Использование Unicode в Python

Юрий Юревич
http://gorod-omsk.ru/blog/pythy/

yury@yurevich.ru

Конференция по Ruby и Python.
Омск, 10 февраля 2007

Понятия

Кодировки

Кодировка — это таблица соответствий между символами и их машинными представлениями.

Буква «а»:

- windows-1251 — `E0h`
- cp866 — `A0h`
- koi8-r — `C1h`
- utf-8 — `D0h B0h`

Юникод (unicode)

Юникод — стандарт кодирования. Не говорит о конкретном представлении символа.

Буква «а»:

- unicode — кодовая точка `U+0430`

Типы строк в Python

Обычная строка

```
>>> regular_string = 'обычная строка'
>>> type(regular_string)
<type 'str'>
>>> 'а'
'\xd0\xb0'
```

Юникод-строка

```
>>> unicode_string = u'юникод-строка'
>>> type(unicode_string)
<type 'unicode'>
>>> u'а'
u'\u0430'
```

Преобразования между типами строк

Строка → юникод

```
>>> regular_string = 'обычная строка'
>>> type(regular_string)
<type 'str'>
>>> unicode_string = regular_string.decode('utf-8')
>>> type(unicode_string)
<type 'unicode'>
```

Юникод → строка

```
>>> unicode_string = u'юникод-строка'
>>> type(unicode_string)
<type 'unicode'>
>>> regular_string = unicode_string.encode('utf-8')
>>> type(regular_string)
<type 'str'>
```

Юникод != UTF-8

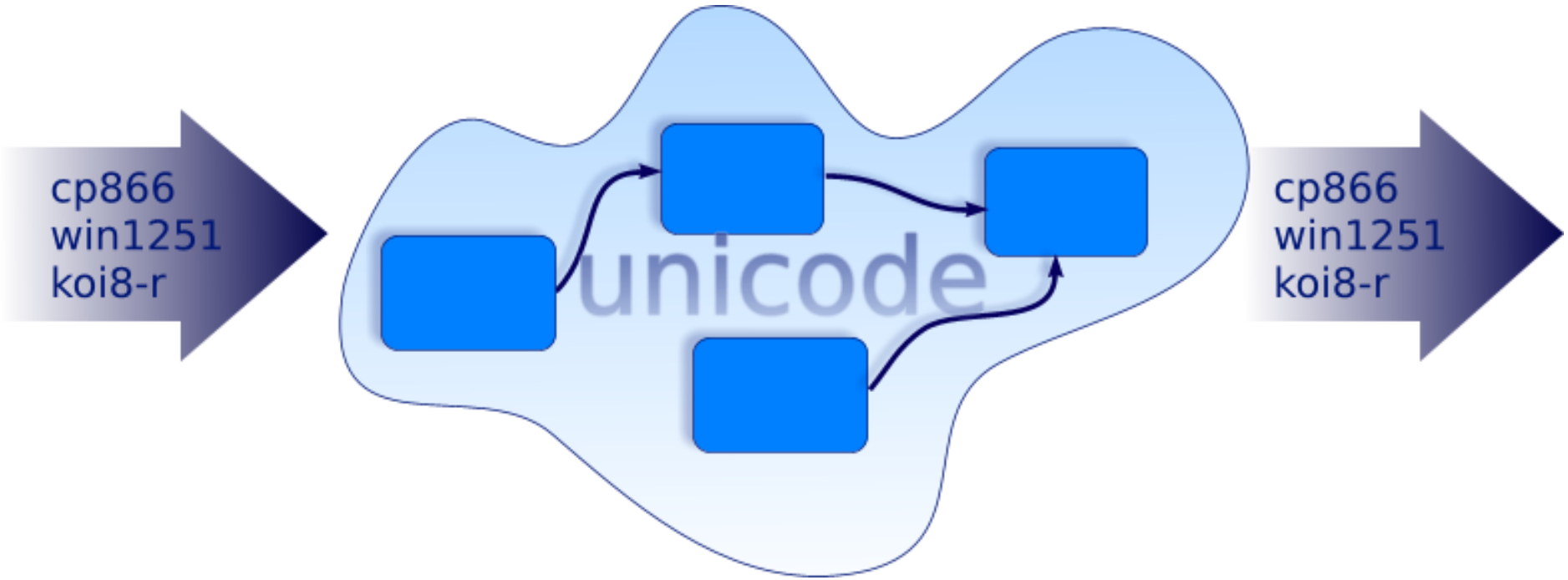
Юникод

```
>>> unicode_string = u'юникод строка'
>>> print unicode_string[:3]  # -> юни
юни
>>> len(unicode_string)       # -> 13
13
```

UTF-8

```
>>> utf8_string = 'utf8 строка'
>>> print utf8_string[:3]   # -> utf
utf
>>> print utf8_string[-3:]  # -> ока?
?a  #  WTF?
>>> len(utf8_string)        # -> 11?
17  #  WTF?
```

Юникод в ваших программах



Типичные ошибки: неявное перекодирование

Неявное перекодирование

```
>>> unicode(regular_string)
UnicodeDecodeError: 'ascii' codec can't decode byte ...
```

- Часто встречается: у англоязычных разработчиков
- Неправильное решение: исправить `'ascii'` в `site.py` на используемую кодировку
- **Правильное решение**: использовать явную перекодировку

  ```
  >>> unicode(regular_string, 'utf-8')
  ```

  либо

  ```
  >>> regular_string.decode('utf-8')
  ```

Типичные ошибки: не тот тип данных

Не тот тип данных

```
>>> 'ожидается юникод'.encode('utf-8')
UnicodeDecodeError: 'ascii' codec can't decode byte ...

>>> u'ожидается строка'.decode('utf-8')
UnicodeEncodeError: 'ascii' codec can't encode characters ...
```

- Часто встречается: у ленивых или невнимательных русскоязычных разработчиков
- Неправильное решение: исправить 'ascii' в site.py на используемую кодировку
- **Правильное решение**: проверять типы данных

```
>>> isinstance('ожидается юникод', unicode)
False
```

Типичные ошибки: изменение site.py

Изменение site.py — «универсальное» решение проблем

- Часто встречается: у ленивых русскоязычных разработчиков
- Неправильное решение: исправить 'ascii' в site.py на используемую кодировку
  - Теряется переносимость программы
    - Зависимость поведения программы от окружения
    - Завязка на конкретную кодировку
  - Ложное чувство правильности работы программы
- **Правильное решение**: использовать явную перекодировку, указывать англоязычным разработчикам на ошибки использования юникод

Спасибо

Спасибо за внимание

Вопросы?