

ОснОвы Python

Синтаксис, стиль, философия...

- zen, pep8
- КОНСОЛЬ, ТИПЫ ДАННЫХ, ОПЕРАТОРЫ
- terminal, tools, IDE
- исключения и разбор стектрейса
- синтаксис управляющих конструкций

Философия прежде всего!

- код мы чаще читаем чем пишем
- лаконичность синтаксиса и минимум мусора
- пробелы имеют значение!
- строгие гайды форматирования pep8
- не синтаксисом и форматированием достигается zen

import this

Beautiful is better than ugly.

Explicit is better than implicit.

Simple is better than complex.

Complex is better than complicated.

Flat is better than nested.

Sparse is better than dense.

Readability counts.

Special cases aren't special enough to break the rules.

Although practicality beats purity.

Errors should never pass silently.

Unless explicitly silenced.

...

- PEP8 Стандарт форматирования
- PEP20 The Zen of Python
- Официальный туториал
- Документация

Hello world!

- print it!
- СВОЯ КОНСОЛЬ, запуск из коноли
- типизация - динамическая, но строгая
- встроенные простые типы данных:
 - int, str, bool
- преобразование типов
- встроенные сложные типы данных:
 - list, tuple, dict
- встроенные функции
- функции типов данных

int, float

- МОЖНО И НУЖНО ПОЛЬЗОВАТЬСЯ КАК КАЛЬКУЛЯТОРОМ
- операторы: + - * / // % **
- round, ...
- группировка действий ()

str

- ' ' " " ''' ''' """ """

- операторы: + * %
- строка это итерируемый тип
- функции:
 - title, capitalize, upper, lower
 - strip, lstrip, rstrip
 - count, index
 - isalnum, isalpha, isdigit

bool

- True, False
- булевые операторы:
> >= == != <= <
and or is not

Преобразование типов

- ТИПЫ ДАННЫХ

`int()`

`float()`

`str()`

- преобразование типов данных
- преобразование к булевому значению

builtin functions

- `__builtin__`
- `dir(__builtin__)`
- **ТИПЫ ДАННЫХ** (int, bool, str, list, tuple, dict)
- **некоторые функции** (len, dir, type, ord, chr)

list

- []
- операторы: + *
- индексы
- слайсы
- функции
 - index, count
 - sort, reverse
 - append, extend, insert, pop, remove
- функции строк split, join
- range, min, max, sum, sorted

tuple

- ()
- операторы: + *
- индексы
- слайсы
- функции
 - index, count

dict

- { }
- особенности: не сортированный, изменяемый
- ключ-значение, взятие значения по индексу
- преобразования
- функции
 - keys, values, items
 - pop, popitem, update
 - get, has_key (operator 'in')

execute file

- bash terminal (навигация, по директориям, создание файлов и директорий)
- редактирование файла в IDE
- альтернативные интерактивные консоли

Exceptions!

- наши лучшие друзья
- читаем трейс ошибки
- ЛОВИМ ОШИБКИ

if ... elif ... else

- ВЫЧИСЛЯЕТ логическое значение
- ВЫПОЛНЯЕТ соответствующий блок (отступы имеют значение!)

```
if True:  
    print "True!"  
else:  
    print "False!"
```


for ... in ...

- проходит по всем значениям последовательности
- выполняет для каждого тело цикла

```
for i in (1, 2, 3):  
    t = i**2  
    print t
```

- break, continue, else, pass

while ...

- проверяет логическое условие
- выполняет тело цикла пока выполняется условие

```
while True:  
    print "Выводим это сообщение бесконечно"
```

- break, continue, else

try ... except ...

- просто идет выполнение первого блока
- если происходит исключение описанное в except прерывает выполнение и переходит к блоку except

```
try:  
    a = 10 / 0  
except ZeroDivisionError:  
    print u"Деление на ноль!"
```

- else, finally

интересности

- распаковка
- генераторы списков
- unicode
- `# -*- coding: utf-8 -*-`