

# Python



# Download

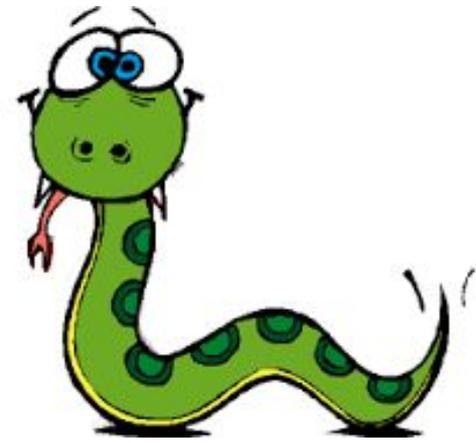
- *python.org*

- *Курс по Питону:*

- *[http://courses.prometheus.org.ua/courses/KPI/Programming101/2015\\_T1/about](http://courses.prometheus.org.ua/courses/KPI/Programming101/2015_T1/about)*

# Почему Python?

- Качество ПО
- Библиотеки поддержки
- Переносимость программ
- Скорость разработки



# + /- Python

- **Python - интерпретируемый язык программирования.** С одной стороны, это позволяет значительно упростить отладку программ, с другой - обуславливает сравнительно низкую скорость выполнения.
- **Динамическая типизация.** В python не надо заранее объявлять тип переменной, что очень удобно при разработке.
- **Хорошая поддержка модульности.** Вы можете легко написать свой модуль и использовать его в других программах.
- Встроенная поддержка Unicode в строках.
- Поддержка объектно-ориентированного программирования. При этом его реализация в python является одной из самых понятных.
- Автоматическая сборка мусора, отсутствие утечек памяти.

# + /- Python

- Интеграция с C/C++, если возможностей Python недостаточно.
- Понятный и лаконичный синтаксис Понятный и лаконичный синтаксис, способствующий ясному отображению кода. Огромное количество модулей, как входящих в стандартную поставку Python 3, так и сторонних. В некоторых случаях для написания программы достаточно лишь найти подходящие модули и правильно их скомбинировать.
- **Кроссплатформенность.** Программа, написанная на Python, будет функционировать совершенно одинаково вне зависимости от того, в какой операционной системе она запущена.

# 1. Где используется Python?

- Компания **Google** использует **Python** в своей поисковой системе и оплачивает труд создателя **Python** — [Гвидо ван Россума](#)
- **Intel, Cisco, Hewlett-Packard, Seagate, Qualcomm** и **IBM**, используют **Python** для тестирования аппаратного обеспечения
- **YouTube** в значительной степени реализована на **Python**
- **NSA** (National Security Agency ) использует **Python** для шифрования и анализа разведданных

## 2. Где используется Python?

- Компании **JPMorgan Chase, UBS, Getco** и **Citadel** применяют **Python** для прогнозирования финансового рынка
- Популярный веб-фреймворк **App Engine** от компании **Google** использует **Python** в качестве прикладного языка программирования
- Популярная программа **BitTorrent** для обмена файлами в пиринговых сетях написана на языке **Python**
- **NASA, Los Alamos, JPL** и **Fermilab** используют **Python** для научных вычислений.

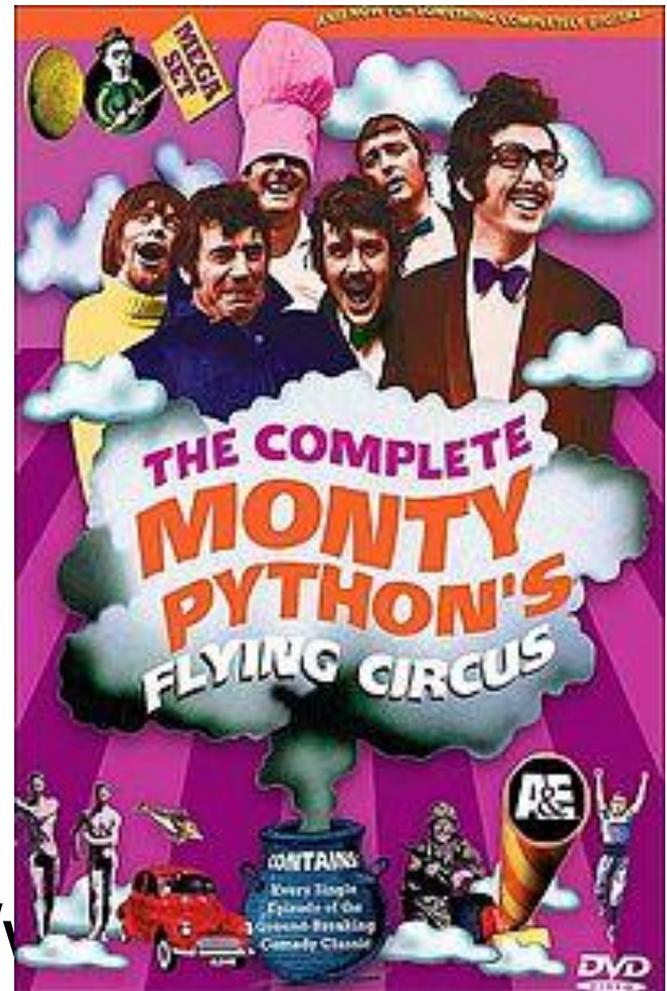
# Философия

- Красивое лучше, чем уродливое.
- Явное лучше, чем неявное.
- Простое лучше, чем сложное.
- Сложное лучше, чем запутанное.
- Особые случаи не настолько особые, чтобы нарушать правила.
- При этом практичность важнее безупречности.
- Ошибки никогда не должны замалчиваться.
- Если не замалчиваются явно.
- Встретив двусмысленность, отбрось искушение угадать.
- Должен существовать один — и, желательно, *только* один — очевидный способ сделать это.
- Хотя он поначалу может быть и не очевиден, если вы не голландец.
- Сейчас лучше, чем никогда.
- Хотя никогда зачастую лучше, чем *прямо* сейчас.
- Если реализацию сложно объяснить — идея плоха.
- Если реализацию легко объяснить — идея, *возможно*, хороша.

# В честь кого назвали?



- <https://www.youtube.com/MPT5RE>



# Гвидо ван Россум



# Арифметические операции

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

```
>>> 2 * 100
```

```
>>> 2 ** 10
```

```
>>> 'Hello' + 'World'
```

```
>>> 'ABC' * 100
```

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

# Строки

```
□ "This is a string"  
□ 'This is also a string'  
□ print "Output Number One"  
□ print 'Output Number Two'  
□ print "Hello"  
□ print  
□ print "One", "Two"  
□ print "One", "Two", "Three"  
□ Print  
□ print "You're awesome!"  
□ print '"Thank you!" I replied.'  
□ # Ошибочные записи  
□ print 'It's mine'  
□ print "I said \"hi\" to him"
```

# Функция `type`.

## Функции преобразования типов

`str, int, float`

- `print 3, -1, 3.14159, -2.8`
- `print type(3), type(3.14159)`
- `print type(3.0)`
- `print int(3.14159), int(-2.8)`
- `print float(3), float(-1)`
- `Print str(34)`

# Точность вещественных чисел

- # floating point number have around 15 decimal digits of accuracy
- # pi is  
3.1415926535897932384626433832795028841971
- # square root of two is  
1.4142135623730950488016887242096980785696
- # approximation of pi, Python displays 12 decimal digits
  
- print  
3.1415926535897932384626433832795028841971
- print  
1.4142135623730950488016887242096980785696

# Порядок выполнения арифметических операций

```
#"please excuse my dear aunt  
sallie" = ( ), **, *, /, +, -
```

```
□ print 1 + 2, 3 - 4, 5 * 6, 2 ** 5
```

```
□ print 1.0 / 3, 5.0 / 2.0, -7 / 3.0
```

```
□ print 1 / 3, 5 / 2, -7 / 3
```

```
□ print 1 + 2 * 3, 4.0 - 5.0 / 6.0,  
7 * 8 + 9 * 10
```

```
□ print 1 * 2 + 3 * 4
```

```
□ print 1 * (2 + 3) * 4
```

```
□ print 1 * 5 * 4
```

# Имена переменных должны быть осмысленными!

```
□ my_name = "Ms V"  
□ print my_name  
□ my_age = 51  
□ print my_age  
□ my_age += 1  
□ print my_age  
□ magic_pill = 30  
□ print my_age - magic_pill  
□ my_grand_dad = 74  
□ print my_grand_dad - 2 * magic_pill
```

# Осмысленные и неосмысленные

# Good

- num\_letters = 26
- favorite\_number = 18
- minutes = 60
- months = 12

# Bad

- nl = 26
- # new line? next line?  
nil?
- fanum = 18
- # I have no idea...
- mi = 60
- # minutes? millennium?  
mice? do re mi?
- mo = 12
- # months? monster?  
mouse? more?

# Формат записи вещественные числа

□ 17.0 / 3

□ 3.14e1

□ 3.14e5

□ 3.14e-3

□ -1e6

□ -1e-6

# Ввод данных

(отличается в IDLE Python и Codeskulptor)

```
>>> a = input()
```

```
>>> a = int(input())
```

```
>>> b = int(input())
```

```
>>> c = raw_input()
```

# Имена переменных

# Правильные

```
□ num = 2
□ my_name = 3
□ it9 = 9
□ it_10 = 10
□ __do__this = 12
```

# Неправильные

```
□ 9ish = 9.001
□ so-cute = 32
□ i<3u = 9001
□ this&that = 24
□ me+u = 2
```

# Задание 1

## ■ Входные данные:

□ 3 числа x, y и z, такие что:

x, y - положительные целые числа, z равно 0 или 1, x  $\neq$  0.

## ■ Выходные данные:

□ строка "Everybody sing a song <текст песни>.", где <текст песни> формируется из куплетов, разделенных пробелами.

□ Все куплеты одинаковы и состоят из x 'la' через дефис.

□ Если z равно единице, в конце ставится восклицательный знак, иначе точка.

□ При отсутствии куплетов пробел перед точкой или восклицательным знаком не ставится.

## ■ Пример:

□ Входные данные: 2 3 1

Результат: Everybody sing a song: la-la la-la la-la!

□ Входные данные: 1 0 0

Результат: Everybody sing a song:.

# Задание 2

- Входные данные:
  - 2 неотрицательных действительных числа  $a$  и  $b$ ,  $b \neq 0$ .
- Выходные данные:
  - действительное число - результат вычисления формулы
- Формула в графическом записи:

$$x = \frac{\sqrt{ab}}{e^a * b} + ae^{\frac{2a}{b}}$$

- Пример
  - Входные данные: 0 1  
Результат: 0.0
  - Входные данные: 0.5 10  
Результат: 0.688209837593

# Д/З

- Входные данные:
  - 3 действительных числа.
- Выходные данные:
  - результат вычисления формулы

$$f(x; \mu, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right)$$

- Аргументы передаются в порядке, указанном в формуле, названия переменных могут использоваться любые.
- Пример
  - Входные данные: 1 1 0.25
  - Результат: 1.59576912161