

# Язык программирования PYTHON

Скоблина Юлия Игоревна

# История PYTHON

Язык программирования PYTHON был создан примерно в 1991 году голландцем Гвидо ван Россумом.

Свое имя – Пайтон (или Питон) – получил от названия телесериала, а не пресмыкающегося.

После того, как Россум разработал язык, он выложил его в Интернет, где уже целое сообщество программистов присоединилось к его улучшению.

Python активно совершенствуется и в настоящее время. Часто выходят его новые версии. Официальный сайт <http://python.org>.

## *Дзен Питона*

Если интерпретатору Питона дать команду `import this` (импортировать "сам объект"),

то выведется так называемый "Дзен Питона", иллюстрирующий идеологию и особенности данного языка. Глубокое понимание этого дзена приходит тем, кто сможет освоить язык Python в полной мере и приобретет опыт практического программирования.

# ДЗЭН PYTHON (философия)

- *Красивое лучше, чем уродливое.*
- *Явное лучше, чем неявное.*
- *Простое лучше, чем сложное.*
- *Сложное лучше, чем запутанное.*
- *Плоское лучше, чем вложенное.*
- *Разреженное лучше, чем плотное.*
- *Читаемость имеет значение.*
- *Особые случаи не настолько особые, чтобы нарушать правила.*
- *Должен существовать один — и, желательно, только один — очевидный способ сделать это.*
- *Если реализацию сложно объяснить — идея плоха.*

- **Python** – высокоуровневый язык программирования общего назначения с акцентом на производительность разработчика и читаемость кода
- Python и подавляющее большинство библиотек к нему бесплатны и поставляются в исходных кодах. Более того, в отличие от многих открытых систем, лицензия никак не ограничивает использование Python в коммерческих разработках
- Python имеет ясный синтаксис. В нем сведены к минимуму такие вспомогательные конструкции как скобки, слова-организаторы\_блоков. Взамен программист обязан четко соблюдать отступы, которые и являются организаторами блоков. В результате код получается незагруженным лишними элементами и легко читаемым.

# Преимущества Python

Скорость выполнения программ написанных на Python очень высока. Это связано с тем, что основные библиотеки Python написаны на C++ и выполнение задач занимает меньше времени, чем на других языках высокого уровня.

В связи с этим вы можете писать свои собственные модули для Python на C или C++

В стандартных библиотеках Python вы можете найти средства для работы с электронной почтой, протоколами Интернета, FTP, HTTP, базами данных, и пр.

Скрипты, написанные при помощи Python выполняются на большинстве современных ОС. Такая переносимость обеспечивает Python применение в самых различных областях.

Python подходит для любых решений в области программирования, будь то офисные программы, веб-приложения, GUI-приложения и т.д.

Над разработкой Python трудились тысячи энтузиастов со всего мира. Поддержкой современных технологий в стандартных библиотеках мы можем быть обязаны именно тому, что Python был открыт для всех желающих.

# Недостатки PYTHON

- PYTHON, как и другие интерпретируемые языки, имеет сравнительно невысокую скорость выполнения программ. Однако, в случае с Python этот недостаток компенсируется уменьшением времени разработки программы. В среднем, программа на Python в 2-4 раза компактнее, чем её аналог на C++ или Java
- Динамическая типизация вызывает вопросы у методистов по обучению программированию
- Учебники, ЕГЭ и олимпиады

# Интерактивный режим

В основном интерпретатор выполняет команды построчно: пишешь строку, нажимаешь Enter, интерпретатор выполняет ее, наблюдаешь результат.

Возможности языка позволяют использовать его как калькулятор, не зная команд программирования.

$$2 + 5$$

$$3 * (5 - 8)$$

$$2.4 + 3.0 / 2$$

и т.д.

Наберите подобные примеры в интерактивном режиме (в конце каждого нажимайте Enter). Ответ выдается сразу после нажатия.



# Синтаксис

Во первых стоит отметить интересную особенность Python. Он не содержит операторных скобок (begin..end в pascal или {..}в Си), вместо этого блоки выделяются отступами: пробелами или табуляцией, а вход в блок из операторов осуществляется двоеточием. Однострочные комментарии начинаются со знака фунта «#», многострочные — начинаются и заканчиваются тремя двойными кавычками «"""».

Чтобы присвоить значение переменной используется знак «=», а для сравнения — «==». Для увеличения значения переменной, или добавления к строке используется оператор «+=», а для уменьшения — «-=». Все эти операции могут взаимодействовать с большинством типов, в том числе со строками.

# Оператор присваивания

- `A = 3.14`  
`print(type(A)) #float(дробное число)`  
`A = 'Hello'`  
`print(type(A)) #str(преобразует переданный ей аргумент в строку)`
- `a = b = c = 0`  
`a += 1 # a = a + 1`
- `c = 5//2 #int(преобразует данную строку в целое)`
- `d = 5/2 #float`  
`b = c**2 # b = c2`
- `a, b = b, a`

# Первая программа

- На PYTHON  
`print("Hello, World!")`
- На C++  
`#include <iostream.h>`  
`void main()`  
`{`  
 `cout << "Hello, World!" << endl;`  
`}`

# Условный оператор

- `if 0 < x < 2:`  
    `if 0 < y < 2:`  
`else:`  
    оператор

- `if x < 0:`  
    блок1  
`elif x == 0:`  
    блок2  
`else:`  
    блок3

Пример кода с веткой else на языке программирования Python:

```
print "Привет"  
tovar1 = 50  
tovar2 = 32  
if tovar1+ tovar2 > 99 :  
    print "Сумма не достаточна"  
else:  
    print "Чек оплачен"  
print "Пока"
```

# Сравнение цикла «пока»

## Pascal

```
a := 10;  
b := 1;  
WHILE (a > 0) do  
  Begin  
    b = b * 2;  
    a = a - 1;  
  End;
```

## Python

```
a = 10  
b = 1  
while a > 0:  
    b = b * 2  
    a = a - 1
```

# Цикл FOR

Например, если имеется список чисел, и необходимо увеличить значение каждого элемента на две единицы, то можно перебрать список с помощью цикла for, выполнив над каждым его элементом соответствующее действие.

```
>>> spisok = [0,10,20,30,40,50,60,70,80,90]
>>> i = 0
>>> for element in spisok:
    spisok[i] = element + 2
    i = i + 1
>>> spisok
[2, 12, 22, 32, 42, 52, 62, 72, 82, 92]
```

В примере переменная *i* нужна для того, чтобы записать изменившееся значение элемента в список.

Это вводная презентация была посвящена знакомству с Python, его возможностям и основам синтаксиса. Как и говорилось, изучение Python в качестве второго языка программирования не представляет особой сложности, так как у него больше общих черт с другими языками, нежели различий, хотя отдельные особенности и являются специфическими чертами Python (использование отступов и т.д.).



Полезные ссылки

[informatics.mccme.  
ru](http://informatics.mccme.ru)

**Спасибо за внимание!**