

Delta-3 Intel

Нижний Новгород, 2015



# Скриптовые языки программирования

Алексей Бухнин, Intel



# «Сценарий»

Программа командной оболочки

Связующий слой

Быстрое решение задач

# Примеры языков сценариев

Python

Ruby

Lua

JavaScript

PHP

Perl

Tcl

Shell scripts: Bash, Batch

# Отличительные особенности

Интерпретируемость

Динамическая типизация

«Сборщик мусора»

Отражение (reflection) и  
метапрограммирование

# Python

высокоуровневый  
интерпретируемый  
объектно-ориентированный  
многоцелевой язык программирования,  
который зачастую используется  
для создания сценариев

# Положительные стороны Python

Качество программного обеспечения

Высокая скорость разработки

Переносимость программ

Библиотеки поддержки

Механизмы интеграции компонентов

Он объектно-ориентированный

Он бесплатный и открытый

# Недостатки Python

Производительность ниже, чем у компилирующихся языков

Ограничение распараллеливания из-за GIL

Проблемы совместимости версий 2.x и 3.x

# Области применения Python

Системное программирование

Интеграция компонентов

Веб-приложения

Приложения баз данных

Быстрое создание прототипов

Программирование научных вычислений

Графический интерфейс



# Преимущества Python перед другими языками

Более широкие возможности, чем у Tcl

Синтаксис и архитектура проще, чем у Perl

Проще и удобнее, чем Java и C++

Более переносимый, чем Visual Basic

Более универсальный, чем PHP

Более ясный синтаксис, чем у Ruby

# ОСНОВЫ синтаксиса Python

Блоки инструкций выделяются отступами

Конец строки завершает инструкцию

Инструкции на одной строке разделяются ;

После объявления функции, класса,  
условного оператора или цикла ставится :

```
for item in items:  
    item += 1; print(item)
```

# Интерактивный режим

```
C:\>c:\Python27\python
```

```
ActivePython 2.7.2.5 (ActiveState Software Inc.)
```

```
based on
```

```
Python 2.7.2 (default, Jun 24 2011, 12:22:14)
```

```
[MSC v.1500 64 bit (AMD64)] on win32
```

```
Type "help", "copyright", "credits" or "license"  
for more information.
```

```
>>>
```

```
>>> print('Hello world!')
```

```
Hello world!
```

```
>>> print(2 ** 8)
```

```
256
```

# Выполнение программы

Сценарий `script0.py`:

```
print('Hello world')
```

```
print(2 ** 100)
```

```
C:\temp> python script0.py
```

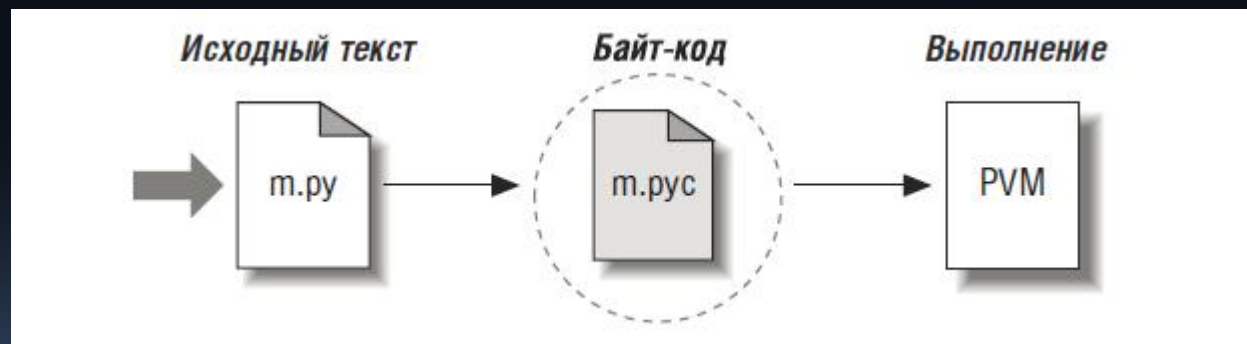
```
Hello world
```

```
1267650600228229401496703205376
```

# С точки зрения Python

Компиляция в байт-код

Выполнение в виртуальной машине Python



# Реализации интерпретатора

CPython

Jython - JVM

IronPython - .NET

PyPy

# Встроенные типы объектов

Числа

Кортежи

Строки

Списки

Словари

Множества

# Числа

Обычные целые числа `int('5')`, `1`, `5`

**Целые неограниченной длины** `long('1')`, `1L`, `11`

Восьмеричные числа `020`

Шестнадцатеричные числа `0xA`, `0xa`

Булевы значения `True`, `False`

Числа с плавающей точкой `float('1.4')`, `1.4`, `1e7`

Комплексные числа `complex(1, 2)`, `5j`, `5J`, `3+4j`  
`z.real`, `z.imag`



# Кортежи (tuple)

Ряд Фибоначчи

```
a, b = 0, 1
```

```
while 1:
```

```
    print b,
```

```
    a, b = b, a + b
```

```
1 1 2 3 5 8
```

```
Traceback (most recent call last):
```

```
  File "<stdin>", line 2, in <module>
```

```
KeyboardInterrupt
```

# Строки

```
>>> 'r' in 'string'
```

```
True
```

```
>>> "string".replace('r', '')
```

```
'sting'
```

```
>>> 'string'.find('ri')
```

```
2
```

```
>>> 'a,b,c'.split(',')
```

```
['a', 'b', 'c']
```

# Форматирование строк: format

```
>>> 'Hello {}'.format('world')
```

```
'Hello world!'
```

```
>>> '{} {}'.format(17, 'students')
```

```
'17 students'
```

```
>>> '{1} {0}'.format('students', 17)
```

```
'17 students'
```

# СПИСКИ

## Генераторы списков

```
>>> list1 = [x * 2 for x in 'spam']
```

```
>>> list1
```

```
['ss', 'pp', 'aa', 'mm']
```

```
>>> list2 = [x + '!' for x in list1
```

```
... if x != 'aa']
```

```
>>> list2
```

```
['ss!', 'pp!', 'mm!']
```

# Словари

```
user = {  
    'name': u'Олег',  
    'email': 'oleg@example.com',  
    'address': {  
        'city': u'Москва',  
        'street': u'Тверская'  
    },  
    'hobby': [u'рисование', u'пение']  
}
```