\"` instead of just `"hi"`, for example.

Thus, the *here script*, which does no parsing at all. Try this:

```
python '-' <<"XXXX"
lines=2
print "\nThis script is %i lines long.\n" %(lines,)
XXXX
```

The `XXXX` is any string you'd like; `EOF` is also popular, and `-----` looks good as long as you get the dash count to match at top and bottom. When the shell sees your chosen string alone on a line, then it will stop sending the script to the program's stdin. That's all the parsing that happens.

discussion:
This tip is a standard shell feature, and so should work on any POSIX system. I know

because the POSIX standard[1] is online and is not all that painful to read, as standards go. Unfortunately, I'm not sure how to link to a certain line, so you'll have to go searching to verify my promise that Here Documents are standard shell features.

There's also a variant that begins with `<<-`. Search the standard or ask `man bash` for details.

As another variant, there's a difference between `<<"XXXX"` and `<<XXXX`. In the second version the shell parses certain elements, which means you can have the shell insert the value of `$shell_variables` for you.

[1] `http://pubs.opengroup.org/onlinepubs/009695399/`