

# Лекция 1

## Основы Python

12 февраля 2015 г.

# О языке Python

# История

Автор: Гвидо ван Россум.  
(Сейчас он является BDFL — Benevolent Dictator For Life)

1994г. — Python 1.0

2000г. — Python 2.0

2008г. — Python 3.0

# Особенности

- Понятность кода
  - Легко писать
  - Легко отлаживать
  - Легко читать и модифицировать

# Особенности

- Понятность кода
  - Легко писать
  - Легко отлаживать
  - Легко читать и модифицировать
- Низкая эффективность
  - *Чистый* Python 5-100 раз медленнее C++

# Версии

## Python 2.x

- 2.5
- 2.6
- **2.7**

## Python 3.x (несовместим с 2.x)

- 3.2
- 3.3
- 3.4

# Работа с интерпретатором

# Обычный режим работы

```
$ python main.py
```

Интерпретатор + исходный код.

Запускается новый процесс с программой.



# Интерактивный режим

```
$ python
```

Интерпретатор без кода.

Исполнение кода в онлайн-режиме.

# Интерактивный режим

```
$ python
```

Интерпретатор без кода.

Исполнение кода в онлайн-режиме.

Выход: Ctrl+D (Unix) или Ctrl+Z+Enter (Windows).

Помощь: `help(X)`, где `X` — то, по чему нужна помощь.

Выход из помощи: `q`.

# Основные типы данных

# Python как калькулятор

```
>>> 2 + 2
```

```
4
```

```
>>> 50 - 5*6
```

```
20
```

```
>>> 8 / 5.0
```

```
1.6
```

```
>>> 5 ** 2
```

```
25
```

# Деление

```
>>> 17 / 3
```

```
5
```

```
>>> 17 / 3.0
```

```
5.666666666666667
```

```
>>> 17 // 3.0
```

```
5.0
```

```
>>> 17 % 3
```

```
2
```

# Логические выражения

```
>>> 1 == 1
```

```
True
```

```
>>> 2 * 2 != 4
```

```
False
```

```
>>> False or 5 > 2 and True
```

```
True
```

```
>>> not 1 < 2 < 10
```

```
False
```

# Переменные

```
>>> x = 1
```

```
>>> y = 2
```

```
>>> x * y
```

```
2
```

# Переменные

```
>>> x = 1  
>>> y = 2  
>>> x * y  
2
```

```
>>> x  
1  
>>> x = 1.0  
>>> x  
1.0
```

Динамическая типизация. Жесткая типизация.



# Изменение “на месте”

```
>>> x = 1
```

```
>>> x += 2
```

```
>>> x
```

```
3
```

```
>>> x /= 2.0
```

```
>>> x
```

```
1.5
```

# Строки

```
>>> s = 'hello, world'
```

```
>>> s
```

```
'hello, world'
```

# Строки

```
>>> s = 'hello, world'
```

```
>>> s
```

```
'hello, world'
```

```
>>> s + '!'
```

```
'hello, world!'
```

```
>>> s[0]
```

```
'h'
```

```
>>> s[1:6]
```

```
'ello,'
```

```
>>> len(s)
```

```
12
```

# Вывод на экран

```
'Hello, world!'
```

```
print 'Hello, world!'
```

```
Hello, world!
```

# Вывод на экран

```
'Hello, world!'
```

```
print 'Hello, world!'
```

```
Hello, world!
```

```
print '1 + 1 =', 2
```

```
1 + 1 = 2
```

# Переносы строк

```
print 'first line'  
print  
print 'third line'
```

```
first line
```

```
third line
```

# Переносы строк

```
print 'first line'  
print  
print 'third line'
```

```
first line
```

```
third line
```

```
print 'first\nsecond'
```

```
first  
second
```

# Списки

```
>>> l = [1, 2, 3]
```

```
>>> l
```

```
[1, 2, 3]
```



# Списки

```
>>> l = [1, 2, 3]
```

```
>>> l
```

```
[1, 2, 3]
```

```
>>> l[0]
```

```
1
```

```
>>> l[-2:]
```

```
[2, 3]
```

```
>>> l + [4, 5]
```

```
[1, 2, 3, 4, 5]
```

```
>>> len(l)
```

```
5
```

# Изменение списков

```
>>> l = [1, 2, -1, 4, 5]
```

```
>>> l
```

```
[1, 2, -1, 4, 5]
```

```
>>> l[2] = 3
```

```
>>> l
```

```
[1, 2, 3, 4, 5]
```

```
>>> l.append(6)
```

```
>>> l
```

```
[1, 2, 3, 4, 5, 6]
```

## Другие операции

```
>>> l = [1, 2, 3, 4]
```

```
>>> l[1:3] = [0, 0]
```

```
>>> l
```

```
[1, 0, 0, 4]
```

```
>>> l[1:3] = [-1]
```

```
>>> l
```

```
[1, -1, 4]
```

## Другие операции

```
>>> l = [1, 2, 3, 4]
```

```
>>> l[1:3] = [0, 0]
```

```
>>> l
```

```
[1, 0, 0, 4]
```

```
>>> l[1:3] = [-1]
```

```
>>> l
```

```
[1, -1, 4]
```

```
>>> len([])
```

```
0
```

```
>>> l[1:2] = []
```

```
>>> l
```

```
[1, 4]
```

# Составные операторы

# Условный оператор if

```
if x < 0:
    sign = -1
    print "Negative"
elif x == 0:
    sign = 0
    print "Zero"
else:
    sign = 1
    print "Positive"
```

# Названия типов данных

Числовые типы: `int`, `float`.

Строка: `str`.

Список: `list`.

# Комментарии

```
# this is a comment  
x = 1  
# x += 1  
print x
```

```
1
```