A screenshot of a Python 3.7.2 Shell window. The window has a yellow title bar with the text "Python 3.7.2 Shell" and standard window controls (minimize, maximize, close). Below the title bar is a menu bar with options: File, Edit, Shell, Debug, Options, Window, and Help. The main text area contains the following text: "Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit (Intel)] on win32", "Type 'help', 'copyright', 'credits' or 'license()' for more information.", and a prompt ">>> |". The status bar at the bottom right shows "Ln: 3 Col: 4".

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> |
```

НАЧАЛО ИЗУЧЕНИЯ ЯЗЫКА ПИТОН

Загрузка среды разработки
<https://www.python.org/downloads/>

PYTHON

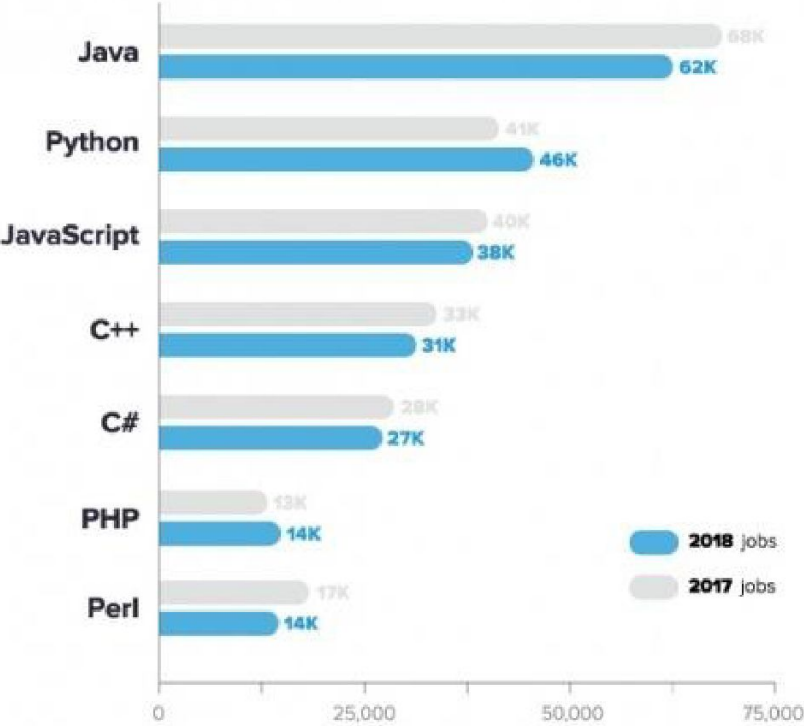
- Python - это интерпретируемый, объектно-ориентированный язык программирования высокого уровня, предназначенный для решения самого широкого круга задач.
- Создатель языка Гвидо ван Россум (примерно в 1991)

- ⦿ Простота в изучении, легкость в написании и чтении кода и лаконичность делают этот язык привлекательным и популярным
- ⦿ Одна из сильных сторон языка Python - его стандартная библиотека, которая позволяет выполнить широкий спектр задач без установки дополнительного ПО

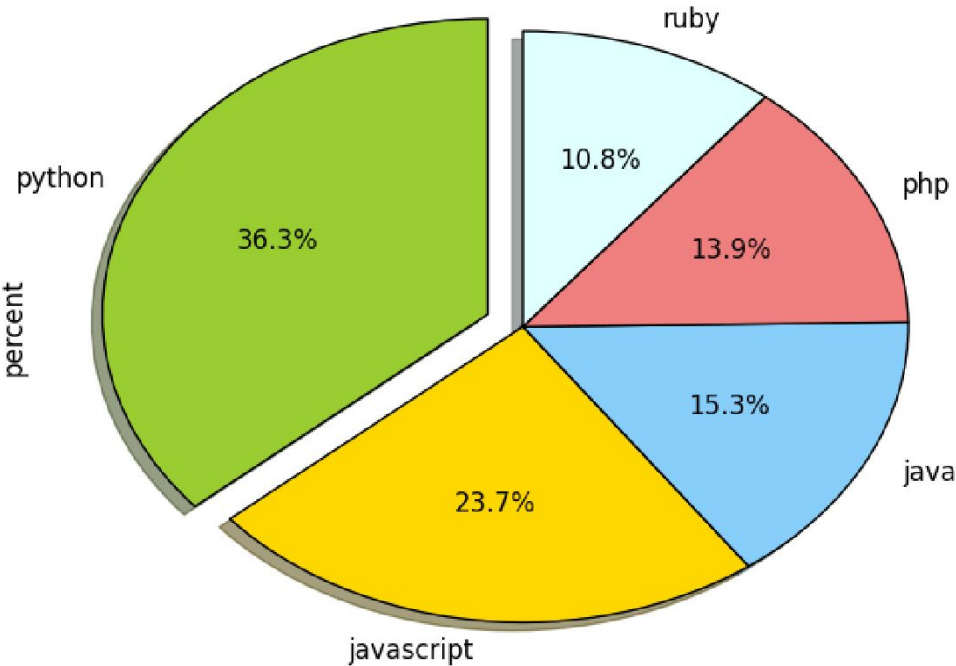
PYTHON ПРОЧНО ЗАНИМАЕТ 3-4-Е МЕСТА В РЕЙТИНГЕ ЯЗЫКОВ, А ИНОГДА ЯВЛЯЕТСЯ ЛИДЕРОМ СРЕДИ ПОПУЛЯРНЫХ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

Job postings containing top languages

Indeed.com - November, 17th 2017



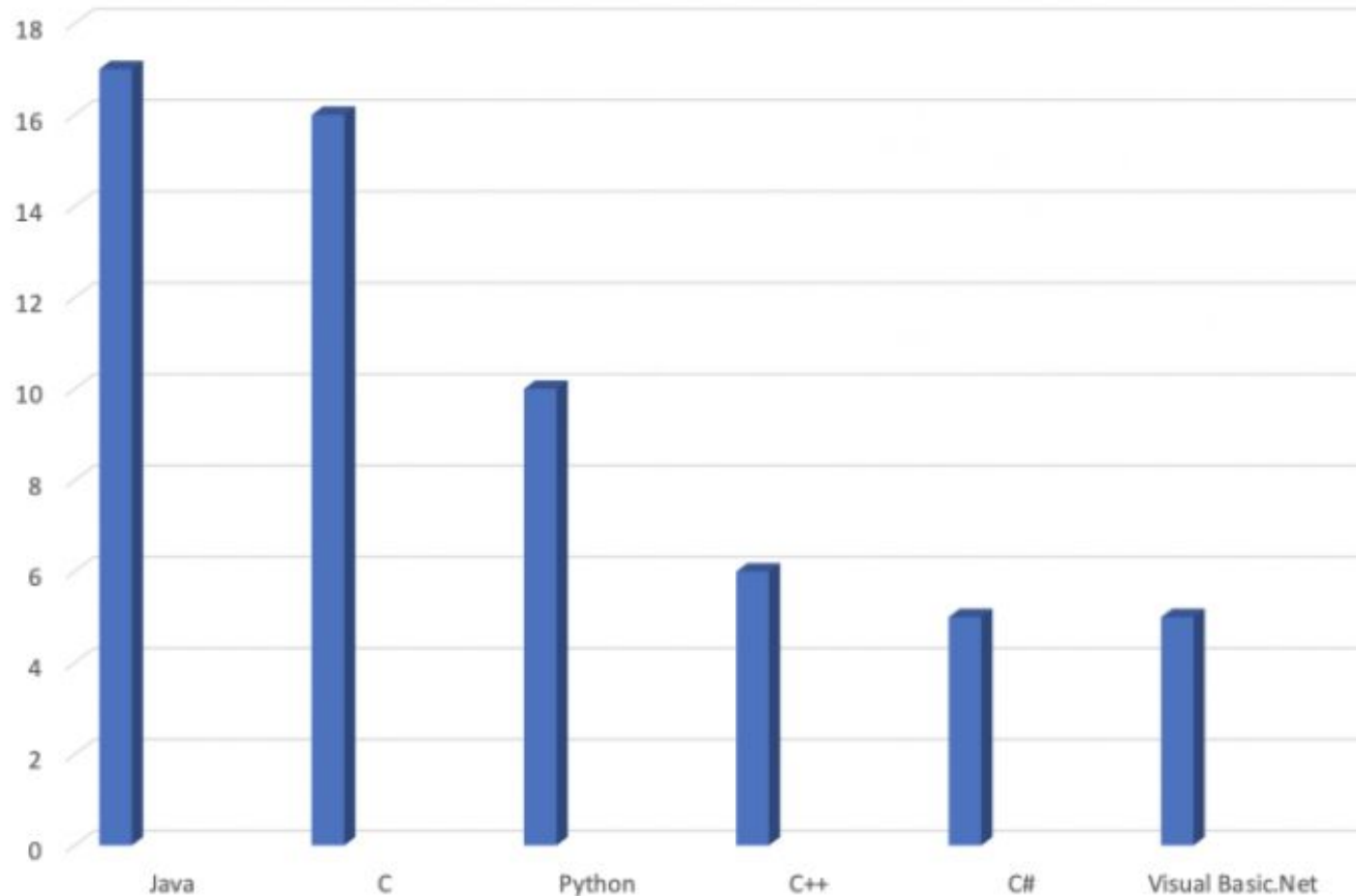
The Top 5 languages on Reddit



[HTTPS://WWW.TIOBE.COM/TIOBE-INDEX/](https://www.tiobe.com/tiobe-index/)

1. Java
2. C
3. Python
4. C++
5. C#
6. Visual Basic .NET
7. JavaScript
8. PHP
9. SQL
10. Objective-C

РЕЙТИНГ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ



Рейтинг популярности языков программирования по данным индекса TIOBE на январь 2020 года.

[HTTPS://GITHUB.COM/](https://github.com/)

1. JavaScript
2. Java
3. Python
4. PHP
5. C++
6. C#
7. TypeScript
8. Shell
9. C
10. Ruby

Большое
влияние на
рейтинг GitHub
оказывает
продуктивность
разработчиков

ОБЛАСТИ ПРИМЕНЕНИЯ PYTHON

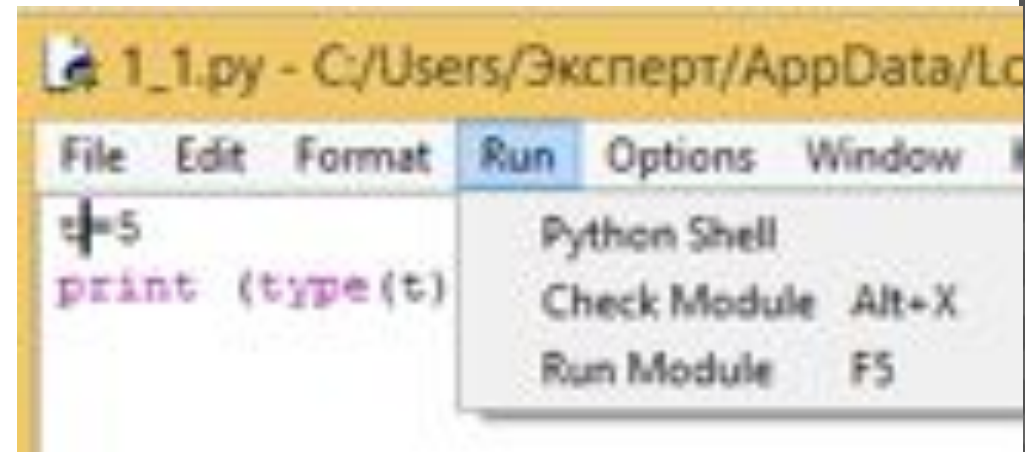
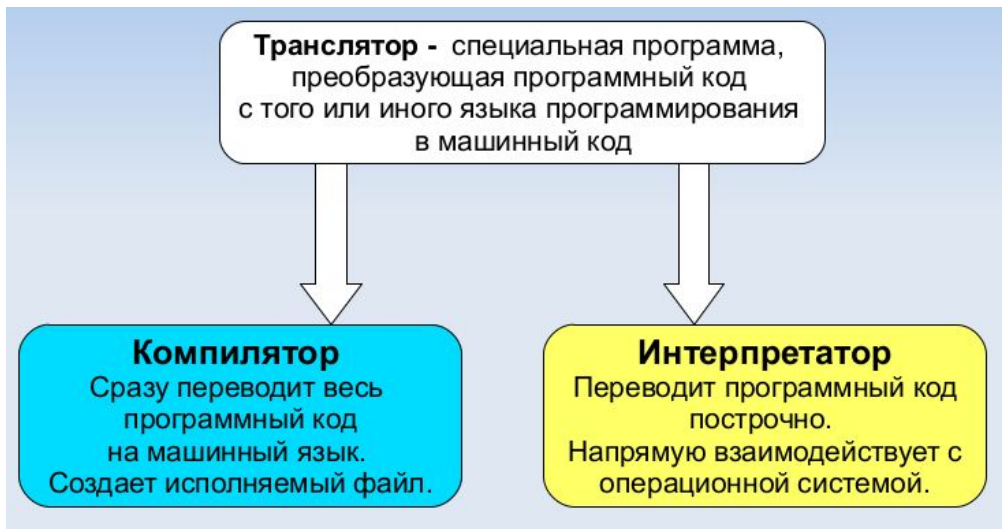
- ◉ Python подходит для решения ежедневных реальных задач
- ◉ Python имеет огромное количество высококачественных уже готовых модулей, распространяемых бесплатно
 - Программы для работы с базами данных (import sqlite3)
 - Игры, искусственный интеллект (Civilization)
 - Программирование математических и научных вычислений (numpy, matplotlib и scipy)

PYTHON ИСПОЛЬЗУЮТ В СВОИХ РАЗРАБОТКАХ ТАКИЕ КРУПНЫЕ ИЗВЕСТНЫЕ КОМПАНИИ КАК YANDEX, GOOGLE, DROPBOX, NASA, ЕГО ИЗУЧАЮТ ВО ВСЕХ ВЕДУЩИХ МЕЖДУНАРОДНЫХ ВУЗАХ



ИНТЕРАКТИВНЫЙ РЕЖИМ

- Интерпретатор выполняет команды построчно.
- Несмотря на удобства интерактивного режима, чаще всего необходимо сохранить исходный программный код для последующего выполнения и использования.



Синтаксис



- ⦿ **Конец строки выражения является окончанием инструкции, никаких знаков на конце не требуется, исключением являются вложенные инструкции**
- ⦿ **Инструкция** - элемент языка, определяющий действие, которое требуется выполнить
 - Пример инструкции:
 - ▣ `>>> print("Python")`
- ⦿ **Строковые выражения** могут заключаться в одинарные, двойные и тройные кавычки

Синтаксис



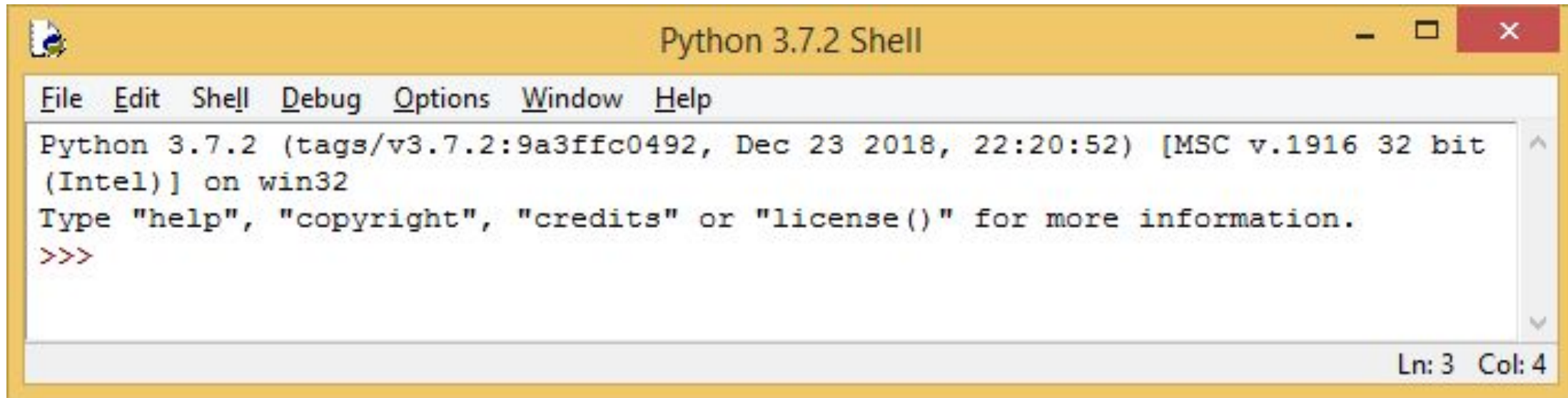
- ⦿ **Комментарии** - вспомогательные строки, не обрабатываемые программой, обозначаются знаком # перед началом строки и действуют до конца строки
- ⦿ **Вложенная инструкция** - часть общей инструкции выполняющаяся при определенных условиях, в этом случае условие заканчивается двоеточием, а само вложенное выражение должно отступать на 4 пробела от места, откуда начинается условие
 - *Отступ в 4 пробела необходим для того, чтобы программа понимала, где начинается вложенная инструкция и к какой основной она относится.*
 - *Можно использовать и табуляцию, но в руководстве по написанию кода на Python отступ в 4 пробела, главное не мешать разным вариантам отступов в одном коде*

В ПРОГРАММЕ МОЖЕТ БЫТЬ НЕСКОЛЬКО УРОВНЕЙ ВЛОЖЕННЫХ ИНСТРУКЦИЙ, В ТАКОМ СЛУЧАЕ НАДО НА КАЖДОМ УРОВНЕ ДЕЛАТЬ ОТСТУПЫ ОТ НАЧАЛА ПРЕДЫДУЩЕЙ ВЛОЖЕННОЙ ИНСТРУКЦИИ

- ◉ Вопрос об отступах является одним из наиболее распространенных в программировании
- ◉ В 2016 году разработчики компании Google решили произвести анализ кода разработчиков, лежащего на ресурсе GitHub и выявить наиболее популярный способ форматирования текста для разных языков программирования
- ◉ Проанализировав 400 тысяч репозиторий они получили следующую картину:
 - Для большинства языков программирования (Java, Python, JavaScript, PHP) разработчики преимущественно используют пробел, однако в языке C более популярным способом форматирования является табуляция

ПИШЕМ ПРОГРАММУ НА PYTHON

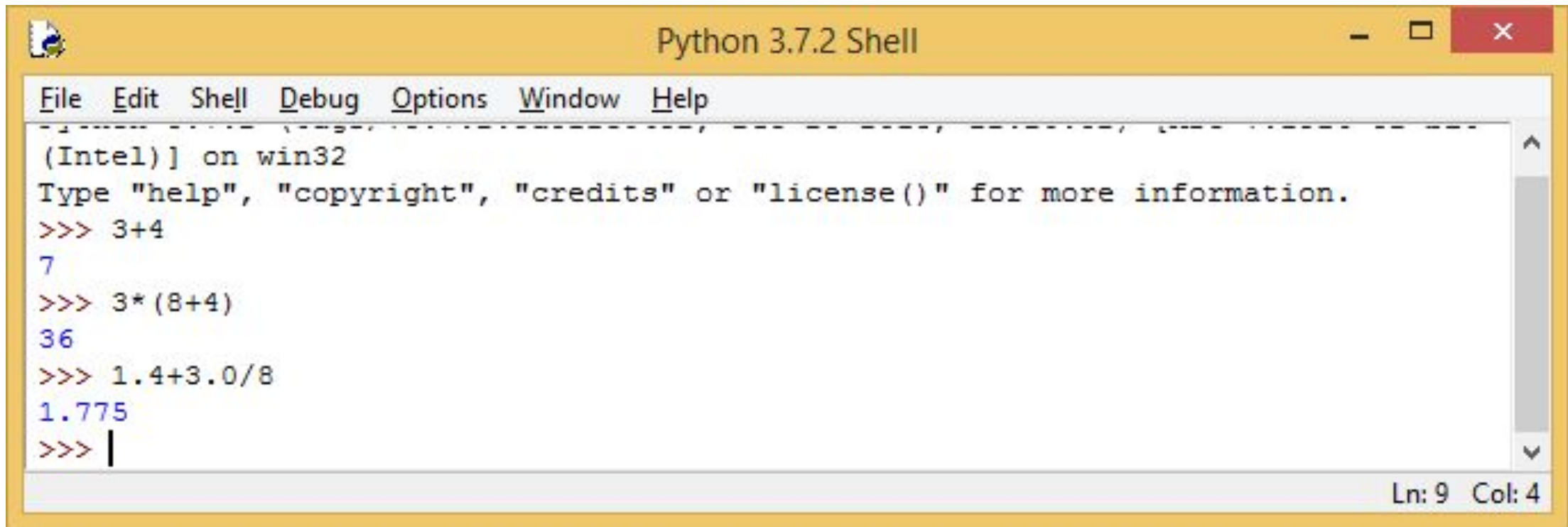
- Запустим IDLE Python

A screenshot of the Python 3.7.2 Shell window. The window has a yellow title bar with the text "Python 3.7.2 Shell" and standard window controls (minimize, maximize, close). Below the title bar is a menu bar with the following items: File, Edit, Shell, Debug, Options, Window, and Help. The main text area contains the following text: "Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit (Intel)] on win32", "Type 'help', 'copyright', 'credits' or 'license()' for more information.", and a prompt ">>>". The status bar at the bottom right shows "Ln: 3 Col: 4".

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```

ИНТЕРАКТИВНЫЙ РЕЖИМ

- Питон как калькулятор



The image shows a screenshot of the Python 3.7.2 Shell window. The window has a yellow title bar with the text "Python 3.7.2 Shell" and standard Windows window controls (minimize, maximize, close). Below the title bar is a menu bar with the following options: File, Edit, Shell, Debug, Options, Window, and Help. The main area of the window is a text editor with a white background and a vertical scrollbar on the right. The text in the editor shows the following sequence of commands and outputs:

```
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 3+4
7
>>> 3*(8+4)
36
>>> 1.4+3.0/8
1.775
>>> |
```

At the bottom right of the window, the status bar shows "Ln: 9 Col: 4".

ИСПОЛНИТЕЛЬ ЧЕРЕПАХА TURTLE

```
>>> import turtle
>>> turtle.clear()
```



```
>>> turtle.forward(40)
```

Forwa rd(n)	Back ward (n)	Up()	Down()	Right (k)	Left (n)	Goto (x,y)	Width (n)	Tracer (flag)	Clear)
Передви жение вперед в направл ении стрелки на n точек	Передви жение назад на n точек	Подняти е пера, чтобы не оставал ось следа при переме щении	Опускан ие пера	Поворот направо по часовой стрелке на k градусо в	Поворот налево по часовой стрелке	Перем ещени е пера с коорд ината ми x,y	Установ ка толщин ы пера	Включен ие и выключе ние режима отображе ния черепахи	Очистка области рисован ия

Задания



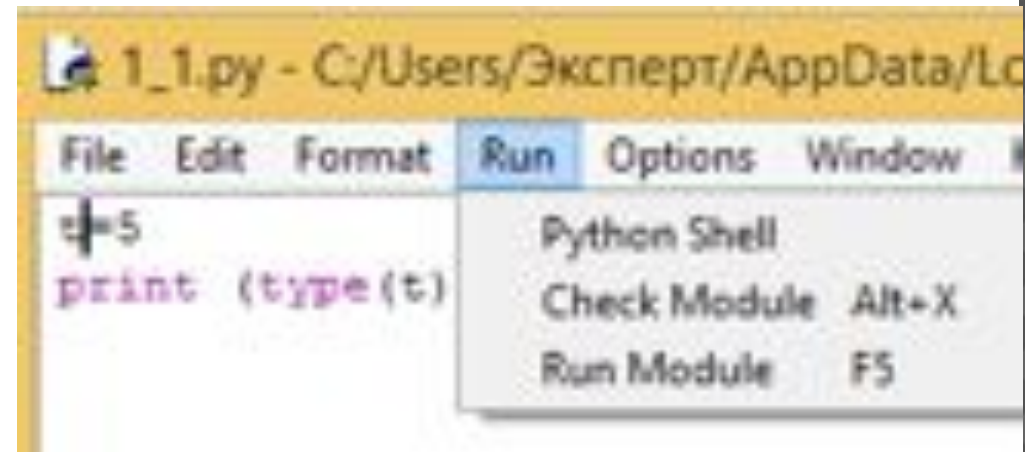
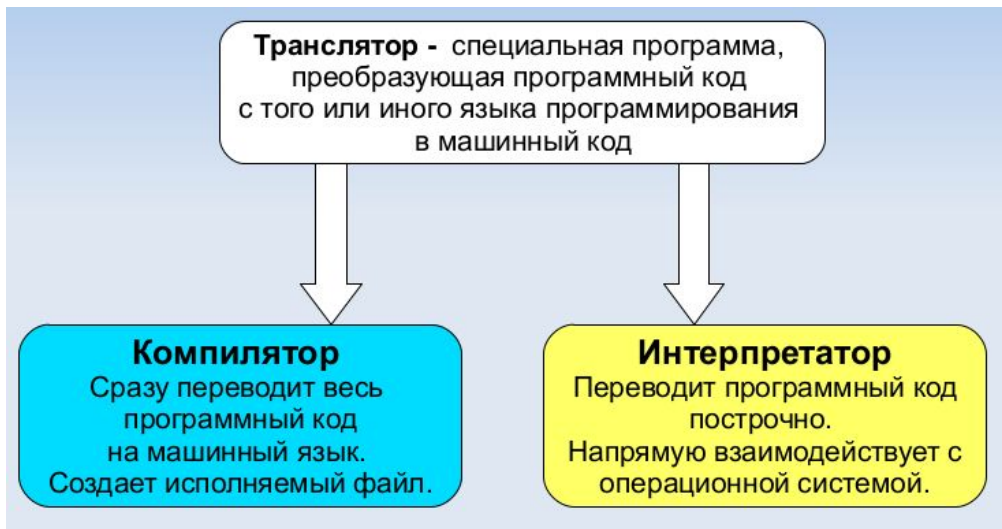
- Напишите программу, в которой на экране получится:
 - изображение прямоугольника 50 на 100 точек
 - правильного шестиугольника
 - равностороннего треугольника
 - трапеции

ИСТОРИЯ ЯЗЫКА ПРОГРАММИРОВАНИЯ

- ◎ **Python** - это достаточно молодой язык программирования. Он был создан примерно в 1991 году голландцем **Гвидо ван Россумом**.
 - Свое название Пайтон (или Питон) получил от названия телесериала, а не пресмыкающегося. Однако по созвучию все равно в качестве эмблемы был удостоен змеи.
 - После того, как Гвидо разработал язык, он выложил его в Интернет, где уже целое сообщество программистов присоединилось к его улучшению.
- ◎ Поскольку Python достаточно молодой язык программирования, то он активно совершенствуется и в настоящее время. Поэтому часто выходят новые версии.

ИНТЕРАКТИВНЫЙ РЕЖИМ

- Интерпретатор выполняет команды построчно.
- Несмотря на удобства интерактивного режима, чаще всего необходимо сохранить исходный программный код для последующего выполнения и использования.



ПОЛУЧЕНИЕ СПРАВКИ

- **Получение справки.** С помощью встроенной функции `help` можно получить справку о команде

```
>>> help (abs)
Help on built-in function abs in module builtins:

abs(x, /)
    Return the absolute value of the argument.
```

ПРОГРАММА НА ЯЗЫКЕ PYTHON

- ⦿ Программа на языке Python представляет собой обычный текстовый файл с инструкциями.
- ⦿ Каждая инструкция располагается на отдельной строке и должна начинаться с начала строки, иначе будет выведено сообщение об ошибке.
- ⦿ Расширение файлов `.py`.

```
>>> 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.
```

ДЗЕН ПИТОНА

- Если интерпретатору Питона дать команду `import this` ("импортируй это" здесь видимо следует понимать как "импортируй самого себя"), то выведется так называемый "Дзен Питона", иллюстрирующий идеологию и особенности данного языка
- *Понимание смысла этих постулатов в приложении к программированию придет тогда, когда вы освоите язык в полной мере и приобретете опыт практического программирования*

ПРОГРАММЫ НА ПИТОН

- Программы на языке Питон представляют собой обычные текстовые файлы, в которых записана последовательность команд. Код легко читается и интуитивно понятен.

```
print('Введите A: ')\nA=input()\nprint('Введите B: ')\nB=input()\nif A==B:\n    print('A равно B')
```


ВВОД И ВЫВОД ДАННЫХ

- `print()` - это команда языка Python, которая выводит то, что в ее скобках на экран.
- `input()` когда вызывается эта функция, программа останавливает свое выполнение и ждет, когда пользователь введет текст.

```
name=input("Введите Ваше имя ")  
gorod=input("С какого Вы города? ")  
print("Здравствуйте {0} с города {1} ".format(name,gorod))
```

```
Введите Ваше имя Лариса  
С какого Вы города? Великий Новгород  
Здравствуйте Лариса с города Великий Новгород  
>>>
```


ВВОД ДАННЫХ

- Для ввода данных предназначена функция `input()`, которая имеет следующий формат: `[переменная]= input ([Сообщение])`.

```
>>> x=input('Введите имя ')\nВведите имя Лариса\n>>> x\n'Лариса'
```

ВЫВОД ДАННЫХ В PYTHON

- Для вывода данных используется функция **print**, которая может выводить не только значения переменных, но и значения любых выражений

```
File Edit Format Run Options Window Help
print('1','2','3',sep=' + ',end='')
print(' =',1+2+3)
print()
```

```
>>>
= RESTART: C:/Users/Эксперт/AppData/Local
1 + 2 + 3 = 6
```

sep (от англ. separator — разделитель),

По умолчанию параметр sep равен строке из одного пробела и между значениями выводится пробел.

end указывает на то, что выводится после вывода всех значений, перечисленных в функции print.

По умолчанию параметр end

ЧТЕНИЕ ДАННЫХ

- ⦿ Функция `input()` считывает строку из консоли, чтобы закончить ввод строки нужно нажать Enter.
 - Под строкой в данном случае понимается английское слово `line`, что означает "строка, оканчивающаяся переводом строки".
- ⦿ Во многих задачах требуется работать со введенными числами, а читать можно только строки. Чтобы преобразовать строку, состоящую из цифр (и, возможно, знака "-" перед ними) в целое число можно воспользоваться функцией `int` (сокращение от английского `integer`, "целое число").
 - `a = int(input())`
 - `b = int(input())`
 - `print(a + b)`

ВЫВОД РЕЗУЛЬТАТОВ РАБОТЫ ПРОГРАММЫ

- ⦿ Вывести результаты работы программы можно с помощью функции `print ()`.
- ⦿ Функция имеет следующий формат:
`print([<Объекты>][, sep=' '][, end='\n'][, file=sys.stdout][, flush=False])`

ВЫВОД РЕЗУЛЬТАТОВ РАБОТЫ ПРОГРАММЫ

```
print("Строка 1")  
print("Строка 2")
```

Строка 1
Строка 2

```
print("Строка 1", "Строка 2")
```

Строка 1 Строка 2

```
print("Строка 1", "Строка 2", sep="")|
```

Строка 1Строка 2

```
print("Строка 1", "Строка 2", end=" ")  
print("Строка 3")
```

```
# Выведет: Строка 1 Строка 2 Строка 3
```

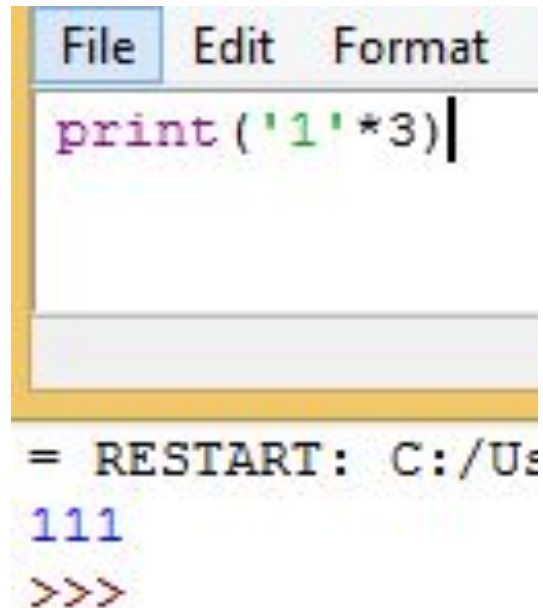
Строка 1 Строка 2 Строка 3

ОПЕРАЦИИ НАД СТРОКАМИ

- Строки также можно сохранять в переменные и использовать в некотором ограниченном количестве выражений. В частности, можно склеивать две строки с помощью операции +
- Складывать число со строкой (и наоборот) нельзя. Но можно воспользоваться функцией `str`, которая по числу генерирует строку
- `answer = '2 + 3 = ' + str(2 + 3)`
- `print(answer)`

УМНОЖЕНИЕ СТРОКИ НА ЧИСЛО

- Можно умножить строку на целое неотрицательное число, в результате получится исходная строка, повторенная заданное число раз:



The image shows a screenshot of a Python interpreter window. The window has a menu bar with 'File', 'Edit', and 'Format'. The main text area contains the code `print('1'*3)` with a cursor at the end. Below the text area, the output of the code is shown: `= RESTART: C:/Us`, `111`, and `>>>`.

```
File Edit Format
print('1'*3)|
= RESTART: C:/Us
111
>>>
```

ЗАДАНИЕ

- ⦿ Напишите программу для вывода 3-х одинаковых фигур (ромбов, треугольников, квадратов), нарисованных в виде звездочек

== RESTART: C:/Users/Эксперт/AppDat
3

```
      ~
    (o o)
   /  v  \
  / (  -  ) \ / (  -  ) \ / (  -  ) \
   ^^    ^^    ^^    ^^    ^^    ^^
>>> |
```

2.py - C:/Users/Эксперт/App

File Edit Format Run Options

```
n=int(input())
print('      ~          ' * n)
print('      (o o)        ' * n)
print('    /  v  \         ' * n)
print('  / (  -  ) \ \ \    ' * n)
print('    ^^    ^^         ' * n)
```

КОММЕНТАРИИ

```
# Это комментарий
```

```
print("Привет, мир!") # Выводим надпись с помощью функции print()
```

```
# print("Привет, мир!") Эта инструкция выполнена не будет
```

```
print("# Это НЕ комментарий")
```

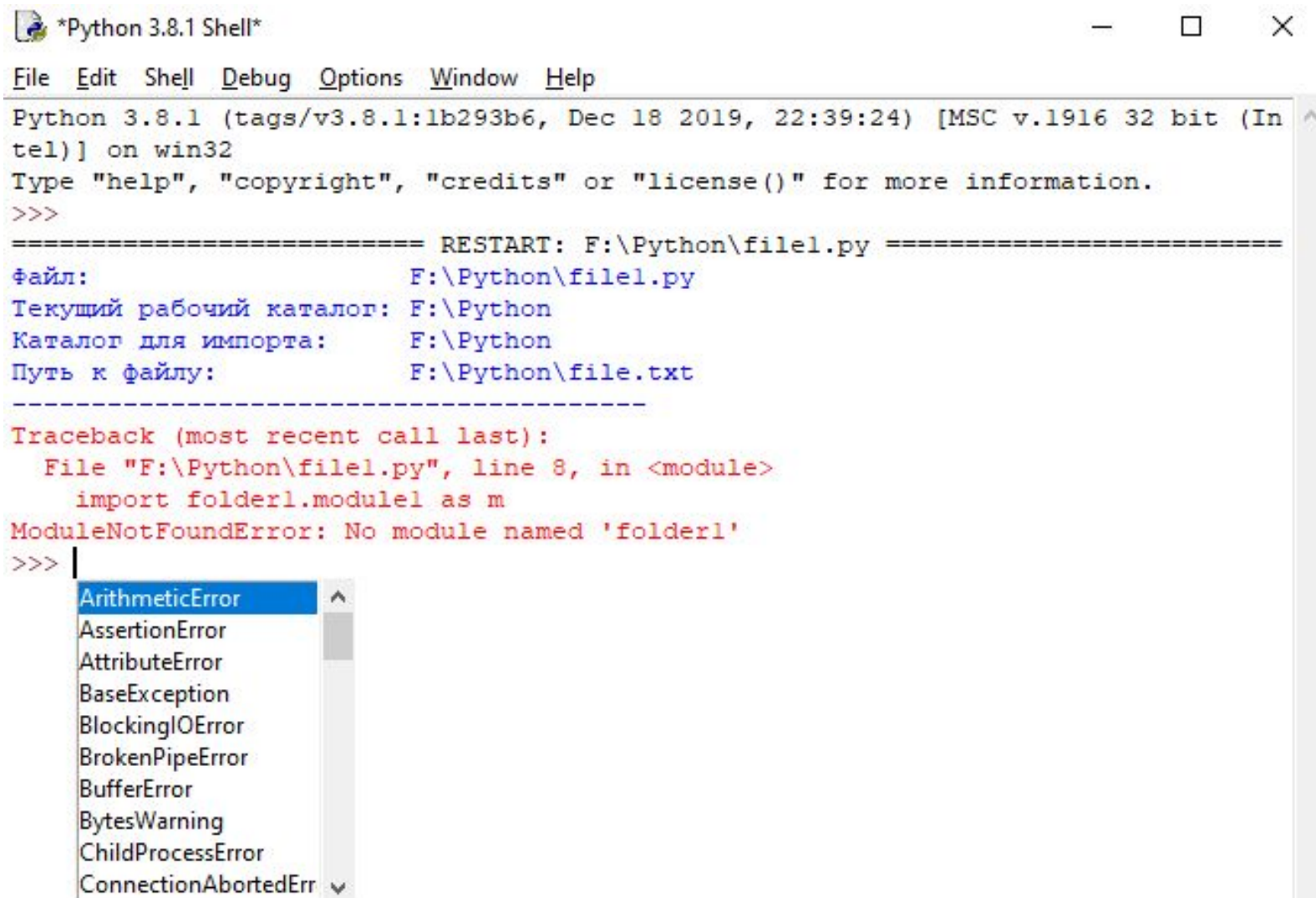
```
"""
```

```
Эта инструкция выполнена не будет
```

```
print ("Привет, мир!")
```

```
"""
```

ВОЗМОЖНОСТИ IDLE



```
*Python 3.8.1 Shell*

File Edit Shell Debug Options Window Help

Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (In
tel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: F:\Python\file1.py =====
Файл: F:\Python\file1.py
Текущий рабочий каталог: F:\Python
Каталог для импорта: F:\Python
Путь к файлу: F:\Python\file.txt
-----
Traceback (most recent call last):
  File "F:\Python\file1.py", line 8, in <module>
    import folder1.module1 as m
ModuleNotFoundError: No module named 'folder1'
>>>
ArithmeticError
AssertionError
AttributeError
BaseException
BlockingIOError
BrokenPipeError
BufferError
BytesWarning
ChildProcessError
ConnectionAbortedErr
```

