

## String manipulation in action

Let's face it. Surveying literature is boring. That's why your guide makes *you* do it. It's easy to make Python do it for you instead. You don't even have to feel bad about it – Python has no feelings! Unless, of course:



`docs` has a collection of documents related to programming history and programming practices. Yawn. Your guide has asked you to work with Python for this project, and you don't want to waste your time sifting through irrelevant literature. You can write a program that:

1. Counts the number of times the word 'Python' appears in a document (if at all)
2. Sorts documents by the frequency of occurrence

Silly as this example may seem, you can easily imagine that even a slightly more complex program can save you a lot of time, sweat and tears.

### Useful tips:

- To convert a pdf to a text file, use the shell command `pdftotext foo.pdf`
- To convert *all* pdfs to text files, do:

```
for file in *.pdf
do
pdftotext $file
done
```

- Pass text files to Python as command-line arguments.
- See `f.read()` and `str.split()`