

# Python

## #4



Системное программирование



# Feedback



**ТЫ МОЛОДЕЦ!!!**

не сомневайся





# генераторы



# Кодировки... мда...



Юникод для чайников - <http://habrahabr.ru/post/135913/>





**WTF !?**

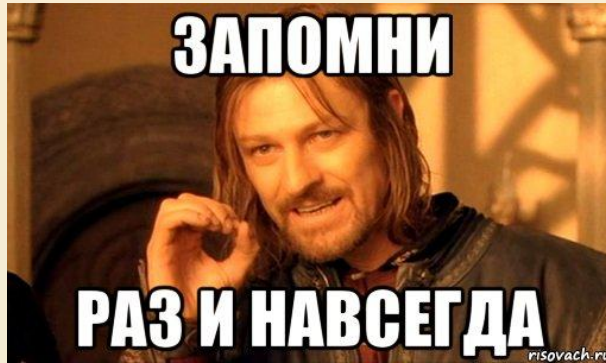
ЮНИКОД == СИМВОЛЫ  
строка == байты

байты -> что-то значащее (символы) —  
это де-кодирование (*decode*),

символы -> байты — кодирование (*encode*)



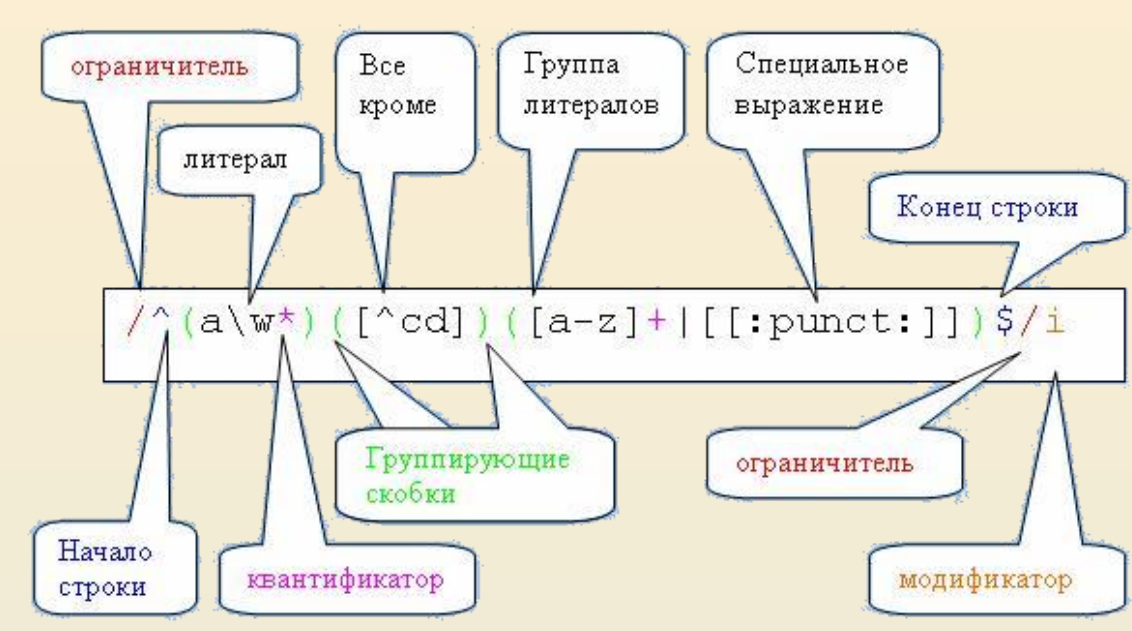
«**UnicodeDecodeError**» —> нужно декодировать строку в юникод, используя правильную кодировку



«**UnicodeEncodeError**» —> нужно указать правильную кодировку превращения юникод-строки в обычную (или использовать параметр 'ignore'\ 'replace'\ 'xmlcharrefreplace' в методе «encode»)



# Регулярные выражения



Фридлл Дж. – «Регулярные выражения»

Фитцджеральд М. – «Регулярные выражения, основы»

Вебинар «Регулярные выражения для новичков и не только» - <http://geekbrains.ru/events/19>

Регулярные выражения, пособие для новичков. Часть 1 - <http://habrahabr.ru/post/115825/>





# Регулярные выражения

## Якоря

<code>^</code>	Начало строки +
<code>\A</code>	Начало текста +
<code>\$</code>	Конец строки +
<code>\Z</code>	Конец текста +
<code>\b</code>	Граница слова +
<code>\B</code>	Не граница слова +
<code>&lt;</code>	Начало слова
<code>&gt;</code>	Конец слова

## Символьные классы

<code>\c</code>	Управляющий символ
<code>\s</code>	Пробел
<code>\S</code>	Не пробел
<code>\d</code>	Цифра
<code>\D</code>	Не цифра
<code>\w</code>	Слово
<code>\W</code>	Не слово
<code>\xhh</code>	Шестнадцатиричный символ hh
<code>\Oxxx</code>	Восьмиричный символ xxx

## Символьные классы POSIX

<code>[upper:]</code>	Буквы в верхнем регистре
<code>[lower:]</code>	Буквы в нижнем регистре
<code>[alpha:]</code>	Все буквы
<code>[alnum:]</code>	Буквы и цифры
<code>[digit:]</code>	Цифры

## Образцы шаблонов

<code>([A-Za-z0-9-]+)</code>	Буквы, числа и знаки переноса
<code>(\d{1,2}\V\d{1,2}\V\d{4})</code>	Дата (напр., 21/3/2006)
<code>([^\s]+(?:\.(jpg gif png))\.\2)</code>	Имя файла jpg, gif или png
<code>(^[1-9]{1}\$ ^[1-4]{1}[0-9]{1}\$ ^50\$)</code>	Любое число от 1 до 50 включительно
<code>(#?([A-Fa-f0-9]){3}((([A-Fa-f0-9]){3})?))</code>	Шестнадцатиричный код цвета
<code>((?=\d)(?=[a-z])(?=[A-Z]).{8,15})</code>	От 8 до 15 символов с минимум одной цифрой, одной заглавной и одной строчной буквой (полезно для паролей).
<code>(\w+@[a-zA-Z_]+?\.[a-zA-Z]{2,6})</code>	Адрес email
<code>(\&lt;/?[^\&gt;]+\&gt;)</code>	HTML теги

## Примечание

Эти шаблоны предназначены для ознакомительных целей и основательно не проверялись. Используйте их с осторожностью и предварительно тестируйте.

## Кванторы

<code>*</code>	0 или больше +
<code>*?</code>	0 или больше, нежадный +
<code>+</code>	1 или больше +
<code>+?</code>	1 или больше, нежадный +
<code>?</code>	0 или 1 +
<code>??</code>	0 или 1, нежадный +
<code>{3}</code>	Ровно 3 +
<code>{3,}</code>	3 или больше +
<code>{3,5}</code>	3, 4 или 5 +
<code>{3,5}?</code>	3, 4 или 5, нежадный +

## Диапазоны

<code>.</code>	Любой символ, кроме переноса строки (\n) +
<code>(a b)</code>	a или b +
<code>(...)</code>	Группа +
<code>(?:...)</code>	Пассивная группа +
<code>[abc]</code>	Диапазон (a или b или c) +
<code>[^abc]</code>	Не a, не b и не c +
<code>[a-q]</code>	Буква между a и q +
<code>[A-Q]</code>	Буква в верхнем регистре между A и Q +
<code>[0-7]</code>	Цифра между 0 и 7 +

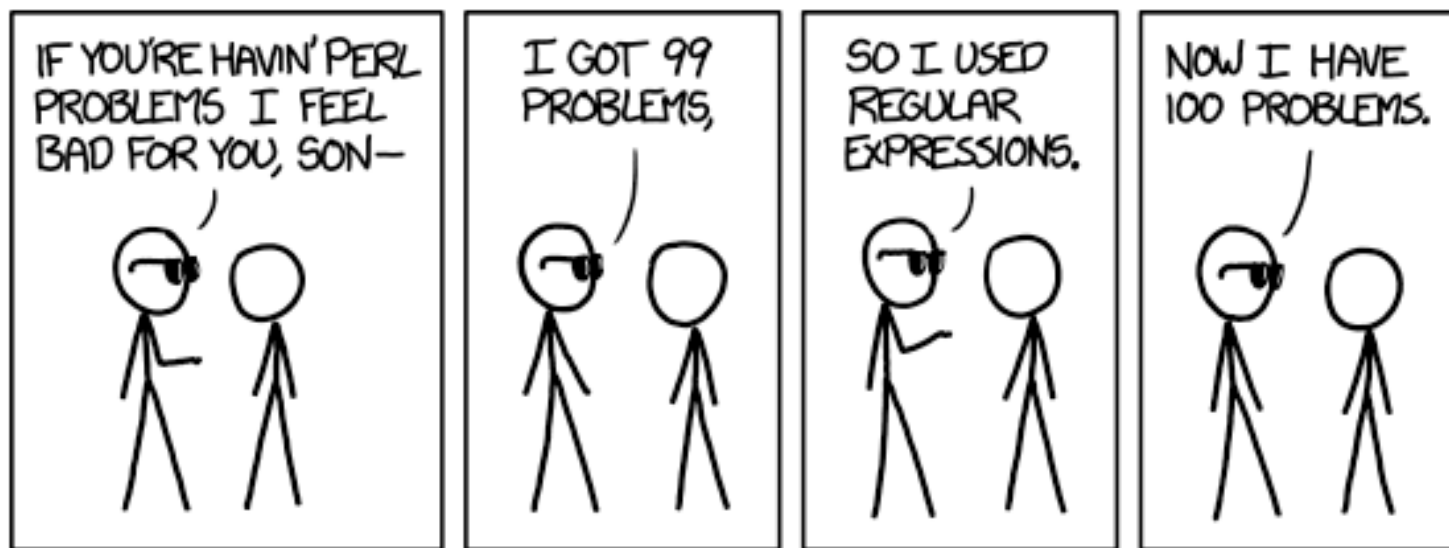




# Регулярные выражения



# Регулярные выражения



# Взаимодействие с ОС

`os`

`os.path`

`shutil`



`os.system`

`sys`

`subprocess`



# Работа с файлами

```
os.getcwd()
```

```
os.chdir('dir')
```

```
os.mkdir('dir')
```

```
os.makedirs('dir1/dir2')
```

```
os.rename('file1', 'file2')
```

```
shutil.copy('file1', 'file2')
```

```
shutil.rmtree('dir')
```

```
os.path.join('dir1', 'dir2', 'file')
```

```
os.path.basename('filename')
```



# Работа с сетью

Python 2.x

`urllib`

`urllib2`

Python 3.x

`urllib`

`webbrowser`

`ftplib`

`poplib`

`smtplib`

`socket`

`ssl`

`asyncore`

`asynchat`



# urllib, urllib2

```
import urllib, urllib2
page = urllib2.urlopen('http://url')
#urllib.request.urlopen()      -    для Python 3.x

html = page.read()              # читаем страницу

args = urllib.urlencode(args_dict)
urllib2.urlopen('url' + '?' + args) # GET-запрос

urllib2.urlopen('url', args)      # POST-запрос

# для передачи дополнительных заголовков
req = urllib2.Request(url, данные, заголовки)
```



# Домашка



**CHUCK NORRIS DOESN'T WRITE CODE**

He stares at a computer screen until he gets the program he wants.





Катастрофа Unicode в Python3 - <http://habrahabr.ru/post/208192/>

[http://local.joelonsoftware.com/wiki/Абсолютный Минимум, который Каждый Разработчик Программного Обеспечения Обязательно Должен Знать о Unicode и Наборах Символов](http://local.joelonsoftware.com/wiki/Абсолютный_Минимум,_который_Каждый_Разработчик_Программного_Обеспечения_Обязательно_Должен_Знать_о_Unicode_и_Наборах_Символов)

Сокеты в Python для начинающих - <http://habrahabr.ru/post/149077/>

