

Основы и методология программирования

Семинар 1

Введение

Ввод и вывод данных

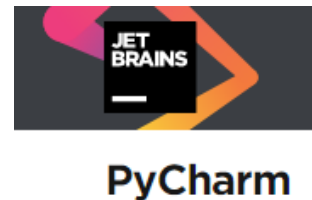
Арифметические операции

Основные идеи Python, среды разработки

- Гвидо ван Россум, 1991 год
- **Python** - интерпретируемый, объектно-ориентированный язык программирования высокого уровня.
- Встроенные высокоуровневые структуры данных в сочетании с динамической типизацией и связыванием делают язык привлекательным для быстрой разработки приложений.
- **Python** прост в изучении, в нем придается особое значение **читаемости кода**, а это сокращает затраты на сопровождение программных продуктов.
- **Интерпретатор Python** и большая стандартная библиотека доступны бесплатно в виде исходных кодов и исполняемых файлов для всех основных платформ.

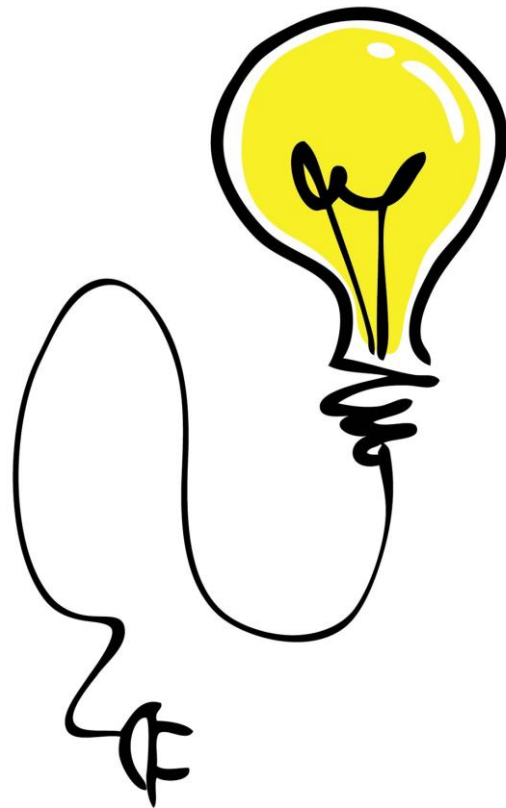


<http://python.org>



Как изучить язык программирования?

- Понять зачем нужен этот язык, место языка среди других
- Стил
- Синтаксис, переменные
- Ввод и вывод
- Операторы, ветвления циклы
- Блоки и модули
- Библиотеки



Установка

Под свою ОС:

Интерпретатор Python версии не ниже 3.4

<https://www.python.org/downloads/>

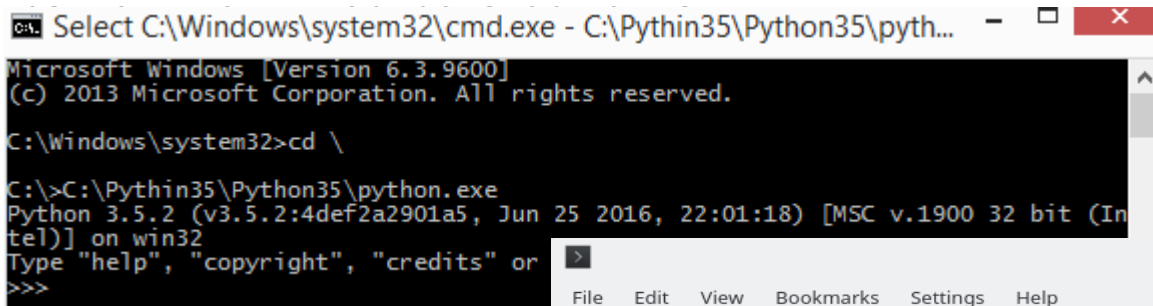
```
sudo apt-get install python3
```

JetBrains PyCharm Community

<https://www.jetbrains.com/pycharm/download/>

Интерпретатор

Наберите python (Windows) или python3 (Linux) в консоли, если всё хорошо,



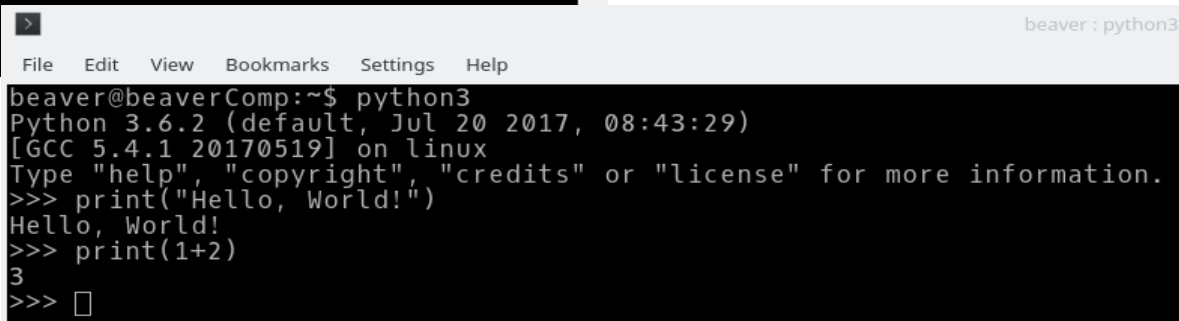
```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd \

C:\>C:\Python35\Python35\python.exe
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or ">>>
```

python3 --version

```
>>> print("Hello, world!")
Hello, world!
>>> print(2 + 39)
41
```



```
beaver@beaverComp:~$ python3
Python 3.6.2 (default, Jul 20 2017, 08:43:29)
[GCC 5.4.1 20170519] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello, World!")
Hello, World!
>>> print(1+2)
3
>>> 
```

Как работают другие основные математические операции?
(попробуйте самостоятельно "*", "/", "**", "//", "%", "<<", ">>")

Создание файлов с исходным кодом (.py)

```
task1.py x
1  # Created by Ivanov Ivan
2  # Date 01.09
3  # Task Description: Распечатать числа строкой и столбиком
4
5  import os
6  from os import path as MyPath
7
8  number = int(input("Введите число: "))
9
10 a = 0
11 b = 10
12 c, d = 20, 30
13
14 print(a, b, c, d, end="\n\t")
15 print(a, b, c, d, sep="\n\t", end="\n")
```

Запуск:



или

python my.py

Вывод:

```
0 10 20 30
      30
      20
      10
      0
```

Важно!

Используем только официальный стиль оформления кода на Python в соответствии с документом "Python Style Guide", <http://python.org>

Ввод данных из консоли

-*- coding: utf-8 -*-

```
s = input("Введите строку:")  
print("Вы ввели: \"", s, "\". Длина строки", len(s))
```

`input()` - запрос строки, строка оканчивается нажатием “*Enter*”

`input(“Строка:”)` - дополнительно вывод приглашения для ввода.

Вывод:

Введите строку: Hello, Python!

Вы ввели: " Hello, Python! ". Длина строки 14

Как изменить формат вывода? (например, убрать пробелы после кавычек)

Преобразование чисел и строк

```
a = int(input()) # преобразование в целое число
```

```
print(a)
```

```
b = int(input(), 16)
```

```
print(b)
```

```
print(int(10.3)) # отбрасывание дробной части
```

```
print(int('0x10',16)) # шестнадцатеричная константа
```

```
print(float('+1.23')) # преобразование в число с плавающей точкой
```

```
print(float(' -12345\n'))
```

```
print(float('1e-003'))
```

```
print(float('+1E6'))
```

```
print(float('-Infinity'))
```

```
str(b) # явное преобразование в строку
```


Задачи на арифметические операции

1. Сформировать число, представляющее собой обратную в порядке следования разрядов запись введенного трехзначного числа. Например, для числа 345 таким будет 543.
2. Дано натуральное число меньше 16. Посчитать количество его единичных битов. Например, если дано число 9, запись которого в двоичной системе счисления равна 1001_2 , то количество его единичных битов равно 2.

Форматирование строк

Формат	Что получится
'%d', '%i', '%u'	Десятичное число.
'%o'	Число в восьмеричной системе счисления.
'%x', '%X'	Число в шестнадцатеричной системе счисления.
'%e', '%E'	Число с плавающей точкой с экспонентой.
'%f', '%F'	Число с плавающей точкой (обычный формат).
'%g' '%G'	Число с плавающей точкой или с экспонентой в зависимости от точности.
'%c'	Символ (строка из одного символа или число - код символа).
'%s'	Строка (как обычно воспринимается пользователем).
'%%'	Знак '%'

Математические функции

math.ceil(X) – округление до ближайшего большего числа.

math.fabs(X) - модуль X.

math.factorial(X) - факториал числа X.

math.floor(X) - округление до ближайшего меньшего числа.

math.isfinite(X) - является ли X числом.

math.isinf(X) - является ли X бесконечностью.

math.log(X, [base]) - логарифм X по основанию base. Если base не указан, вычисляется натуральный логарифм.

math.pi - Константа π

math.pow(X,Y) - X^Y

round(X[,Y]) - округление X до Y знаков или до целого (встроенная функция). “.5” округляется к ближайшему четному(!)

Другие функции библиотеки **math** тут: <https://docs.python.org/3/library/math.html>

Задача

3. Запросить у пользователя длины сторон треугольника, вывести его площадь.

```
from math import sqrt
```

```
a = int(input("Введите a: "))
```

```
b = int(input("Введите b: "))
```

```
c = int(input("Введите c: "))
```

```
p = (a + b + c) / 2
```

```
s = sqrt(p * (p - a) * (p - b) * (p - c))
```

```
print("Площадь равна: " + "%g" % s)
```