

PYTHON

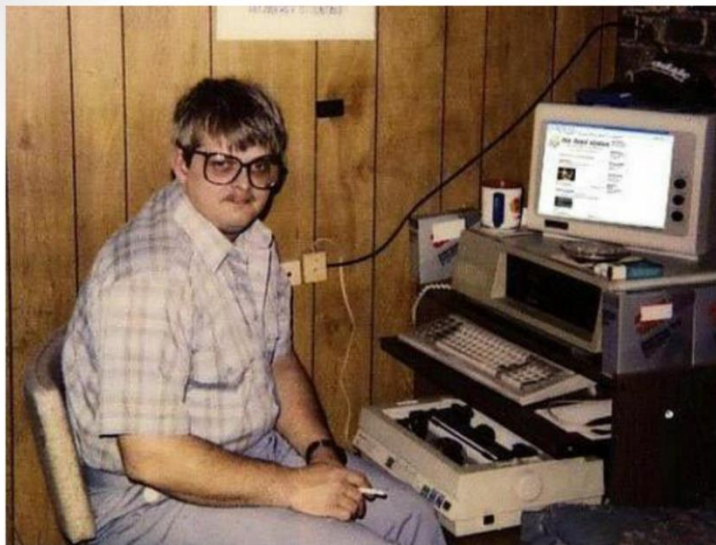
ВВЕДЕНИЕ



ПОЧЕМУ PYTHON?

- Прост в изучении
- Большое количество модулей
- Простые конструкции
- ...

HOLY WAR!

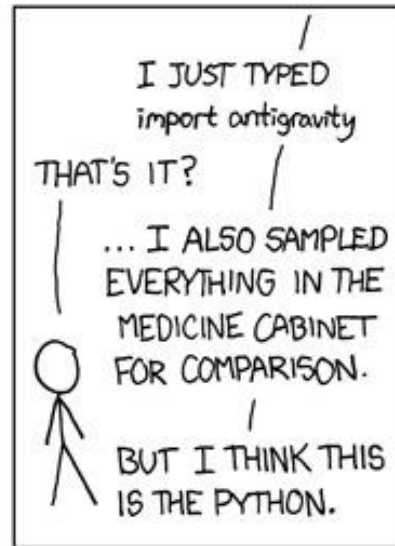
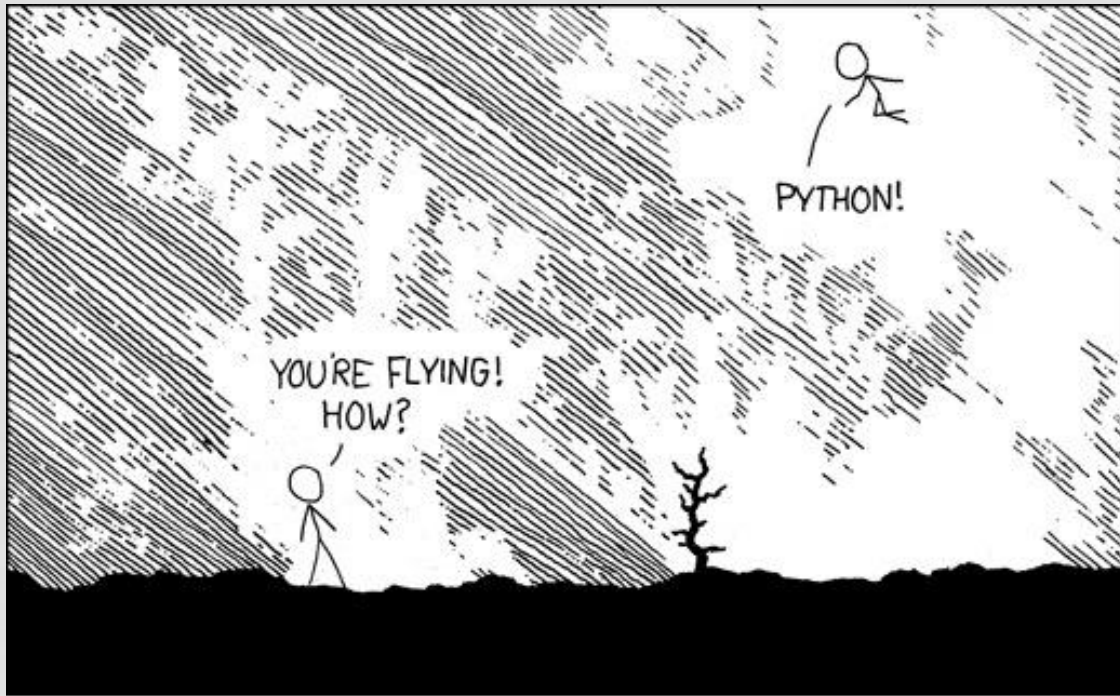


Программист на Perl смотрит на тебя как на `$_[0]`

Программист на Haskell поднимает штангу именем теории категорий



Ruby-программисты — заботливые отцы, а значит их любят девушки



PYTHON

Python - высокоуровневый язык программирования общего назначения с акцентом на производительность разработчика и читаемость кода. Синтаксис ядра Python минималистичен.

Python поддерживает несколько парадигм программирования, в том числе структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное. Основные архитектурные черты — динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений.

PYTHON - ДЗЕН

- `>>> import this`
- The Zen of Python, by Tim Peters
- 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.
- In the face of ambiguity, refuse the temptation to guess.
- There should be one-- and preferably only one --obvious way to do it.
- Although that way may not be obvious at first unless you're Dutch.
- Now is better than never.
- Although never is often better than **right** now.
- If the implementation is hard to explain, it's a bad idea.
- If the implementation is easy to explain, it may be a good idea.
- Namespaces are one honking great idea -- let's do more of those!

КАК ВЫПОЛНЯТЬ

- CPython
- IronPython - .NET
- Jython - Интерпретатор Python, реализованный на Java. Позволяет компилировать программы на Python в байт-код Java.
- PyPy
-

ЗАПУСК

- python
- `#!/usr/bin/env python`
- ...
- IPython
- Python Notebook

PYTHON 2.X VS 3.X

- В третьей версии улучшена стандартная библиотека и добавлены новые функции.
- Много библиотек не стабильно работают на версии 3.
- На данный момент 2.7.12 vs 3.5.2

ТИПЫ ДАННЫХ

- int
- "long int"
- float
- complex

```
>> 4j + 2 + 3j
```

```
(2+7j)
```

```
>> complex (2,7)
```

```
(2+7j)
```

```
>> (2+7j ). real + (2+7j ). imag
```

```
9.0
```

```
>> (2+7j ). conjugate ()
```

```
(2-7j)
```

ПРО ВСТРОЕННЫЕ ТИПЫ

False = None, 0, 0.0, 0j, "", (), [], {}

Boolean operations:

- X or Y
- X and Y
- not X

Numeric types

- $x+y$, $x-y$, $x*y$, x/y , $-x$, $+x$
- $x//y$, $x\%y$, $\text{pow}(x, y)$, $x**y$
- $\text{math.trunc}(x)$
- $\text{round}(x [, n])$
- $\text{math.floor}(x)$, $\text{math.ceil}(x)$

ЗАДАНИЕ

Необязательные:

1. <http://www.pythonchallenge.com/>
2. Посчитать количество существительных в романе «Война и мир». Смотреть на **Py morphology**

Обязательные

3. Найдите все составные числа меньше N , которые представимы в виде произведения двух простых чисел. Число N пользователь вводит с клавиатуры при запуске.
4. Написать функцию, вычисляющую произведение двух матриц (матрица – список списков). Если матрицы нельзя перемножить, сгенерировать исключение. Также написать функцию для вывода такой матрицы в красивом виде.