Unidecode

Tomaž Šolc

tomaz.solc@tablix.org

Python meetup, Kiberpipa, 8. 10. 2013

»kožušček«

xbbkou017euu0161u010dekxab

?ko?u??ek?

kouek

"kozuscek"

Unicode -> ASCII

- V nekaterih sistemih ne želimo nabora Unicode
  - URL naslovi spletnih strani, imena datotek
  - `django.utils.slugify()`
- Uporabniki
  - Na tipkovnici ni prave tipke. Kaj je to „character map“?!
  - „7 bits are enough for anyone“ mentaliteta[1]
  - `solr.ASCIIFoldingFilterFactory()`
- ~~Varnostni razlogi~~

[1] http://widgetsandshit.com/teddziuba/2009/07/this-is-america-take-your-unic.html

Če ne veste kaj je...

- ...razlika med Unicode in UTF-8?
- ...razlika med `u"kožušček"` in `"kožušček"`?
- ...razlika med `u"\u0161"` in `"\xc5\xa1"`?
- ...narobe z `u"kožušček".decode("utf8")`?

- Beri http://docs.python.org/2/howto/unicode.html

```
>>> u"»kožušček«".encode("ascii")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
UnicodeEncodeError: 'ascii' codec can't
encode character u'\u017e' in position 2:
ordinal not in range(128)
```

Bah…

```
>>> u"»kožušček«".encode("ascii", "ignore")
'kouek'
```

Boljše...

```
>>> import unicodedata
>>> unicodedata.normalize('NFKD',
u"»kožušček«").encode('ascii', 'ignore')
'kozuscek'
```

Še boljše[1]...

[1] http://stackoverflow.com/questions/2365411/python-convert-unicode-to-ascii-without-errors

Ampak...

```
>>> unicodedata.normalize('NFKD',
u'»kožušček«')
u'\xbbkoz\u030cus\u030cc\u030cek\xab'
```

(normal form KD – canonical decomposition)

```
>>> from unidecode import unidecode
>>> unidecode(u"»kožušček«")
'>>kozuscek<<'
```

Najboljše (?)

Unidecode

- Preprosta substitucija znak za znak
  - 110000+ Unicode -> 127 ASCII
- Ročno narejena tabela
  - Sean Burke
    (Perl programer in lingvist[1])
  - Ločila, matematični znaki, simboli, itd.

[1] https://en.wikipedia.org/wiki/Sean_M._Burke

Kako dobro deluje?

- Za večino evropskih jezikov dovolj dobro, da se brez težav razume, kaj je bil original.

```
>>> unidecode(u"Wörterbuch")
'Worterbuch'
```

- Ruščina, arabščina, kitajščina zadovoljivo.

```
>>> unidecode(u"сообще́ния")
'soobshcheniia'
>>> unidecode(u"北京 ")
'Bei Jing '
```

- Japonščina, korejščina neuporabno.

Pozor, transliteracija!

- Tako zahteven problem kot strojno prevajanje[1]
- Transliteracija z Unidecode je tako, kot prevajanje besedila po slovarju besedo po besedo.
- Ljudje so užaljeni, če vidijo svoj jezik pohabljen
  - Kdaj je bolje imeti nečitljiv hex blob kot približno čitljiv zmazek
  - Uporabite rešitev, specifično za jezik[2] (*katerega?*)

[1] http://interglacial.com/tpj/22/

[2] https://github.com/miurahr/unihandecode

Mimogrede

- `assert isinstance(foo, unicode)`
  `unidecode(foo)`
- Ne zanašajte se na to, da se med različnimi verzijami rezultat ne bo spreminjal.
- URL, domene, datoteke (večinoma) podpirajo Unicode že nekaj časa
  - https://github.com/mozilla/unicode-slugify
- stringprep, nameprep
  - http://docs.python.org/2/library/stringprep.html

Namestitev

- https://pypi.python.org/pypi/Unidecode

```
pip install unidecode
```

- GPL 2+
- Python 2.x, Python 3.x, PyPy, ...
  (tudi Perl, Ruby, C#, JavaScript, itd.)

Povezave

- Blog ob objavi Unidecode 0.04.14

  http://www.tablix.org/~avian/blog/archives/2013/09/python_unidecode_release_0_04_14

- Dokumentacija Perl modula

  http://search.cpan.org/~sburke/Text-Unidecode-0.04/lib/Text/Unidecode.pm

Vprašanja?

Tomaž Šolc

tomaz.solc@tablix.org

@avian2

https://pypi.python.org/pypi/Unidecode