

Лекция 6
Работа с текстом.
Декораторы.

24 марта 2016 г.

Работа с кодировками

Кодировки

Кодировка

- Алгоритм кодирования E
Символы → **Байты**
- Алгоритм декодирования D
Байты → **Символы**

$$D = E^{-1}$$

Кодировки

Кодировка

- Алгоритм кодирования E
Символы \rightarrow **Байты**
- Алгоритм декодирования D
Байты \rightarrow **Символы**

$$D = E^{-1}$$

Обычно просто таблица **Символ** \leftrightarrow **Байты**

ASCII

ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

UTF

UTF-32

Символ → 4 байта.

a		0x00	0x00	0x00	0x61
b		0x00	0x00	0x00	0x62
Й		0x00	0x00	0x04	0x19

UTF-8

Символ → 1-4 байта.

Совпадает с ASCII на символах ASCII.

a		0x61
b		0x62
Й		0xD0 0x99

Стандарт Unicode

Стандарт Unicode

- UCS (Universal Character Set).
Таблица символов.

a	97
b	98
Й	1049

Стандарт Unicode

- UCS (Universal Character Set).

Таблица символов.

a	97
b	98
Й	1049

- UTF (Unicode Transformation Format).

Кодировки UTF-8, UTF-16, UTF-32.

Тип `str`

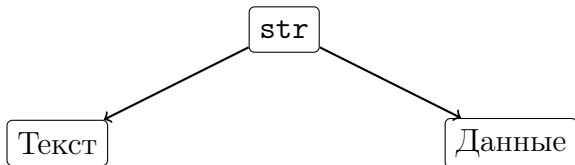
`str` — последовательность байт

'a'	0x61
'\n'	0x0A
'\xff'	0xFF

Тип `str`

`str` — последовательность байт

'a'		0x61
<hr/>		
'\n'		0x0A
<hr/>		
'\xff'		0xFF



Отображение str

```
>>> 'a'  
'a'  
>>> '\x61'  
'a'  
>>> '\x0a'  
'\n'  
>>> '\xff'  
'\xff'
```

Отображение str

```
>>> 'a'
'a'
>>> '\x61'
'a'
>>> '\x0a'
'\n'
>>> '\xff'
'\xff'

>>> '\xf0\x20\xd5\x63\x33'
'\xf0 \xd5c3'
```

Тип unicode

unicode — последовательность символов.

u'Q'	Q
u'Щ'	Щ
u'\u0449'	Щ

Тип unicode

unicode — последовательность символов.

u'Q'	Q
u'щ'	щ
u'\u0449'	щ

Отображение

```
>>> u'привет'  
u'\u043f\u0440\u0438\u0432\u0435\u0442'  
>>> u'йQ ?ф'  
u'\u0439Q ?\u0444'
```

str и unicode

str и unicode

unicode → str

```
>>> u = u'эюя'  
>>> s = u.encode('utf-8')  
>>> s  
'\xd1\x8d\xd1\x8e\xd1\x8f'
```

str и unicode

unicode → str

```
>>> u = u'эюя'  
>>> s = u.encode('utf-8')  
>>> s  
'\xd1\x8d\xd1\x8e\xd1\x8f'
```

str → unicode

```
>>> u2 = s.decode('utf-8')  
>>> u2  
u'\u044d\u044e\u044f'
```

Кодирование данных

```
with open('in.txt') as fin:  
    data_in = fin.read()  
text_in = data_in.decode('utf-8')
```

Кодирование данных

```
with open('in.txt') as fin:  
    data_in = fin.read()  
    text_in = data_in.decode('utf-8')  
  
... = text_in  
...  
text_out = ...
```

Кодирование данных

```
with open('in.txt') as fin:
    data_in = fin.read()
    text_in = data_in.decode('utf-8')

... = text_in
...
text_out = ...

data_out = text_out.encode('utf-8')
with open('out.txt', 'w') as fout:
    fout.write(data_out)
```

Кодирование данных и codecs

```
import codecs

with codecs.open('int.txt',
                 encoding='utf-8') as fin:
    text_in = fin.read()

... = text_in
...
text_out = ...

with codecs.open('out.txt', 'w',
                 encoding='utf-8') as fout:
    fout.write(text_out)
```

Кодирование исходного кода

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

```
x = u'эюя'
```

```
# Комментарий
```

Строки в Python 3

Тип	Запись	Аналог в Python 2
<code>str</code>	<code>'Привет'</code>	<code>unicode</code>
<code>bytes</code>	<code>b'abc\xff'</code>	<code>str</code>

Регулярные выражения

Регулярные языки

Алфавит $\mathcal{A} = \{0, 1\}$

Выражение	Формальный язык
0	$\{0\}$
$(0 1)1$	$\{01, 11\}$
001^+	$\{001, 0011, 00111, \dots\}$
0^*10^*	$\{1, 01, 10, 010, 001, \dots\}$

Простые регулярные выражения

```
text = """In September 1769 she reached New  
Zealand, the first European vessel to visit  
in 127 years."""
```

```
import re
```

Простые регулярные выражения

```
text = """In September 1769 she reached New  
Zealand, the first European vessel to visit  
in 127 years."""
```

```
import re
```

```
m = re.search('i', text)  
print m
```

```
<_sre.SRE_Match object at 0x3693648>
```

Простые регулярные выражения

```
text = """In September 1769 she reached New  
Zealand, the first European vessel to visit  
in 127 years."""
```

```
import re
```

```
m = re.search('i', text)  
print m
```

```
<_sre.SRE_Match object at 0x3693648>
```

```
print m.start(), m.end()
```

```
48 49
```

Простые регулярные выражения

```
text = """In September 1769 she reached New  
Zealand, the first European vessel to visit  
in 127 years."""
```

```
def search_results(pattern, text):  
    for m in re.finditer(pattern, text):  
        print "'{0}': {1}-{2}".format(  
            m.group(), m.start(), m.end())
```

Простые регулярные выражения

```
text = """In September 1769 she reached New  
Zealand, the first European vessel to visit  
in 127 years."""
```

```
def search_results(pattern, text):  
    for m in re.finditer(pattern, text):  
        print "{0}: {1}-{2}".format(  
            m.group(), m.start(), m.end())  
  
search_results('i', text)
```

```
'i': 48-49  
'i': 73-74  
'i': 75-76  
'i': 79-80
```

Множества символов

```
text = """In September 1769 she reached New  
Zealand, the first European vessel to visit  
in 127 years."""
```

```
search_results('(i|I)n', text)
```


Множества СИМВОЛОВ

```
text = """In September 1769 she reached New  
Zealand, the first European vessel to visit  
in 127 years."""
```

```
search_results('(i|I)n', text)
```

```
'In': 0-2
```

```
'in': 79-81
```

Множества СИМВОЛОВ

```
text = """In September 1769 she reached New  
Zealand, the first European vessel to visit  
in 127 years."""
```

```
search_results('(i|I)n', text)
```

```
'In': 0-2
```

```
'in': 79-81
```

```
search_results('[0-9] ', text)
```

Множества символов

```
text = """In September 1769 she reached New  
Zealand, the first European vessel to visit  
in 127 years."""
```

```
search_results('(i|I)n', text)
```

```
'In': 0-2
```

```
'in': 79-81
```

```
search_results('[0-9] ', text)
```

```
'9 ': 16-18
```

```
'7 ': 84-86
```

Экранирование символов

```
search_results('\\\\', '\\')
```

Экранирование символов

```
search_results('\\', '\\')
```

```
error: bogus escape (end of line)
```

Экранирование СИМВОЛОВ

```
search_results('\', '\')
```

```
error: bogus escape (end of line)
```

```
search_results('\\\\', '\\')
```

```
'\': 0-1
```

Экранирование СИМВОЛОВ

```
search_results('\\', '\\')
```

```
error: bogus escape (end of line)
```

```
search_results('\\\\', '\\')
```

```
'\': 0-1
```

```
search_results(r'\\', '\\')
```

```
'\': 0-1
```

Любой символ

```
text = """In September 1769 she reached New  
Zealand, the first European vessel to visit  
in 127 years."""
```

```
search_results('s.', text)
```


Любой СИМВОЛ

```
text = """In September 1769 she reached New  
Zealand, the first European vessel to visit  
in 127 years."""
```

```
search_results('s.', text)
```

```
'sh': 18-20
```

```
'st': 50-52
```

```
'ss': 64-66
```

```
'si': 74-76
```

```
's.': 90-92
```

Любой СИМВОЛ

```
text = """In September 1769 she reached New  
Zealand, the first European vessel to visit  
in 127 years."""
```

```
search_results('s.', text)
```

```
'sh': 18-20  
'st': 50-52  
'ss': 64-66  
'si': 74-76  
's.': 90-92
```

```
search_results(r's\.', text)
```

Любой СИМВОЛ

```
text = """In September 1769 she reached New  
Zealand, the first European vessel to visit  
in 127 years."""
```

```
search_results('s.', text)
```

```
'sh': 18-20  
'st': 50-52  
'ss': 64-66  
'si': 74-76  
's.': 90-92
```

```
search_results(r's\.', text)
```

```
's.': 90-92
```

Звезда и плюс Клини

```
text = """In September 1769 she reached New  
Zealand, the first European vessel to visit  
in 127 years."""
```

```
search_results('sh*e', text)
```

Звезда и плюс Клини

```
text = """In September 1769 she reached New  
Zealand, the first European vessel to visit  
in 127 years."""
```

```
search_results('sh*e', text)
```

```
'she': 18-21
```

```
'se': 65-67
```

Звезда и плюс Клини

```
text = """In September 1769 she reached New  
Zealand, the first European vessel to visit  
in 127 years."""
```

```
search_results('sh*e', text)
```

```
'she': 18-21
```

```
'se': 65-67
```

```
search_results('.s+.', text)
```

Звезда и плюс Клини

```
text = """In September 1769 she reached New  
Zealand, the first European vessel to visit  
in 127 years."""
```

```
search_results('sh*e', text)
```

```
'she': 18-21
```

```
'se': 65-67
```

```
search_results('.s+.', text)
```

```
' sh': 17-20
```

```
'rst': 49-52
```

```
'esse': 63-67
```

```
'isi': 73-76
```

```
'rs.': 89-92
```

Другие диапазоны

```
text = """In September 1769 she reached New  
Zealand, the first European vessel to visit  
in 127 years."""
```

```
search_results('i?n', text)
```


Другие диапазоны

```
text = """In September 1769 she reached New  
Zealand, the first European vessel to visit  
in 127 years."""
```

```
search_results('i?n', text)
```

```
'n': 1-2
```

```
'n': 39-40
```

```
'n': 60-61
```

```
'in': 79-81
```

Другие диапазоны

```
text = """In September 1769 she reached New  
Zealand, the first European vessel to visit  
in 127 years."""
```

```
search_results('i?n', text)
```

```
'n': 1-2  
'n': 39-40  
'n': 60-61  
'in': 79-81
```

```
search_results('[0-9]{4,6}', text)
```

Другие диапазоны

```
text = """In September 1769 she reached New  
Zealand, the first European vessel to visit  
in 127 years."""
```

```
search_results('i?n', text)
```

```
'n': 1-2  
'n': 39-40  
'n': 60-61  
'in': 79-81
```

```
search_results('[0-9]{4,6}', text)
```

```
'1769': 13-17
```

Жадность поиска

```
text = """In September 1769 she reached New  
Zealand, the first European vessel to visit  
in 127 years."""
```

```
search_results('e.*e', text)
```

Жадность поиска

```
text = """In September 1769 she reached New  
Zealand, the first European vessel to visit  
in 127 years."""
```

```
search_results('e.*e', text)
```

```
'eptember 1769 she reached Ne': 4-32
```

```
'ealand, the first European vesse': 35-67
```

Жадность поиска

```
text = """In September 1769 she reached New  
Zealand, the first European vessel to visit  
in 127 years."""
```

```
search_results('e.*e', text)
```

```
'eptember 1769 she reached Ne': 4-32
```

```
'ealand, the first European vesse': 35-67
```

```
search_results('e.*?e', text)
```

Жадность поиска

```
text = """In September 1769 she reached New Zealand, the first European vessel to visit in 127 years."""
```

```
search_results('e.*e', text)
```

```
'eptember 1769 she reached Ne': 4-32  
'ealand, the first European vesse': 35-67
```

```
search_results('e.*?e', text)
```

```
'epte': 4-8  
'er 1769 she': 10-21  
'each': 23-28  
'ealand, the': 35-46  
'ean ve': 58-64
```

Другие спецсимволы

```
text = """In September 1769 she reached New  
Zealand, the first European vessel to visit  
in 127 years."""
```

```
search_results(r'n\s+', text)
```


Другие спецсимволы

```
text = """In September 1769 she reached New  
Zealand, the first European vessel to visit  
in 127 years."""
```

```
search_results(r'n\s+', text)
```

```
'n ': 1-3
```

```
'n ': 60-62
```

```
'n ': 80-82
```

Другие спецсимволы

```
text = """In September 1769 she reached New  
Zealand, the first European vessel to visit  
in 127 years."""
```

```
search_results(r'n\s+', text)
```

```
'n ': 1-3
```

```
'n ': 60-62
```

```
'n ': 80-82
```

```
search_results(r't\S', text)
```

Другие спецсимволы

```
text = """In September 1769 she reached New Zealand, the first European vessel to visit in 127 years."""
```

```
search_results(r'n\s+', text)
```

```
'n ': 1-3  
'n ': 60-62  
'n ': 80-82
```

```
search_results(r't\S', text)
```

```
'te': 6-8  
'th': 43-45  
'to': 69-71
```

Группы

```
text = """In September 1769 she reached New  
Zealand, the first European vessel to visit  
in 127 years."""
```

```
m = re.search('(\w+)\s*([0-9]+)', text)  
print m.group(0)
```

```
September 1769
```

Группы

```
text = """In September 1769 she reached New Zealand, the first European vessel to visit in 127 years."""
```

```
m = re.search('(\w+)\s*([0-9]+)', text)
print m.group(0)
```

```
September 1769
```

```
print m.group(1)
print m.group(2)
```

```
September
1769
```

Другие возможности

Функции

- `match` — поиск только в первой позиции.
- `split` — разбиение текста на части.
- `sub` — замена в тексте.
- `compile` — компиляция выражения для многократного использования.

Другие возможности

Функции

- `match` — поиск только в первой позиции.
- `split` — разбиение текста на части.
- `sub` — замена в тексте.
- `compile` — компиляция выражения для многократного использования.

Флаги

- `IGNORECASE`
- `UNICODE`
- `VERBOSE`
- `MULTILINE`

Расширенная работа с аргументами

Аргументы функций

Обычная функция

```
def complex(real, imag):  
    return real + 1j * imag  
print complex(1, 2)
```

(1+2j)

Аргументы функций

Обычная функция

```
def complex(real, imag):  
    return real + 1j * imag  
print complex(1, 2)
```

(1+2j)

Аргументы по умолчанию

```
def complex(real, imag=0):  
    return real + 1j * imag  
print complex(1)
```

(1+0j)

Порядковые аргументы

Распаковка аргументов

```
def complex(*args):  
    return args[0] + 1j * args[1]  
complex(1, 2)
```

(1+2j)

Порядковые аргументы

Распаковка аргументов

```
def complex(*args):  
    return args[0] + 1j * args[1]  
complex(1, 2)
```

```
(1+2j)
```

Смешанные аргументы

```
def complex(real, *other):  
    print other  
    return real + 1j * other[0]  
print complex(1, 2, 3, 4)
```

```
(2, 3, 4)  
(1+2j)
```

Запаковка порядковых аргументов

Запаковка аргументов

```
def complex(real, imag):  
    return real + 1j * imag  
args = (1, 2)  
print complex(*args)
```

(1+2j)

Именованные аргументы

Передача аргументов по ключу

```
print complex(real=1, imag=2)  
print complex(3, imag=4)
```

```
(1+2j)  
(3+4j)
```

Именованные аргументы

Распаковка по ключу

```
def complex(**kwargs):  
    print kwargs  
    return (kwargs['real'] +  
            1j * kwargs['imag'])  
print complex(real=1, imag=2, hi=3)
```

```
{'real': 1, 'imag': 2, 'hi': 3}  
(1+2j)
```

Запаковка именованных аргументов

Запаковка словаря

```
kwargs = {'real': 1, 'imag': 2}  
print complex(**kwargs)
```

```
{'real': 1, 'imag': 2}  
(1+2j)
```


Функция с любыми аргументами

Декораторы

Определение

Декоратор

- Функция, модифицирующая другую функцию
- Используется для упрощения кода

ИСПОЛЬЗОВАНИЕ

```
def notify_about_calls(func):  
    def decorated(*args, **kwargs):  
        print "Called", func.__name__  
        return func(*args, **kwargs)  
    return decorated
```

ИСПОЛЬЗОВАНИЕ

```
def notify_about_calls(func):  
    def decorated(*args, **kwargs):  
        print "Called", func.__name__  
        return func(*args, **kwargs)  
    return decorated  
  
def f(a, b):  
    return a + b  
g = notify_about_calls(f)  
print g(1, 2)
```

ИСПОЛЬЗОВАНИЕ

```
def notify_about_calls(func):  
    def decorated(*args, **kwargs):  
        print "Called", func.__name__  
        return func(*args, **kwargs)  
    return decorated  
  
def f(a, b):  
    return a + b  
g = notify_about_calls(f)  
print g(1, 2)
```

```
Called f  
3
```

Упрощенная запись

```
@decorate  
def f(a, b):  
    return a + b
```

Эквивалентно

```
def f(a, b):  
    return a + b  
f = decorate(f)
```

Несколько декораторов

```
@decorator1
@decorator2
def f(a, b):
    return a + b
```

Эквивалентно

```
def f(a, b):
    return a + b
f = decorator1(decorator2(f))
```


Классы-декораторы

```
class Logger(object):  
    def __init__(self, func):  
        self.func = func  
        self.log = []  
    def __call__(self, *args, **kwargs):  
        self.log.append((args, kwargs))  
        return self.func(*args, **kwargs)
```

```
logged = Logger
```

Классы-декораторы

```
@logged  
def f(x, y=0):  
    pass
```

```
f(1)  
f(1, y=2)  
print f.log
```

```
[((1,), {}), ((1,), {'y': 2})]
```

Метаданные функций

```
def notify_about_calls(func):  
    def decorated(*args, **kwargs):  
        print "Called", func.__name__  
        return func(*args, **kwargs)  
    return decorated
```

Метаданные функций

```
@notify_about_calls  
def some_function(x, y):  
    """Docstring."""  
    return x + 2 * y
```

Метаданные функций

```
@notify_about_calls
def some_function(x, y):
    """Docstring."""
    return x + 2 * y

some_function(1, 2)
print some_function.__name__
print some_function.__doc__
```

```
Called some_function
decorated
None
```

functools.wraps

```
import functools

def notify_about_calls(func):
    @functools.wraps(func)
    def decorated(*args, **kwargs):
        print "Called", func.__name__
        return func(*args, **kwargs)
    return decorated
```

functools.wraps

```
@notify_about_calls
def some_function(x, y):
    """Docstring."""
    return x + 2 * y
```

functools.wraps

```
@notify_about_calls
def some_function(x, y):
    """Docstring."""
    return x + 2 * y

some_function(1, 2)
print some_function.__name__
print some_function.__doc__
```

```
Called some_function
some_function
Docstring.
```


Декоратор `staticmethod`

`staticmethod`

- Применяется к методу класса
- Делает метод статическим
- Позволяет игнорировать экземпляр (`self`)

Декоратор `staticmethod`

```
class A(object):  
    @staticmethod  
    def f(a, b):  
        return a + b
```

```
a = A()  
print a.f(1, 2)  
print A.f(1, 2)
```

3

3

Декораторы с аргументами

Как сделать такой декоратор?

```
@check_return_type(float)
def calculate_something(a, b, c):
    ...
    return x
```

Декораторы с аргументами

Как сделать такой декоратор?

```
@check_return_type(float)
def calculate_something(a, b, c):
    ...
    return x

def check_return_type(type_):
    def decorator(func):
        def decorated(*args, **kwargs):
            val = func(*args, **kwargs)
            assert isinstance(val, type_)
            return val
        return decorated
    return decorator
```