

# ИСПОЛЬЗОВАНИЕ КОМАНДЫ INPUT С РАЗНЫМИ ТИПАМИ ДААННЫХ

---

 Модуль 2. Занятие 10.



# Тема и цель

---

## Тема:

- Команда input с разными типами данных

## Цель занятия:

- Изучение способов работы с командой input



# Глоссарий

---

- **Команда (в программировании)** — это указание компьютерной программе действовать как некий интерпретатор для решения задачи.



# Подтемы

---

1. Что такое **список**?
2. Каким образом можно выводить данные на консоль?
3. Для чего, чаще всего, используется логическая конструкция **if - else**?
4. Какие типы данных бывают в языке Python?
5. За какие значения отвечает тип данных **int**?



# Вспомним

---

Вспомним. В данной программе мы вызываем встроенную функцию `print()`. Важно — запомнить, что у неё есть имя, по которому к ней можно обратиться, плюс она принимает данные и обрабатывает их.

The screenshot shows a code editor interface. At the top, there is a toolbar with buttons for 'Run' (green), 'Debug' (blue), 'Stop' (red), 'Share' (orange), 'Save' (blue), and a blue button with a brace icon. Below the toolbar, the file name 'main.py' is displayed. The code editor shows a single line of Python code: `1 print("Synergy!")`. The output console at the bottom shows the text 'Synergy!'.

```
main.py
1 print("Synergy!")
Synergy!
```



# Вспомним

---

Вспомним, что в Python строки заключены в кавычки — допускается использование одинарных, двойных и тройных (рекомендуется их составлять из трёх двойных). Внутри кавычек одного вида можно использовать другой вид кавычек как часть выводимой на экран строки.

```
main.py
1 print('Synergy')
2 print("Synergy")
3 print("""Synergy""")
```

Synergy  
Synergy  
Synergy



# Задание

---

Напишем программу, которая получает от пользователя строку и выводит на экран:

```
main.py
1 a = input()
2 print(a)
3
```

▼ ↗ 📄

```
Synergy start
Synergy start
```



# Задание

---

После запуска программы происходит приостановка выполнения и в терминале появляется приглашение для ввода данных:

```
main.py
1 a = input()
2 print(a)
3
```

The screenshot shows a code editor window titled 'main.py' with three lines of Python code. The first line is '1 a = input()', the second is '2 print(a)', and the third is '3'. The code is color-coded: 'a' is blue, '=' is red, 'input()' is red, 'print' is orange, and 'a' is blue. The 'print(a)' line has green brackets around the 'a'. Below the code editor, there are three icons: a downward arrow, a double arrow pointing up and right, and a clipboard icon. The bottom part of the screenshot is a solid black rectangle, likely representing a terminal window where the program is running.





# Задание

---

У функции `input()` можно задать параметр-строку перед вводом значения — и тогда указанный текст будет сопровождать приглашение к вводу данных:

```
main.py
1 a = input("Введите строку: ")
2 print(a)
3
```

Введите строку: |



# Задание

У функции `input()` можно задать параметр-строку перед вводом значения — и тогда указанный текст будет сопровождать приглашение к вводу данных:

```
main.py
1 a = input("Введите строку: ")
2 print(a)
3
```

Введите строку: |

```
main.py
1 a = input("Введите строку: ")
2 print(a)
3
```

Введите строку: synergy start  
synergy start



# Нельзя

---

Нельзя использовать следующие однобуквенные имена переменных:

- I (большая английская i);
- l (маленькая английская L);
- O.



# Строковый и числовой **ВВОД**

С числами нужно работать отдельно, поскольку они тоже изначально являются строками. Следующий пример показывает, как можно получить информацию числового типа:

```
main.py
1 # Ввод запрашивается и сохраняется в переменной
2 a = input ("Введите число: ")
3
4 # Преобразуем строку в целое число.
5 # функция float() используется вместо int(),
6 # для преобразования пользовательского ввода в десятичный формат,
7 n = int(a)
8
9 # Выводим в консоль переменную
10 print ("Введенное число: ", n)
11
12
```

input

Введите число:

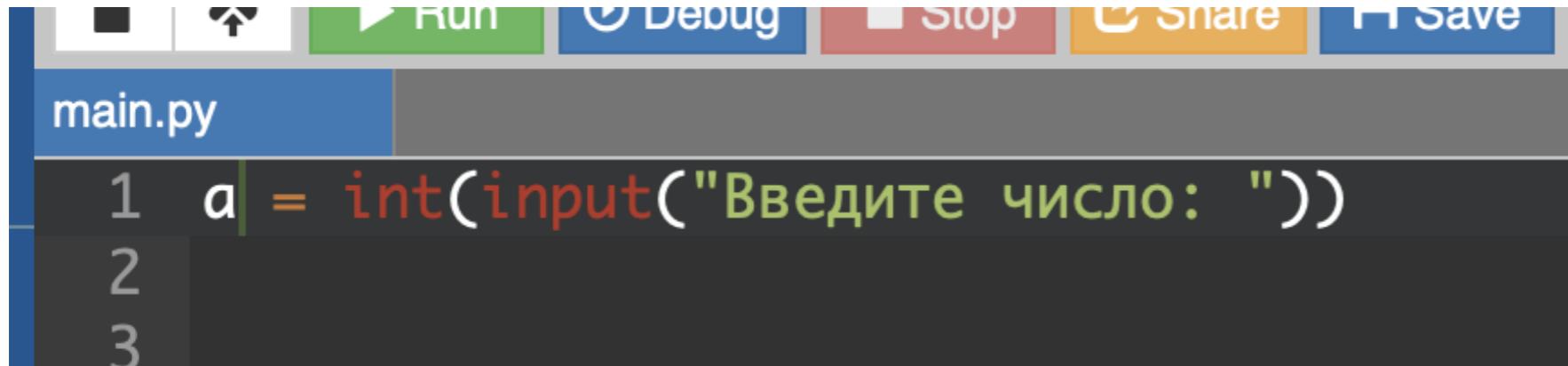
input

Введите число: 3  
Введенное число: 3



# Строковый и числовой **ВВОД**

Данные действия можно упростить и записать в одну строку:



```
main.py  
1 a = int(input("Введите число: "))  
2  
3
```



# Обработка исключений **ввода**

Есть несколько способов, как можно удостовериться в том, что пользователь ввел корректные данные. Один из них — перехватывать все возможные ошибки, которые могут возникнуть. Вот такой код считается небезопасным:

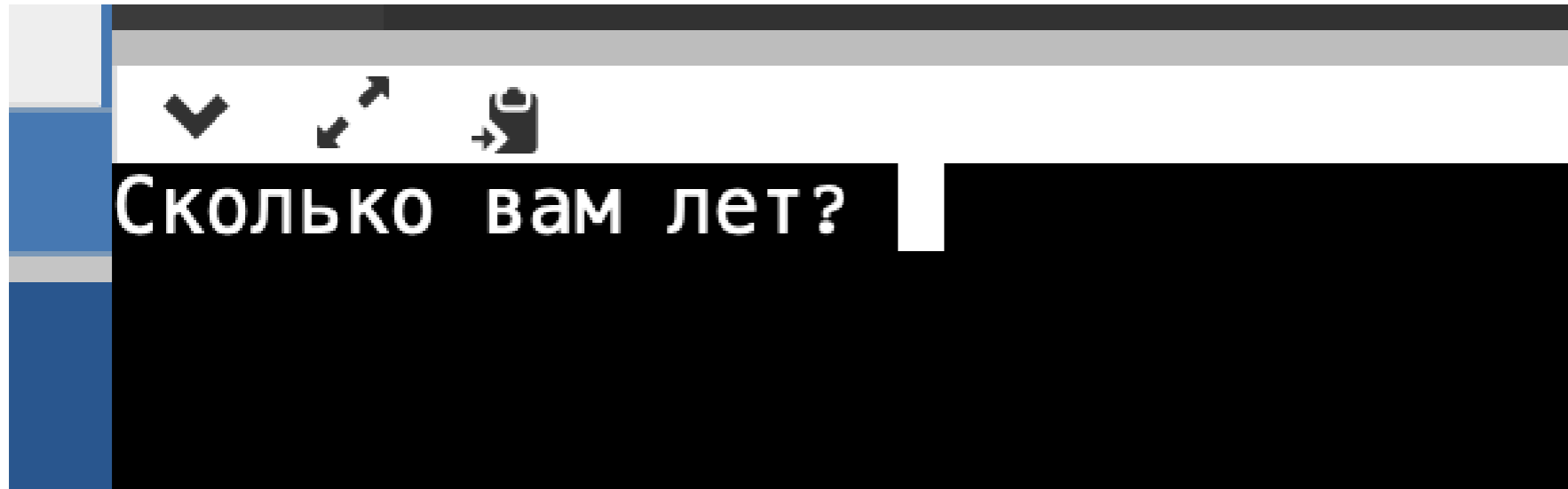
main.py

```
1 a = input("Сколько вам лет? ")
2 b = int(a)
3 print("Ваш возраст ", b)
4
```

# Обработка исключений **ВВОДА**



Запустим его и введем следующее:



# Обработка исключений **ВВОДА**



Запустим его и введем следующее:

```
9  
10  
▼ ↗ 📄  
Сколько вам лет? 18  
Ваш возраст 18
```



# Обработка исключений ввода



Код является небезопасным так как при вводе текстовой информации возникает ошибка

```
main.py
1 a=input("Сколько вам лет?:")
2 b=int(a)
3 print('Ваш возраст', b)
4
5
6
7
```

input

```
Сколько вам лет?:пять
Traceback (most recent call last):
  File "/home/main.py", line 2, in <module>
    b=int(a)
ValueError: invalid literal for int() with base 10: 'пять'
```



# Обработка исключений **ВВОДА**

Вот как можно сделать код безопаснее и обработать ввод:

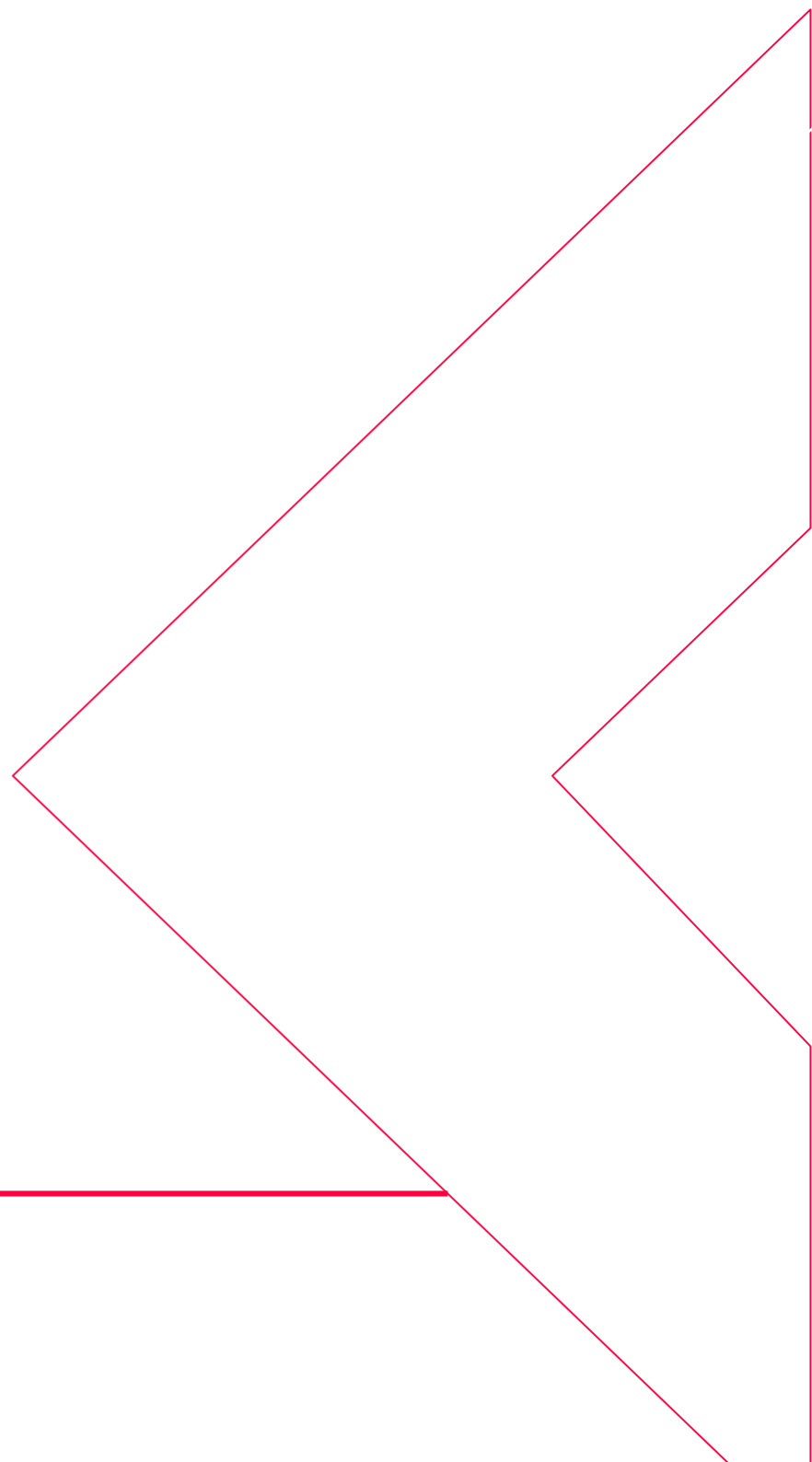
```
1 a = input("Введите свое счастливое число: ")
2 |
3 try:
4     h = int(a)
5     print("Это правильный ввод! Ваше счастливое число: ", h)
6 except ValueError:
7     print("Это не правильный ввод. Это не число вообще! Это строка, попробуйте еще раз.")
8
9
```

```
10
11
12 Введите свое счастливое число: 2
13 Это правильный ввод! Ваше счастливое число: 2
```



# Практические задачи

---





# Задача 1

---

Напишите программу, вычисляющую площадь треугольника по трем сторонам по формуле Герона. На вход программе подаются целые числа, выводом программы должно являться вещественное число, соответствующее площади треугольника.



# Решение

---

Напишем код для решения данной практической задачи и посмотрим на вывод:

```
main.py
1 a=int(input("Введите длину первой стороны:"))
2 b=int(input("Введите длину второй стороны:"))
3 c=int(input("Введите длину третьей стороны:"))
4 p=(a+b+c)/2
5 print((p*(p-a)*(p-b)*(p-c))**0.5)
6
7
```

input

```
Введите длину первой стороны:5
Введите длину второй стороны:6
Введите длину третьей стороны:7
14.696938456699069
```



# Задача 2

---

Напишите программу, принимающую на вход целое число, которая выводит True, если переданное значение попадает в интервал  $(-1, 3] \cup (5, 8) \cup [16, +\infty)$  и False соответственно при непопадании.

Примечание: регистр символов имеет значение. Обратите внимание на скобки!



# Решение

Напишем код для решения данной практической задачи и посмотрим на вывод:

```
main.py
1 a=int(input())
2 print (-1<a<=3 or 5<a<8 or 16<=a)
3
4
5
6
6
True
...Program finished with exit code 0
Press ENTER to exit console.
```

```
main.py
1 a=int(input())
2 print (-1<a<=3 or 5<a<8 or 16<=a)
3
4
5
6
input
4
False
```



# Задача 3

---

Поздоровайтесь с пользователем. Попросите ввести свое имя с клавиатуры.





# Решение

---

main.py

```
1 name = input("Введите свое имя:")
2 print("Добрый день, ", name, ".", sep=" ")
3
4
5
6
```



```
Введите свое имя: Андрей
Добрый день, Андрей.
```



# Вопросы

---

1. Что позволяет выполнить функция **print()**?
2. Что позволяет выполнить функция **int()**?
3. Что позволяет выполнить функция **len()**?
4. Какие нельзя использовать однобуквенные имена переменных?
5. Какие кавычки можно использовать в Python?