Why Cant I Redd Arabic in Mah Files?

[Cross-posted from ICanHazDataScience]

I work on development data. Sometimes on datafiles, sites or streams that cover a large part of the world. Which means that, sooner or later, I'm going to get an error that looks like this when I'm reading something into Python:

"UnicodeEncodeError: ascii codec can’t encode character u’\u015e’”.

At this point, you need Unicode.   Unicode is like the babelfish of written text: it contains characters for most human languages, including Arabic, so for instance it can deal with reading data from websites where multiple human languages are used (e.g. at least 10 on one of the sites that I maintain). Most people's files contain just one character set (things like ASCII and Latin-1) and don't ever see the problem above - we development nerds are likely to see it a lot! For example, this line (placed at the top of your code file) can save you pain when you're writing non-ascii characters in your code (e.g. in comments or text strings):

```
# -*- coding: utf-8 -*-
```

And this line can save you a lot of pain when you're looking at your files:

"Print(data.encode(‘utf-8’))"

But they didn't say "unicode", Emily says (before yawning and settling back down to sleep). That's because you probably won't see the word "unicode" in Python code... you're more likely to see words like "utf-8" and "utf-16". These are both implementations of unicode - different ways of representing unicode characters as 1, 2 or more 8-bit bytes. For Arabic and other common languages, utf-8 is more than sufficient (and very few people need or use utf-16).   I could rattle on here (and if you need it, can supply the magic words for e.g. reading an excel file in UTF-8) but this blogpost about Arabic in Python should cover all (or at least most) of what you need to get started.

Play, experiment, and remember to search StackOverflow if you get stuck (chances are, that whatever Python throws at you already has a question and answer on StackOverflow). More geeky notes on unicode include Green Notes' blogpost and Evan Jones' note.