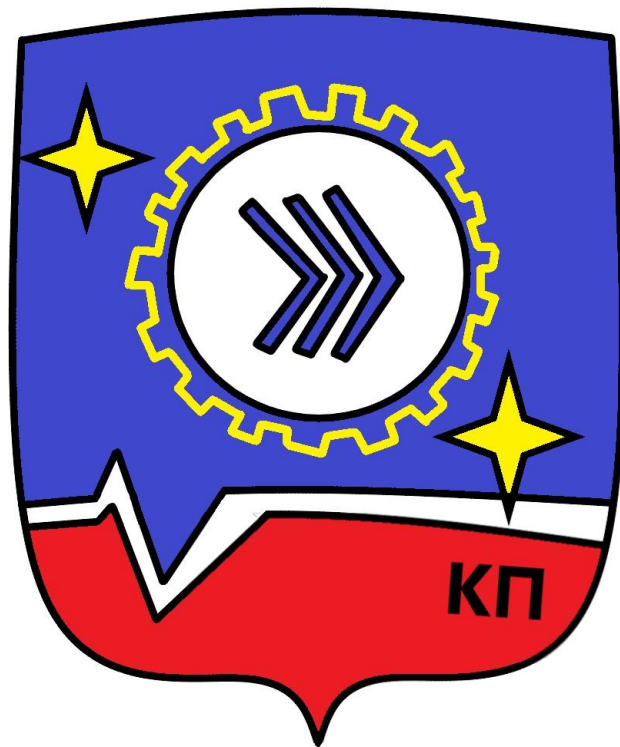


# КЛУБ ПРОГРАММИСТОВ Г.КОРОЛЁВА



# КУРС “Программирование на PYTHON”



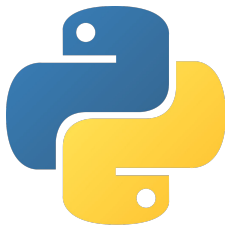


# РЕГЛАМЕНТ



**1. Вопросы можно задавать  
в любое время**





# РЕГЛАМЕНТ



1. Вопросы можно задавать в любое время

**2. Встречаемся 1 раз в неделю  
(или реже) по субботам**





# РЕГЛАМЕНТ



1. Вопросы можно задавать в любое время
2. Встречаемся 1 раз в неделю (или реже) по субботам

**3. Две лекции по 45 минут  
+ дополнительная лекция**






# РЕГЛАМЕНТ



1. Вопросы можно задавать в любое время
2. Встречаемся 1 раз в неделю (или реже) по субботам
3. Две лекции по 45 минут + дополнительная лекция

**4. Домашние задания – ДА,  
аттестация – ДА,  
сертификаты – ДА**

  
**KEEP  
CALM  
AND  
DO YOUR  
HOMEWORK**



# РЕГЛАМЕНТ



1. Вопросы можно задавать в любое время
2. Встречаемся 1 раз в неделю (или реже) по субботам
3. Две лекции по 45 минут + дополнительная лекция
4. Домашние задания - ДА, аттестация - ДА, сертификаты - ДА

## 5. Площадка для общения



[https://vk.com/python\\_korolev](https://vk.com/python_korolev)



+7 (909) 820-56-18



# КАК ВСЕ УСТРОЕНО



Руководитель проектов  
Специалист техподдержки  
Руководитель техподдержки  
Руководитель отдела разработки  
Методолог-консультант  
Веб-программист  
Начальник отдела ИТ  
Менеджер проектов  
Менеджер по продукту  
Бизнес-аналитик  
Администратор проектов  
Технический директор  
Начальник управления инфраструктуры  
Инженер  
Разработчик  
Программист  
Системный администратор  
Верстальщик  
Дизайнер  
Оператор

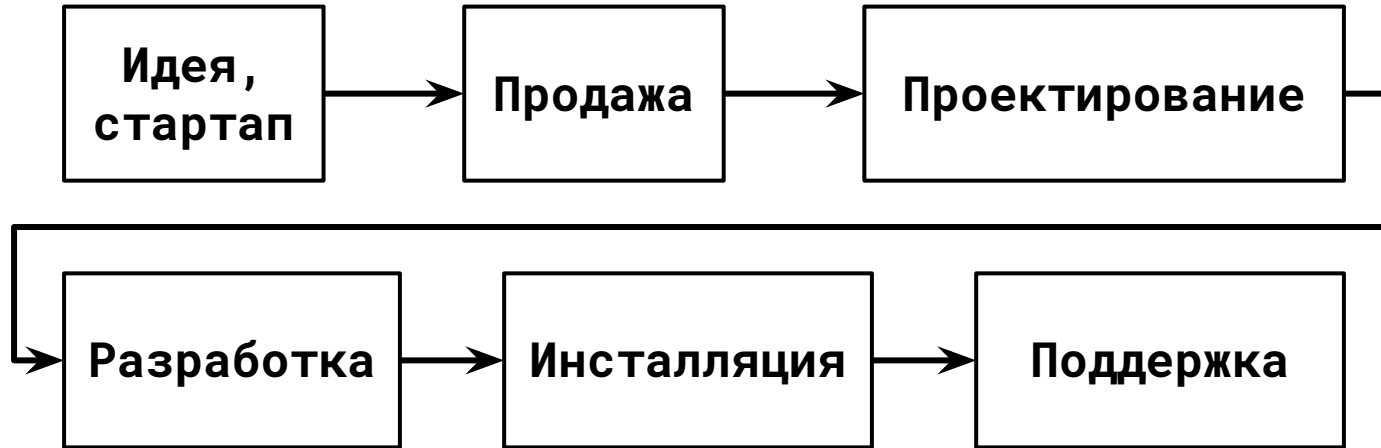


Программирование, Разработка  
Инженер  
Телекоммуникации  
Продажи  
Управление проектами  
Интернет  
Системный администратор  
Поддержка, Helpdesk  
Системная интеграция  
Сетевые технологии  
Аналитик  
Web инженер  
Системы управления предприятием  
Web мастер  
Тестирование  
Администратор баз данных  
Компьютерная безопасность  
Оптимизация сайта (SEO)  
Передача данных и доступ в интернет  
Сотовые, Беспроводные технологии



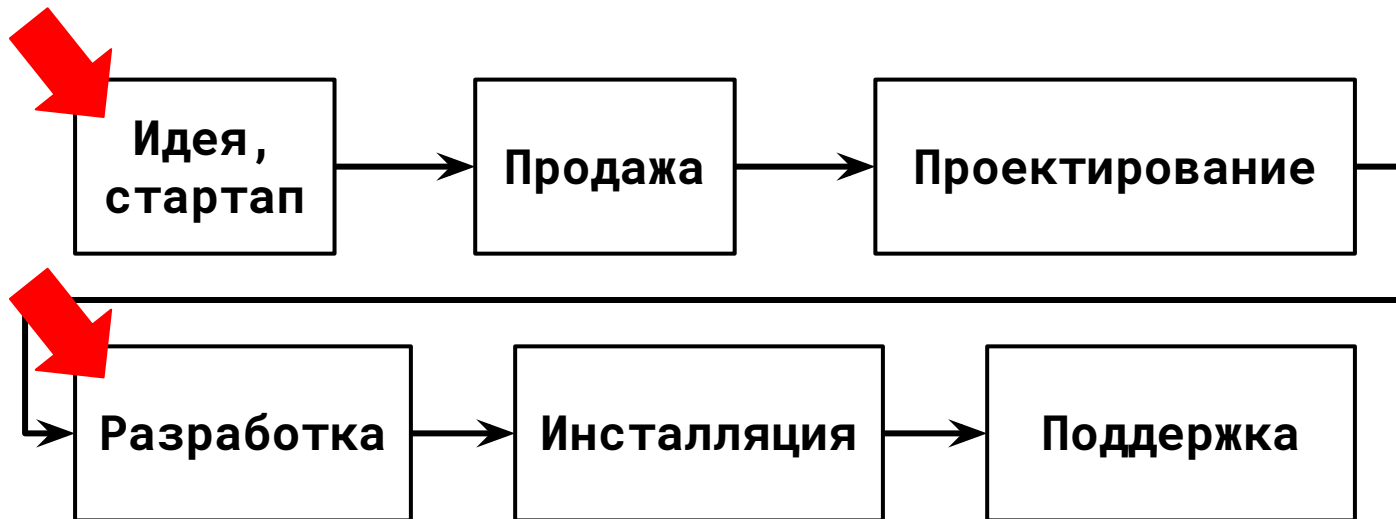


# КАК ВСЕ УСТРОЕНО





# КАК ВСЕ УСТРОЕНО





# КАК ВСЕ УСТРОЕНО

## РАЗРАБОТКА САЙТА

ЧТО ДОЛЖЕН ДЕЛАТЬ РАЗРАБОТЧИК, А ЧТО ЗАКАЗЧИК?

### ЦЕЛИ И ЗАДАЧИ

Выясняем, для чего и зачем нужен сайт. Какую задачу решает?

#### ЮЗАБИЛИТИ-АУДИТ, АНАЛИТИКА

Прорабатываем функционал сайта, анализируем поведение пользователей и их вовлечённость, разрабатываем интерфейс

#### ДИЗАЙН КОНЦЕПЦИЯ

Рисуем внешний вид будущего сайта, отрисовываем интерфейс, иллюстрации

#### ПРОГРАММИРОВАНИЕ, ВЁРСТКА, ДАННЫЕ

Настраиваем CMS, пишем код, переносим данные, подключаем базы данных

#### ЗАПУСК И НАПОЛНЕНИЕ

Выкладываем сайт в интернет. Заполняем контентом. Устраняем ошибки.

#### ТЕХПОДДЕРЖКА

Защищаем сайт от атак конкурентов, поддерживаем от внезапных падений

#### ПРОДВИЖЕНИЕ

Занимаемся контекстом, продвижением и поисковой оптимизацией сайта.





Я.П.



Java  
JavaScript  
Visual Basic  
Python  
C or C++  
C#  
Objective-C  
Perl  
Ruby  
PHP  
R



Java  
JavaScript  
C#  
PHP  
Python  
C++  
Swift  
Ruby  
Go  
TypeScript



JavaScript  
HTML  
CSS  
SQL  
Java  
Bash/Shell  
Python  
C#  
PHP  
C++



Я.П.



Как выбрать  
язык?

PHP JAVA C++ Python  
SWIFT JS RUBY  
C#

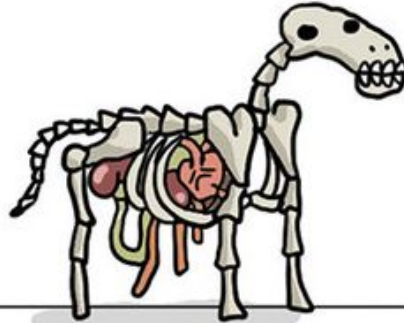




# C++

Вы собрали лошадь

Она адски уродлива и некоторые части  
болтаются, но она работает



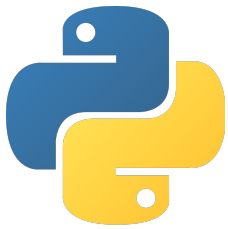
# JAVA

Вы очень хотите  
собрать лошадь



Но сперва вам нужно собрать  
лошадиную фабрику





## JAVASCRIPT

Лошадь прислали  
по частям



Вы собрали лошадь  
но хребет получился угловатым  
и она парализована



## NoSQL

У вас красивая,  
быстрая лошадь



Но у вас нисколько не понятно где она







Приручить ее может только создатель



Во всем остальном  
она дракон

# LISP

Вы собрали

Вы собрали

(((((

(((((

(((((

))))))

))))))

))))))

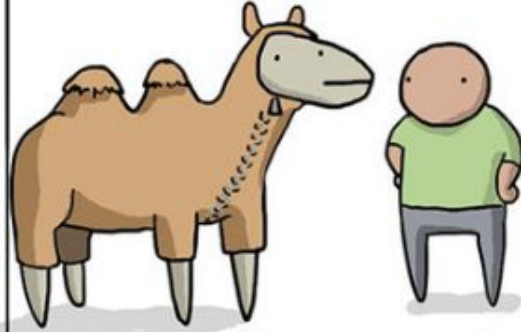






C#

Лошадь работает лучше  
всего в костюме верблюда



Когда вы пытаетесь использовать  
ее не как верблюда  
она становится привередливой



ASSEMBLY

Лошадь получилась немного  
слишком базовой



Но как же быстро она бегает!







ГВИДО



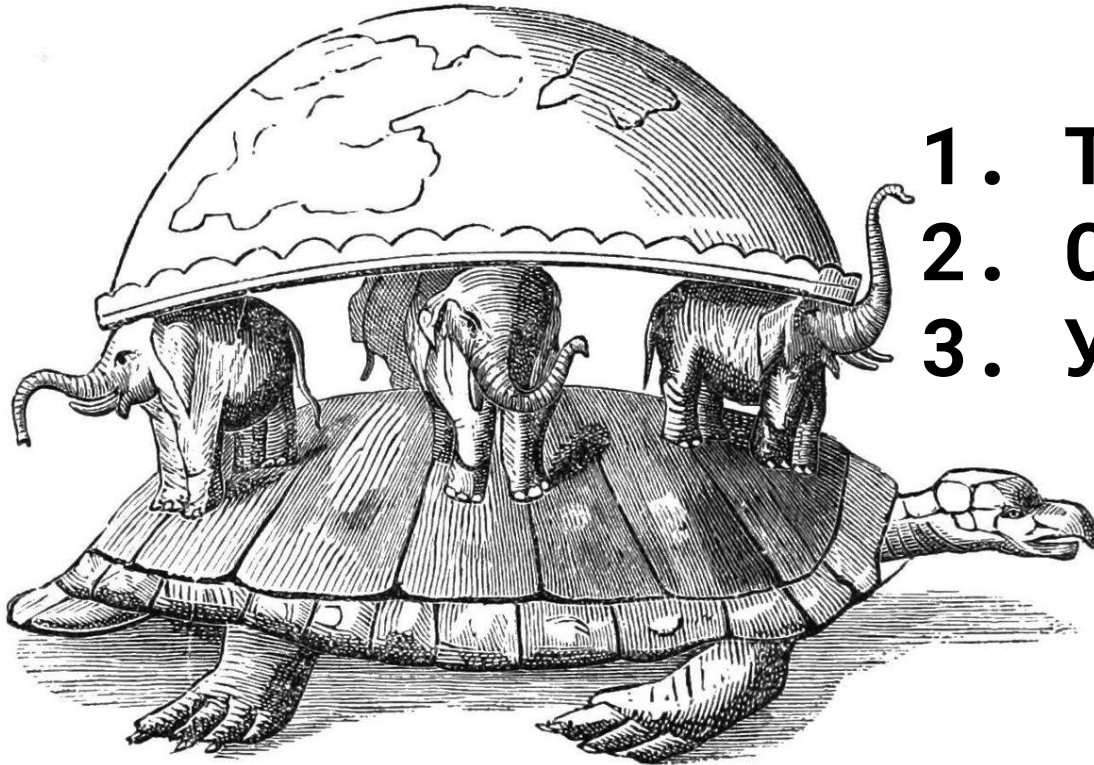




# ЧТО ТАКОЕ ЯП



1. Типы данных
2. Структуры
3. Управляющие конструкции





ДЛЯ ЧЕГО ЯП



WRITE ONLY

EASY READING

```
print("Hello, World!")
```



# ДЛЯ ЧЕГО ЯП WRITE ONLY



```
using System;
```

```
class Program
```

```
{
```

```
    static void Main()
```

```
{
```

```
        Console.Write("Hello, World!");
```

```
    }
```

```
}
```

C#



# ДЛЯ ЧЕГО ЯП WRITE ONLY



## LISP

```
(eval (cons (quote mapcar)
  (cons (cons (quote function)
    (cons (quote princ) ()))
    (cons (cons (quote quote)
      (cons (cons #\H (cons #\e (cons #\l (cons #\l (cons #\o
        (cons #\, (cons #\Space
          (cons #\w (cons #\o (cons #\r (cons #\l (cons #\d (cons #\!
            ()))))))))))))) ())) ())))))
```



# ДЛЯ ЧЕГО ЯП WRITE ONLY



# BRAINF\*CK

+++++++[>++++++>++++++>+++>+<<  
<<-]>++.>+.+++++.+++.>++.<<++++++  
++++.>+.++++.-----.>+.>.





# КРОССПЛАТФОРМЕННОСТЬ



symbian  
OS





# КОМПИЛЯТОР

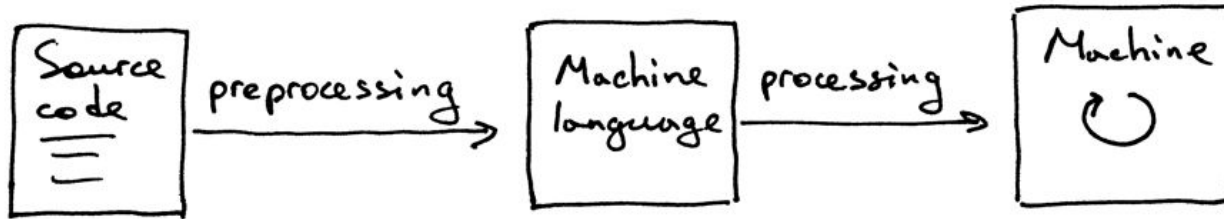


```
00000000 0000 0001 0001 1010 0010 0001 0004 0128
00000010 0000 0016 0000 0028 0000 0010 0000 0020
00000020 0000 0001 0004 0000 0000 0000 0000 0000
00000030 0000 0000 0000 0010 0000 0000 0000 0204
00000040 0004 8384 0084 c7c8 00c8 4748 0048 e8e9
00000050 00e9 6a69 0069 a8a9 00a9 2828 0028 fdfc
00000060 00fc 1819 0019 9898 0098 d9d8 00d8 5857
00000070 0057 7b7a 007a bab9 00b9 3a3c 003c 8888
00000080 8888 8888 8888 8888 288e be88 8888 8888
00000090 3b83 5788 8888 8888 7667 778e 8828 8888
000000a0 d61f 7abd 8818 8888 467c 585f 8814 8188
000000b0 8b06 e8f7 88aa 8388 8b3b 88f3 88bd e988
000000c0 8a18 880c e841 c988 b328 6871 688e 958b
000000d0 a948 5862 5884 7e81 3788 1ab4 5a84 3eec
000000e0 3d86 dcb8 5cbb 8888 8888 8888 8888 8888
000000f0 8888 8888 8888 8888 8888 8888 8888 0000
00001000 0000 0000 0000 0000 0000 0000 0000 0000
*
00001030 0000 0000 0000 0000 0000 0000 0000
0000103e
```

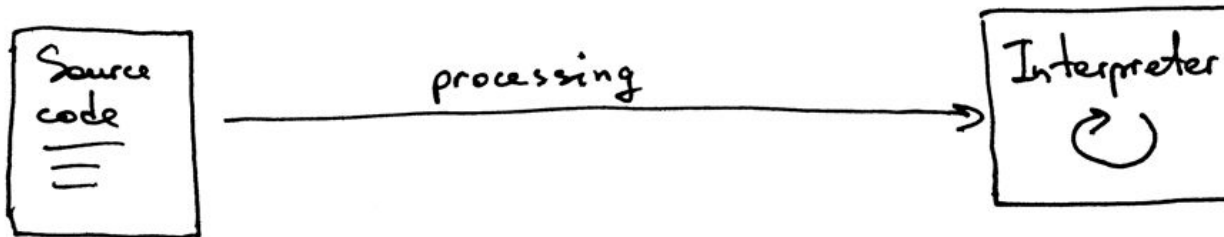


# ИНТЕРПРЕТАТОР

## Compiler

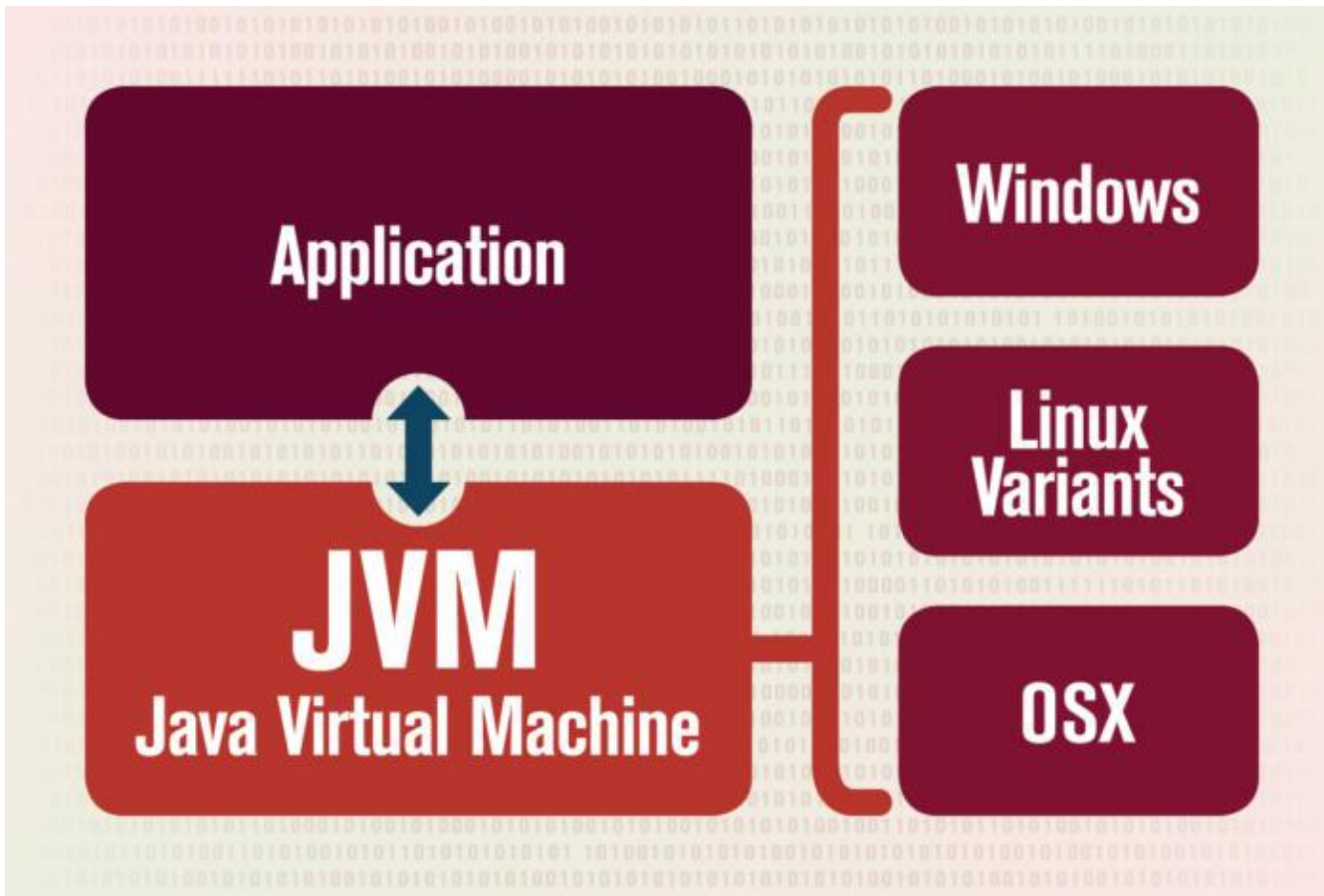


## Interpreter





# ВИРТУАЛЬНЫЕ МАШИНЫ





MY NAME IS

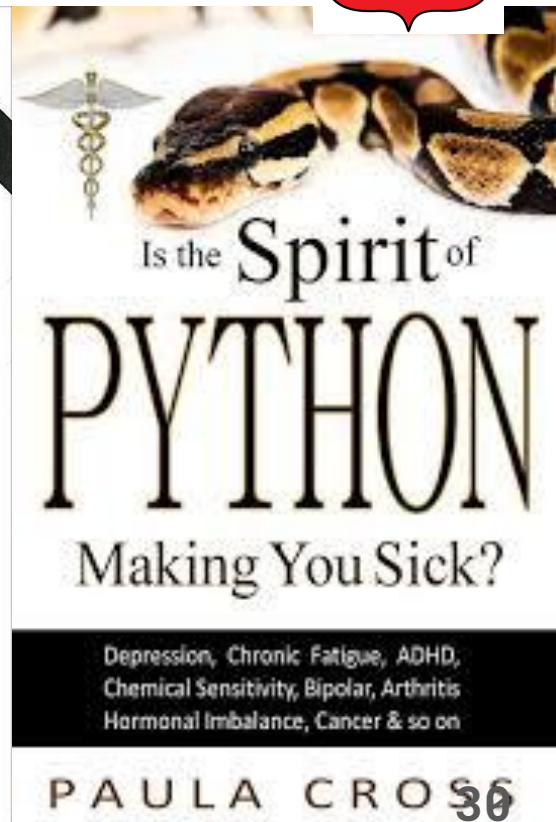
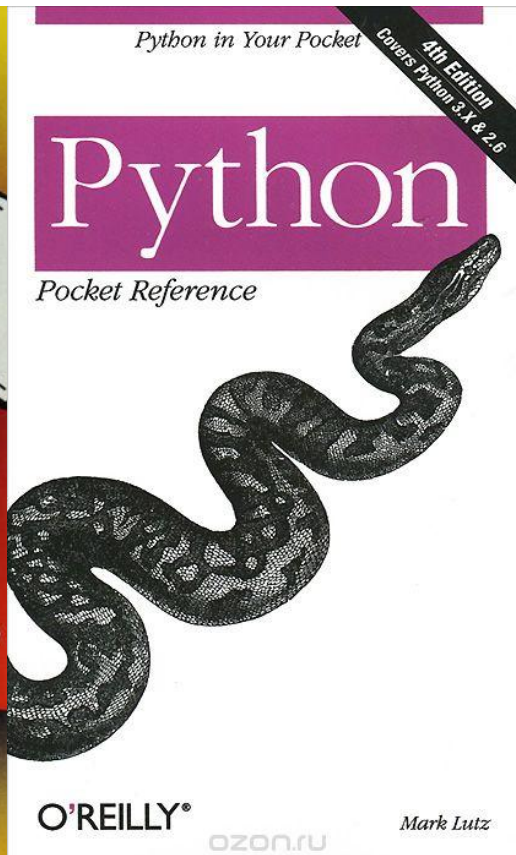
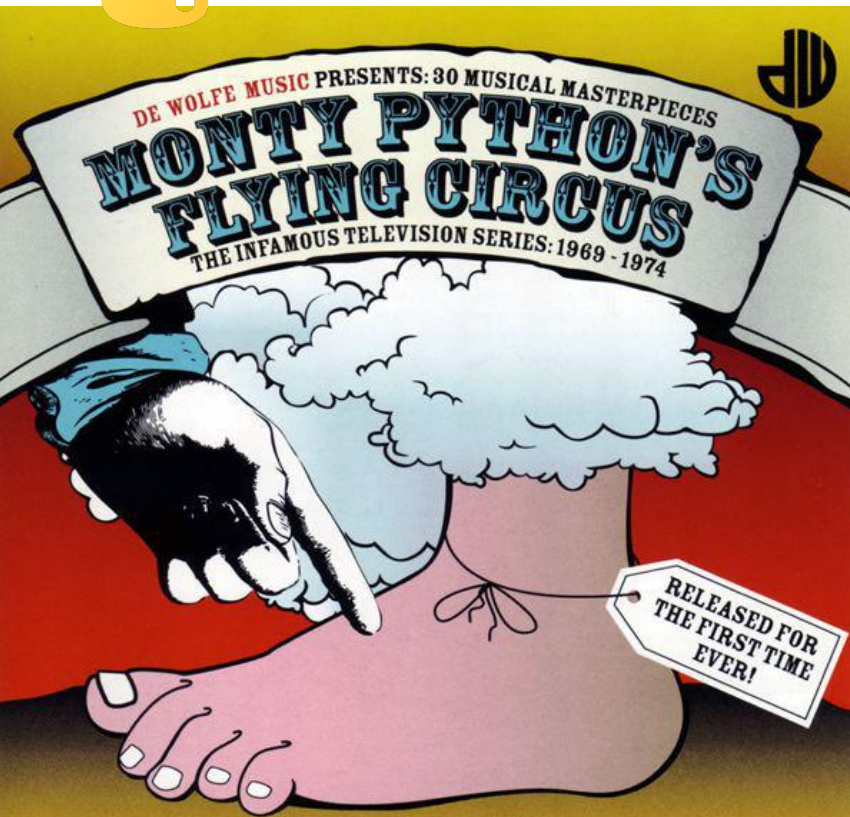


python



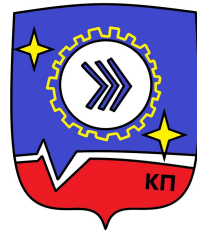


# ЦИРК и ЗМЕИ






# ДОМ: python.org



[Python](#)[PSF](#)[Docs](#)[PyPI](#)[Jobs](#)[Community](#)


 **python**<sup>TM</sup>

[GO](#) [Socialize](#)

[About](#)[Downloads](#)[Documentation](#)[Community](#)[Success Stories](#)[News](#)[Events](#)

```
# Python 3: Simple output (with Unicode)
>>> print("Hello, I'm Python!")
Hello, I'm Python!

# Input, assignment
>>> name = input('What is your name?\n')
>>> print('Hi, %s.' % name)
What is your name?
Python
Hi, Python.
```



### Quick & Easy to Learn

Experienced programmers in any other language can pick up Python very quickly, and beginners find the clean syntax and indentation structure easy to learn. [Whet your appetite](#) with our Python 3 overview.

[1](#)[2](#)[3](#)[4](#)[5](#)



2 / 3



<https://pythonclock.org>

Python 2.7 will retire in...

1

Year

3

Months

23

Days

21

Hours

40

Minutes

3

Seconds

[Enable Gantt Mode](#) [Huh?](#)

### What's all this, then?

Python 2.7 [will not be maintained past 2020](#). Originally, there was no official date. Recently, that date has been updated to [January 1, 2020](#). This clock has been updated accordingly. My original idea was to throw a Python 2 Celebration of Life party at PyCon 2020, to celebrate everything Python 2 did for us. That idea still stands. (If this sounds interesting to you, email [pythonclockorg@gmail.com](mailto:pythonclockorg@gmail.com)).

Python 2, thank you for your years of faithful service.

Python 3, your time is now.





## ПЛЮСЫ



1. ПОПУЛЯРНОСТЬ
2. ЧИТАЕМОСТЬ
3. ПРОСТОЙ СИНТАКСИС
4. БИБЛИОТЕКИ
5. КРОССПЛАТФОРМЕННОСТЬ
6. ПРИМЕНИМОСТЬ ВО МНОГИХ ОБЛАСТЯХ



# МИНУСЫ



1. СКОРОСТЬ
2. ПРОСТОЙ СИНТАКСИС
3. ВТОРОЙ ЯЗЫК



# УСТАНОВКА





PYTHON

## ВАКАНСИИ



Резюме

Вакансии

Услуги

Помощь

Компании

Проекты

4 998 вакансий «PYTHON»

## Регион

Россия 4359

Москва 2048

Санкт-Петербург 773

Еще 114

## Зарплата

Указана 1498

от 60 000 руб. 1283

от 125 000 руб. 717

от 190 000 руб. 340

от 250 000 руб. 145

от 315 000 руб. 45

по соответствию

за месяц

## Python-разработчик

IBA Group ✓ ☆

Минск, ● Уручье

Разработка кода на **Python**. Unit-тестирование.  
2+ лет разработки клиент-серверных приложений под Linux. 2+ лет практическ

[Откликнуться](#)[Не показывать](#)[В избранное](#)

## Python developer

SaM Solutions Самсолюшнс ✓


Минск, ● Московская




# PYTHON ONLINE


## repl.it







@anonymous/DryFloweryLegacysystem 




No description


run 


share 


 repl talk

Sign up

Files   



 main.py



main.py  history

1 | Not sure what to do? Run some [examples](#) (start typing to dismiss)

Python 2.7.10 (default, Jul 14 2015, 19:46:27)  
[GCC 4.8.2] on linux



# PYTHON и command line



```
C:\> Command Prompt - python

Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Eric>python
Python 2.7.12 (v2.7.12:d33e0cf91556, Jun 27 2016, 15:19:22) [MSC v.1500 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> pytest.py
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'pytest' is not defined
>>> _
```



# РЕДАКТОРЫ & IDE

Integrated  
Development  
Environment



**ATOM**



**SUBLIME TEXT 3**



**PYCHARM**



**VS CODE**



# ПАРАДИГМЫ ПРОГРАММИРОВАНИЯ



**Императивное программирование**

**Структурное программирование**

**Функциональное программирование**

**Объектно-ориентированное**

**программирование**





# ТИПЫ ДАННЫХ



# int

ЦЕЛОЕ - INTEGER

1

2

3