

# Функции

---

# ФУНКЦИИ

---

```
def имя_функции ([параметры]) :  
    инструкции
```

#Пример

```
def say_hello() :  
    print("Hello")
```

# ФУНКЦ

```
def say  
    pri
```

```
say_hel
```

```
say_hel
```

```
say_hel
```

```
Python 3.7.3 Shell  
File Edit Shell Debug Options Window Help  
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 22:22:05) [MSC v.1916 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
===== RESTART: C:/Users/Камиль/Downloads/1.py =====  
Hello  
Hello  
Hello  
>>> |
```

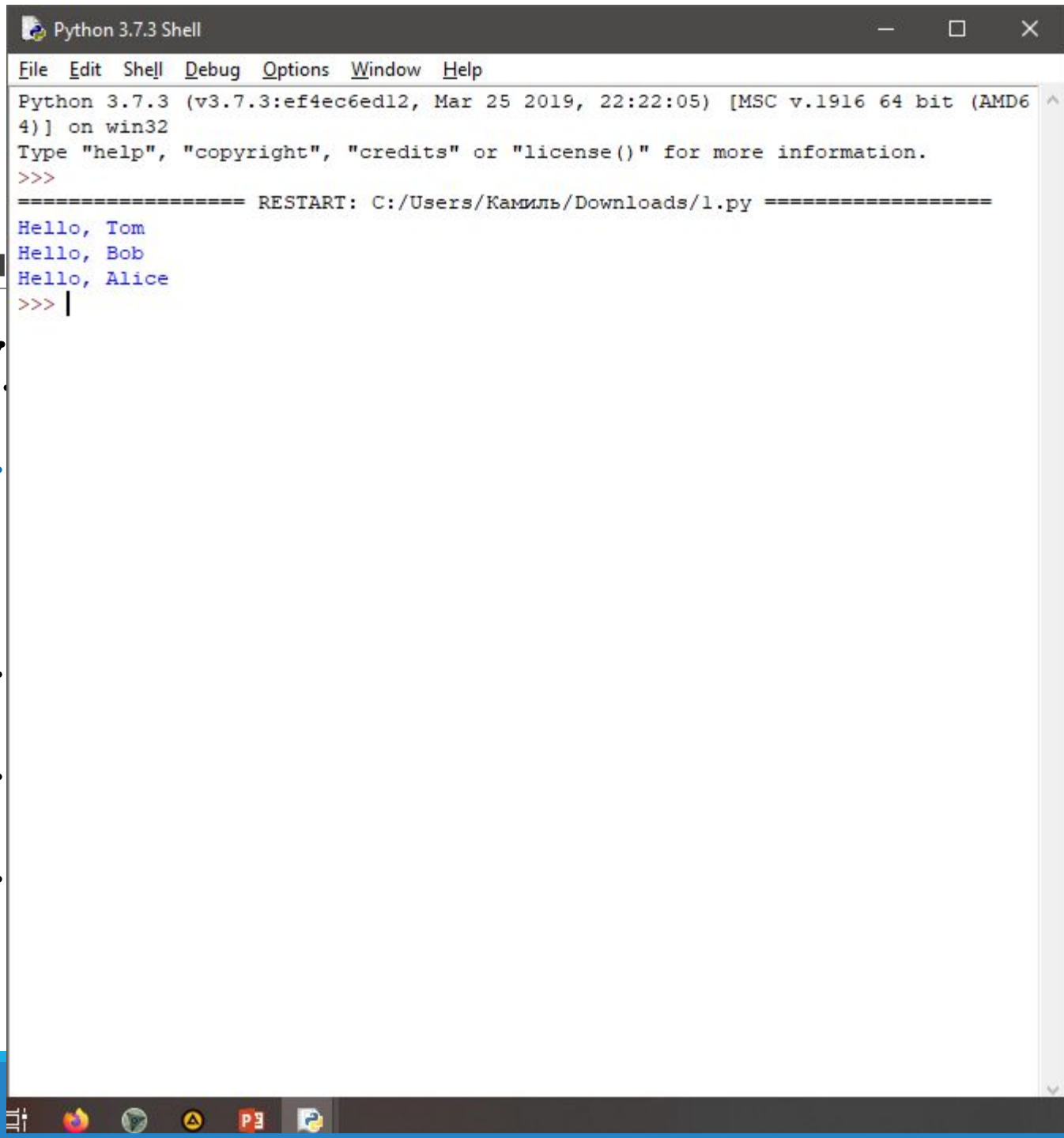
# ФУНКЦ

```
def say  
    pri
```

```
say_hel
```

```
say_hel
```

```
say_hel
```

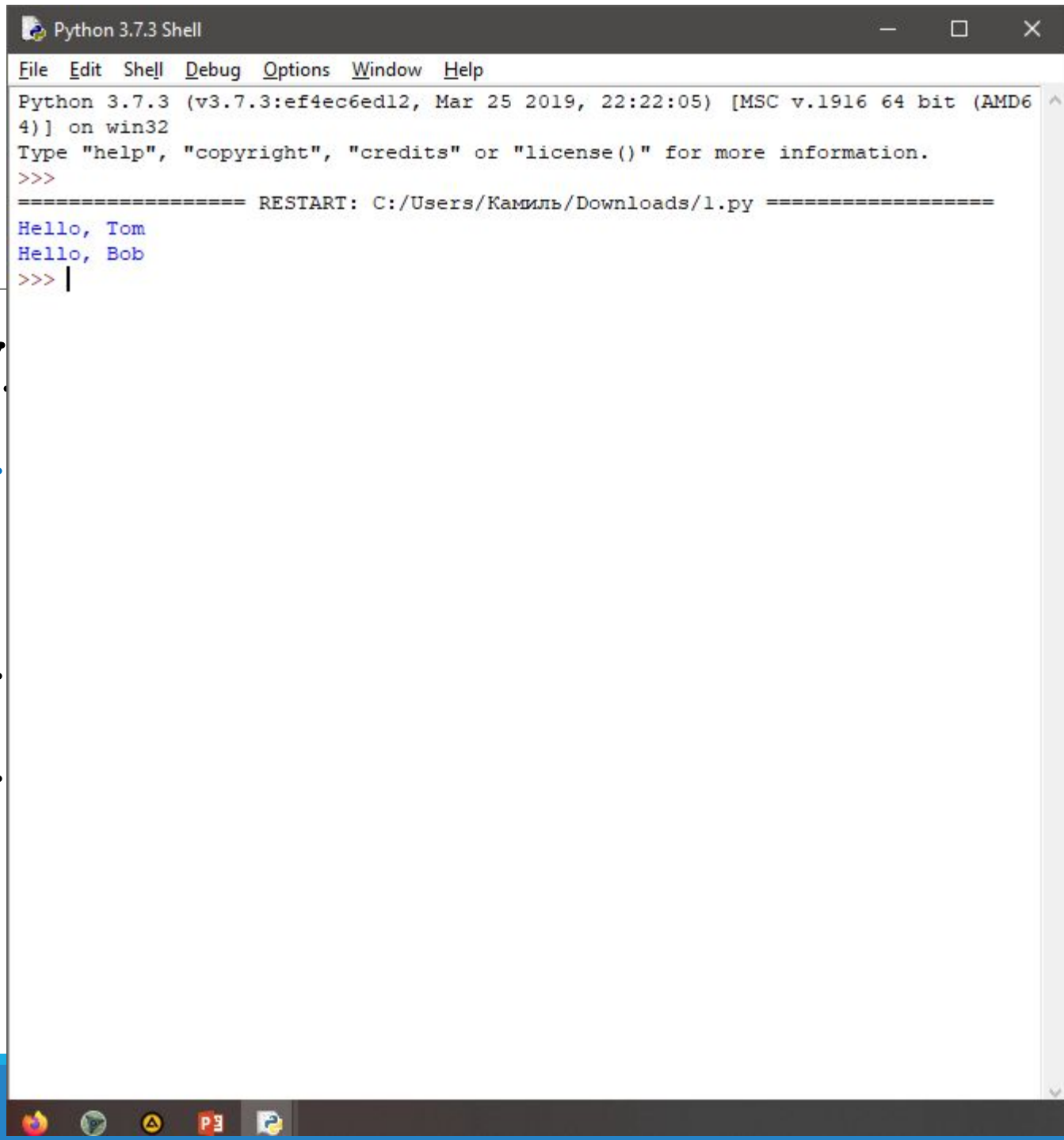


# Значе

```
def say  
    pri
```

```
say_hel
```

```
say_hel
```



```
Python 3.7.3 Shell  
File Edit Shell Debug Options Window Help  
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 22:22:05) [MSC v.1916 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
----- RESTART: C:/Users/Камиль/Downloads/l.py -----  
Hello, Tom  
Hello, Bob  
>>> |
```

# Именованные параметры

---

```
def display_info(name, age):  
    print("Name:", name, "\t", "Age:", age)  
  
display_info("Tom", 22)
```

ИМЕНО

```
def display_name(name, age):  
    print(name, age)
```

```
display_name("Tom", 22)
```

```
display_name("Tom", 22)
```

```
Python 3.7.3 Shell  
File Edit Shell Debug Options Window Help  
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 22:22:05) [MSC v.1916 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
----- RESTART: C:/Users/Камиль/Downloads/l.py -----  
Name: Tom      Age: 22  
Name: Tom      Age: 22  
>>> |
```

```
age: ", age)
```

# Неопределенное количество параметров

---

```
def sum(*params):  
    result = 0  
    for n in params:  
        result += n  
    return result
```

```
sumOfNumbers1 = sum(1, 2, 3, 4, 5)           # 15
```

```
sumOfNumbers2 = sum(3, 4, 5, 6)           # 18
```

```
print(sumOfNumbers1)
```

```
print(sumOfNumbers2)
```



# Возвращение результата

---

```
def exchange(usd_rate, money):  
    result = round(money/usd_rate, 2)  
    return result
```

```
result1 = exchange(60, 30000)
```

```
print(result1)
```

```
result2 = exchange(56, 30000)
```

```
print(result2)
```

```
result3 = exchange(65, 30000)
```

```
print(result3)
```

# Возвращение результата

---

```
def create_default_user():
```

```
    name = "Tom"
```

```
    age = 33
```

```
    return name, age
```

```
user_name, user_age = create_default_user()
```

```
print("Name:", user_name, "\t Age:", user_age)
```

# Функция main

---

```
def main():  
    say_hello("Tom")  
    usd_rate = 56  
    money = 30000  
    result = exchange(usd_rate, money)  
    print("К выдаче", result, "долларов")
```

# Функция main (продолжение)

---

```
def say_hello(name):  
    print("Hello,", name)  
  
def exchange(usd_rate, money):  
    result = round(money/usd_rate, 2)  
    return result  
  
# Вызов функции main  
main()
```

# Область видимости переменных

---

# Глобальные переменные

---

```
name = "Tom"
```

```
def say_hi():  
    print("Hello", name)
```

```
def say_bye():  
    print("Good bye", name)
```

```
say_hi()
```

```
say_bye()
```

# Локальные переменные

---

```
def say_hi():  
    name = "Sam"  
    surname = "Johnson"  
    print("Hello", name, surname)
```

```
def say_bye():  
    name = "Tom"  
    print("Good bye", name)
```

```
say_hi()
```

```
say_bye()
```

# Локальные переменные

---

```
name = "Tom"
```

```
def say_hi():  
    print("Hello", name)
```

```
def say_bye():  
    name = "Bob"  
    print("Good bye", name)
```

```
say_hi()    # Hello Tom  
say_bye()   # Good bye Bob
```



# Глобальные переменные

---

```
def say_bye():  
    global name  
    name = "Bob"  
    print("Good bye", name)
```

# Глобальные переменные

---

```
x = 10
```

```
def foo():  
    print(x)  
    x += 1
```

```
foo()
```

```
#Ошибка!!
```

```
x = 10
```

```
def foo():  
    global x  
    print(x)  
    x += 1
```

```
foo()
```

# Глобальные переменные

---

```
PI = 3.14
```

```
# вычисление площади круга
```

```
def get_circle_square(radius):
```

```
    print("Площадь круга с радиусом", radius, "равна",  
PI * radius * radius)
```

```
get_circle_square(50)
```

# Модули

---

# Модуль

---

Модуль в языке Python представляет отдельный файл с кодом, который можно повторно использовать в других программах.

Для создания модуля необходимо создать собственно файл с расширением \*.py, который будет представлять модуль. Название файла будет представлять название модуля. Затем в этом файле надо определить одну или несколько функций.

# Модуль (пример)

---

#файл account.py

```
def calculate_income(rate, money, month):  
    if money <= 0:  
        return 0  
  
    for i in range(1, month+1):  
        money = round(money + money * rate / 100 / 12, 2)  
    return money
```

# Модуль (пример)

---

```
#файл bank.py
```

```
import account
```

```
rate = int(input("Введите процентную ставку: "))
```

```
money = int(input("Введите сумму: "))
```

```
period = int(input("Введите период ведения счета в месяцах: "))
```

```
result = account.calculate_income(rate, money, period)
```

```
print("Параметры счета:\n", "Сумма: ", money, "\n", "Ставка: ", rate, "\n",  
      "Период: ", period, "\n", "Сумма на счете в конце периода: ", result)
```

# Настройка пространства имен

---

```
import account as acc
```

```
#.....
```

```
result = acc.calculate_income(rate, money, period)
```



# Настройка пространства имен

---

```
from account import calculate_income
```

```
#.....
```

```
result = calculate_income(rate, money, period)
```

# Настройка пространства имен

---

```
from account import *
```

```
#.....
```

```
result = calculate_income(rate, money, period)
```

# Имя модуля

---

```
#файл account.py
```

```
def calculate_income(rate, money, month):
```

```
#...
```

```
#функция main для запуска модуля, как программы
```

```
def main():
```

```
#.....
```

```
if __name__ == "__main__":
```

```
#два подчеркивания с обеих сторон
```

```
    main()
```

# Обработка ИСКЛЮЧЕНИЙ

---

# Исключение

---

```
string = input("Введите число: ")  
number = int(string)  
print(number)
```

#что будет если ввести буквы?

#исключение ValueError

# Обработка исключений

---

`try:`

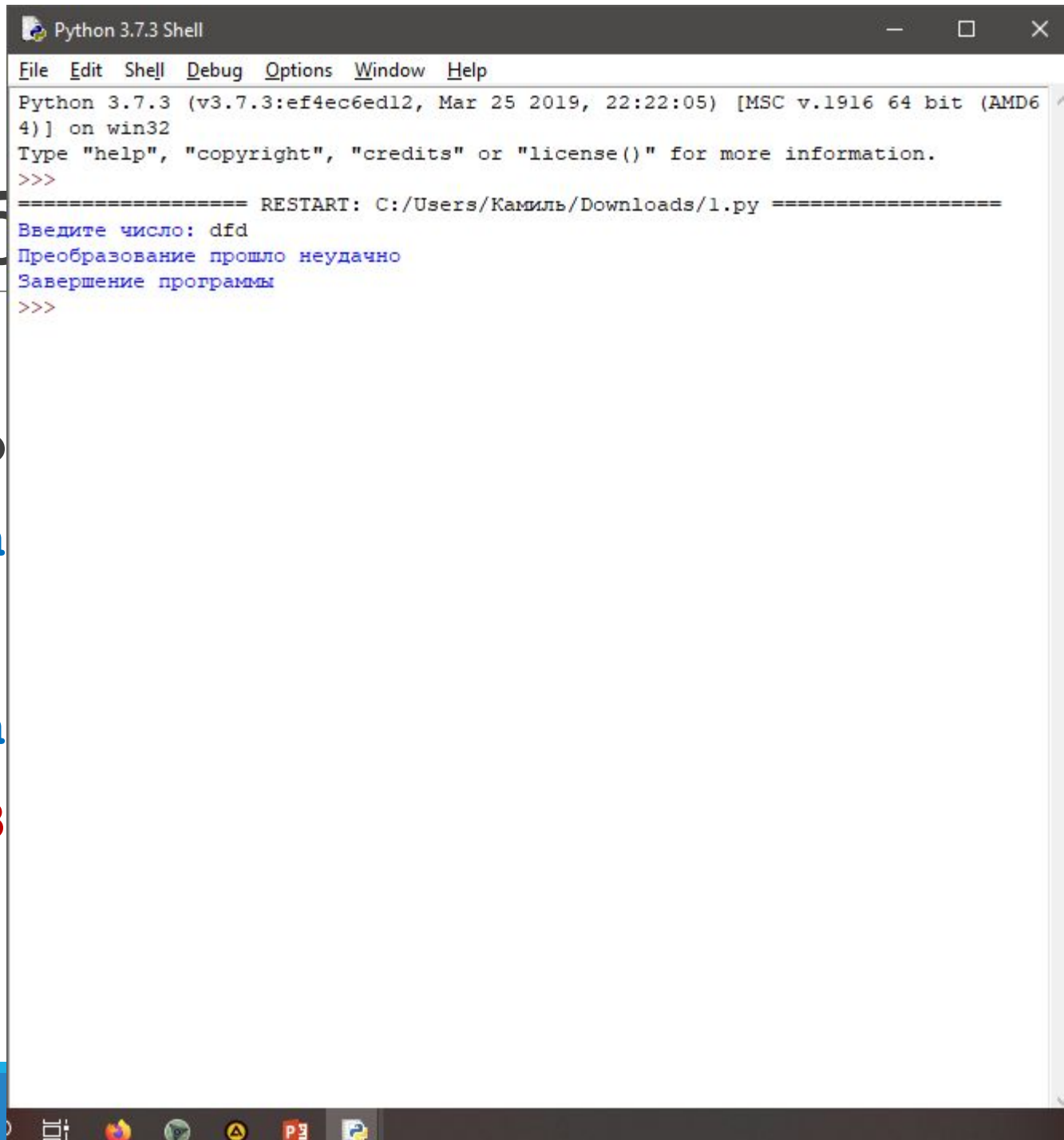
инструкции

`except [Тип_исключения] :`

инструкции

# Обработ

```
try:  
    numb  
    prin  
except:  
    prin  
print("3
```



) )

ЧНО")

# Обработка исключений

---

```
try:
    number1 = int(input("Введите первое число: "))
    number2 = int(input("Введите второе число: "))
    print("Результат деления:", number1/number2)
except ValueError:
    print("Преобразование прошло неудачно")
except ZeroDivisionError:
    print("Попытка деления числа на ноль")
except Exception:
    print("Общее исключение")
print("Завершение программы")
```



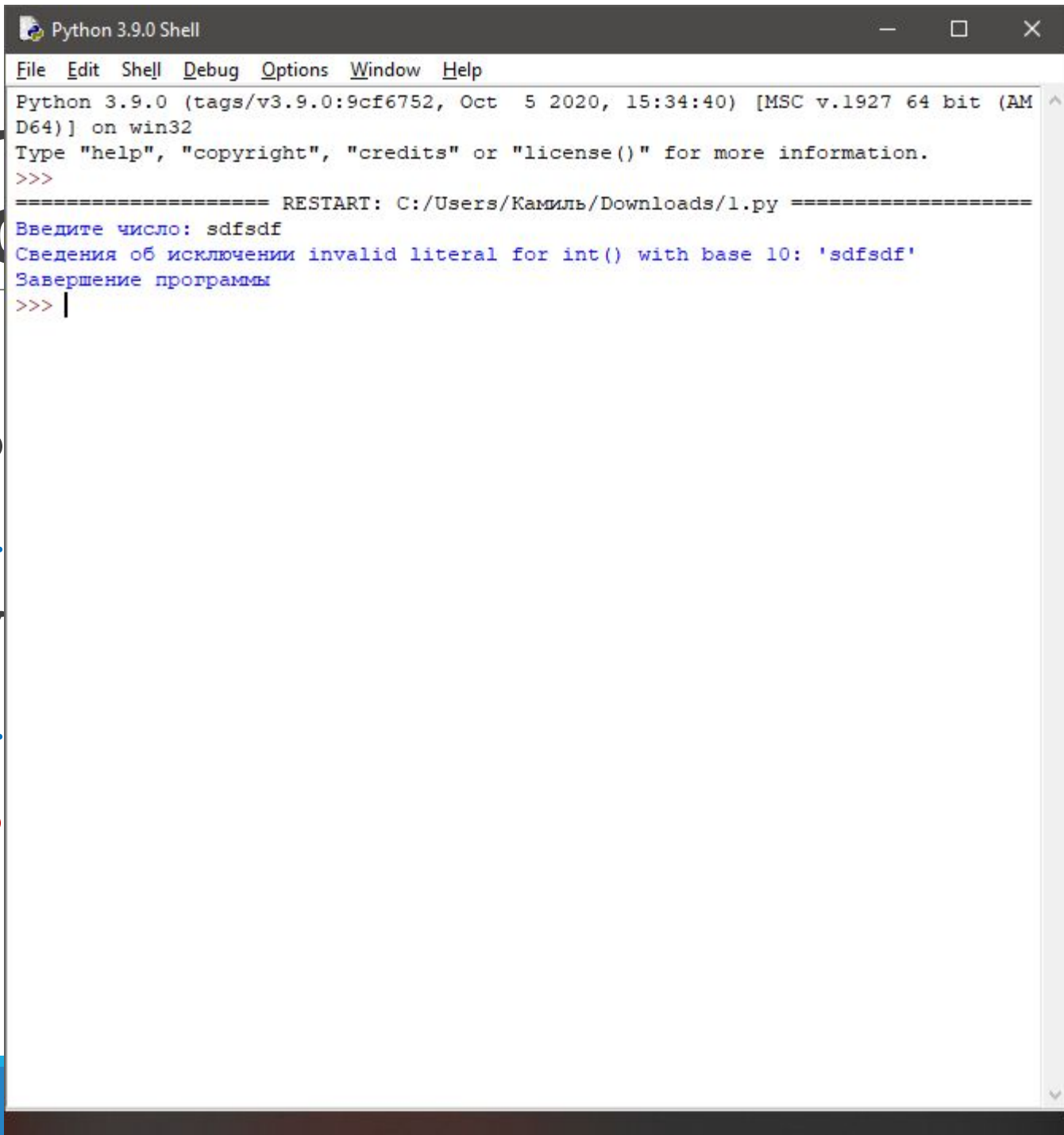
# Блок finally

---

```
try:
    number = int(input("Введите число: "))
    print("Введенное число:", number)
except ValueError:
    print("Не удалось преобразовать число")
finally:
    print("Блок try завершил выполнение")
print("Завершение программы")
```

# Получ ИСКЛЮ

```
try:  
    numb  
    prin  
except V  
    prin  
print("3
```



" ) )

# Генер

try:

```
    number1
```

```
    number2
```

```
    if number
```

```
        rais
```

```
    print ("I
```

```
except Value
```

```
    print ("I
```

```
except Excep
```

```
    print (e)
```

```
print ("Завер
```

```
Python 3.9.0 Shell
File Edit Shell Debug Options Window Help
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Камиль/Downloads/l.py =====
Введите первое число: 2
Введите второе число: 0
Второе число не должно быть равно 0
Завершение программы
>>> |
```

авно 0")  
ber2)