

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ФГАОУ высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Основы кроссплатформенного программирования

Отчет по лабораторной работе №2

Основы языка Python

Выполнил студент группы ИТС-б-з-22-1

Дубинин Олег Александрович

« » _____ 2023г.

Подпись студента _____

Работа защищена « » _____ 2023г.

Проверил доцент, кандидат технических
наук, доцент кафедры инфокоммуникаций
Воронкин Роман Александрович

(подпись)

Ставрополь, 2023 г.

Цель работы: исследование процесса установки и базовых возможностей языка Python версии 3.

Ход работы:

<https://github.com/OLEG1232155/2> - репозиторий

Задача 1.

8. Напишите программу (файл `user.py`), которая запрашивала бы у пользователя:

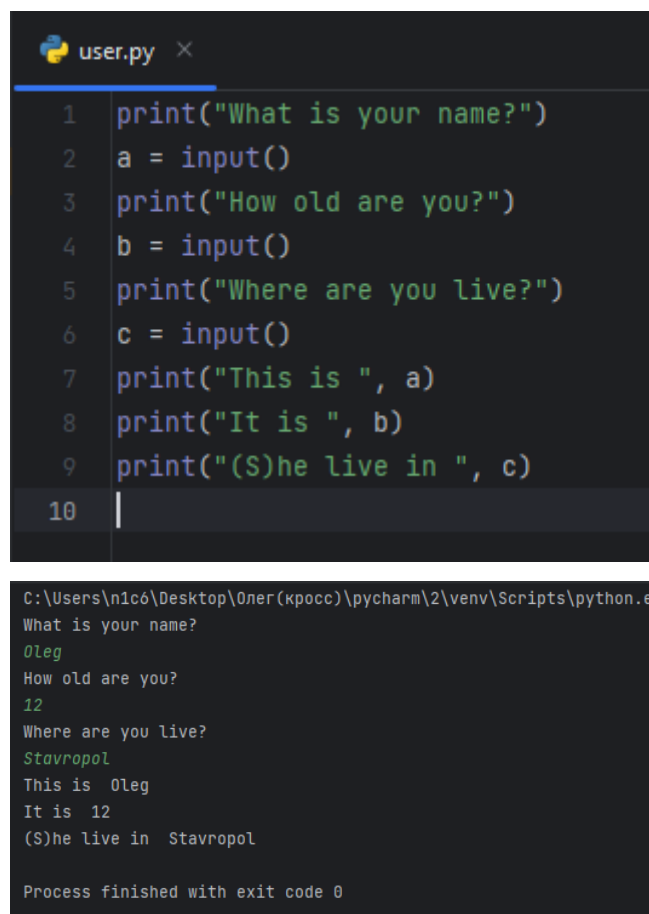
- его имя (например, "What is your name?")
- возраст ("How old are you?")
- место жительства ("Where are you live?")

После этого выводила бы три строки:

```
"This is `имя`"  
"It is `возраст`"  
"(S)he live in `место_жительства`"
```

Вместо `имя`, `возраст`, `место_жительства` должны быть данные, введенные пользователем.

Примечание: можно писать фразы на русском языке, но если вы планируете стать профессиональным программистом, привыкайте к английскому.

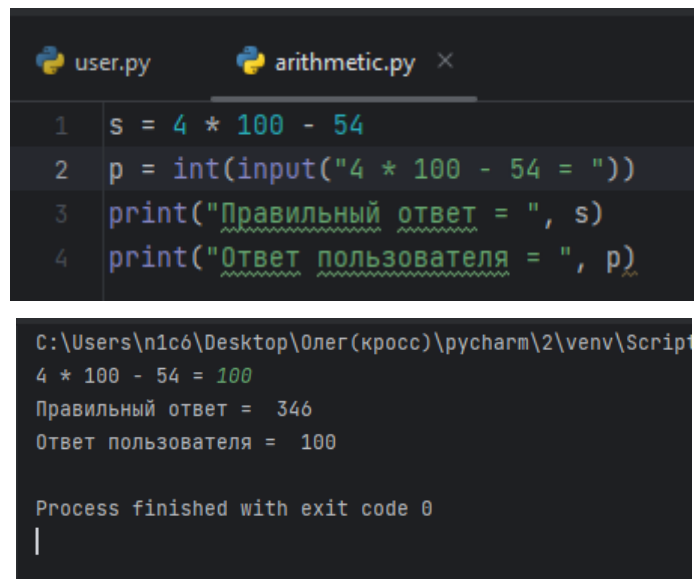


```
user.py x  
1 print("What is your name?")  
2 a = input()  
3 print("How old are you?")  
4 b = input()  
5 print("Where are you live?")  
6 c = input()  
7 print("This is ", a)  
8 print("It is ", b)  
9 print("(S)he live in ", c)  
10  
  
C:\Users\n1c6\Desktop\0ner(кросс)\pycharm\2\venv\Scripts\python.exe  
What is your name?  
Oleg  
How old are you?  
12  
Where are you live?  
Stavropol  
This is Oleg  
It is 12  
(S)he live in Stavropol  
  
Process finished with exit code 0
```

Рисунок 1 – Окно кода для первой задачи

Задача 2.

9. Напишите программу (файл *arithmetic.py*), которая предлагала бы пользователю решить пример $4 * 100 - 54$. Потом выводила бы на экран правильный ответ и ответ пользователя. Подумайте, нужно ли здесь преобразовывать строку в число.



```
user.py arithmetic.py x
1 s = 4 * 100 - 54
2 p = int(input("4 * 100 - 54 = "))
3 print("Правильный ответ = ", s)
4 print("Ответ пользователя = ", p)

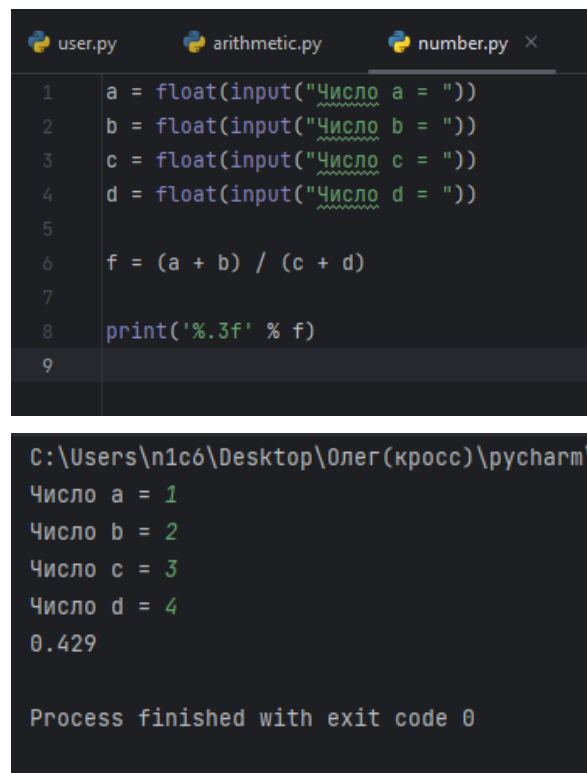
C:\Users\n1c6\Desktop\Олег (кросс)\pycharm\2\venv\Scripts
4 * 100 - 54 = 100
Правильный ответ = 346
Ответ пользователя = 100

Process finished with exit code 0
|
```

Рисунок 2 – Окно кода для второй задачи

Задача 3.

10. Запросите у пользователя четыре числа (файл *numbers.py*). Отдельно сложите первые два и отдельно вторые два. Разделите первую сумму на вторую. Выведите результат на экран так, чтобы ответ содержал две цифры после запятой.



```
user.py arithmetic.py number.py x
1 a = float(input("Число a = "))
2 b = float(input("Число b = "))
3 c = float(input("Число c = "))
4 d = float(input("Число d = "))
5
6 f = (a + b) / (c + d)
7
8 print("%.3f" % f)
9

C:\Users\n1c6\Desktop\Олег (кросс)\pycharm\
Число a = 1
Число b = 2
Число c = 3
Число d = 4
0.429

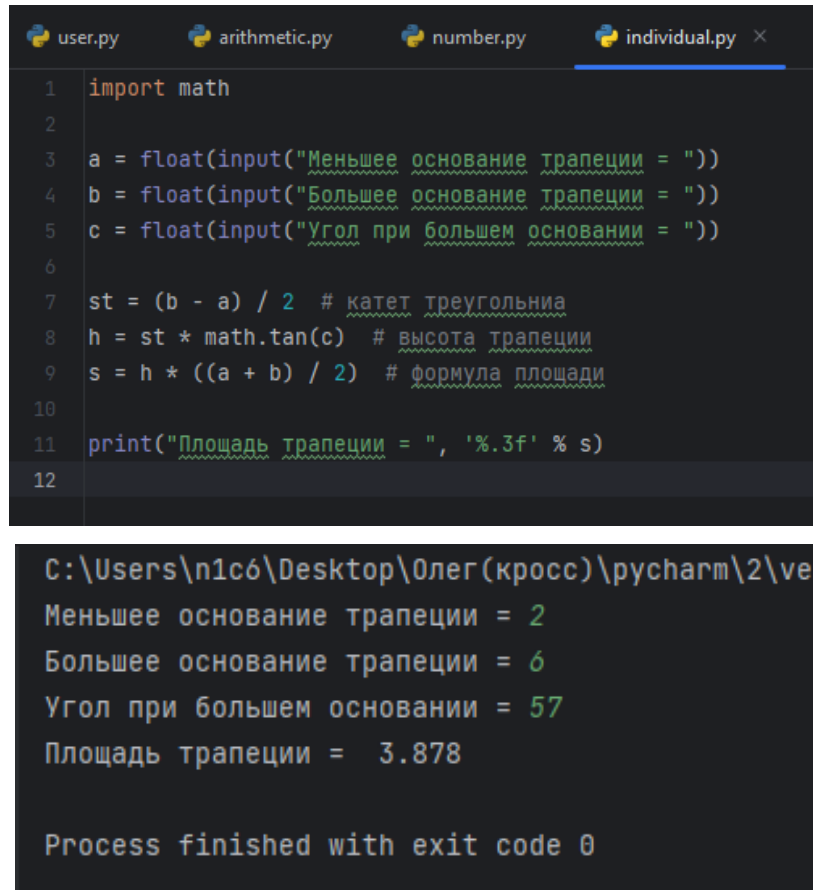
Process finished with exit code 0
```

Рисунок 3 – Окно кода для третьей задачи

Задача 4.

Вариант - 8

11. Напишите программу (файл *individual.py*) для решения индивидуального задания. Вариант индивидуального задания уточните у преподавателя.
8. Даны основания равнобедренной трапеции и угол при большем основании. Найти площадь трапеции.



```
1 import math
2
3 a = float(input("Меньшее основание трапеции = "))
4 b = float(input("Большее основание трапеции = "))
5 c = float(input("Угол при большем основании = "))
6
7 st = (b - a) / 2 # катет треугольника
8 h = st * math.tan(c) # высота трапеции
9 s = h * ((a + b) / 2) # формула площади
10
11 print("Площадь трапеции = ", '%.3f' % s)
12
```

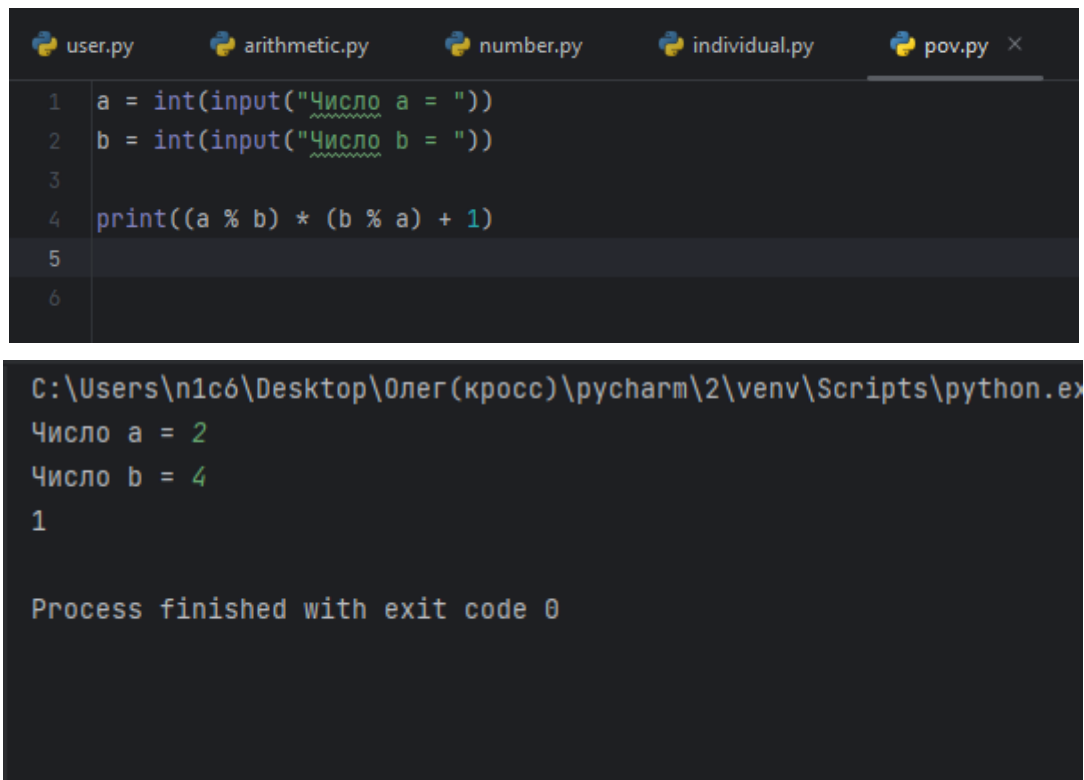
```
C:\Users\n1c6\Desktop\Олег(кросс)\pycharm\2\ve
Меньшее основание трапеции = 2
Большее основание трапеции = 6
Угол при большем основании = 57
Площадь трапеции = 3.878

Process finished with exit code 0
```

Рисунок 4 – Окно кода для четвертой задачи

Повышенная сложность.

8. Даны два целых числа a и b . Если a делится на b или b делится на a , то вывести 1, иначе – любое другое число. Условные операторы и операторы цикла не использовать.



The screenshot shows a code editor with five tabs: user.py, arithmetic.py, number.py, individual.py, and pov.py. The active tab is user.py, which contains the following Python code:

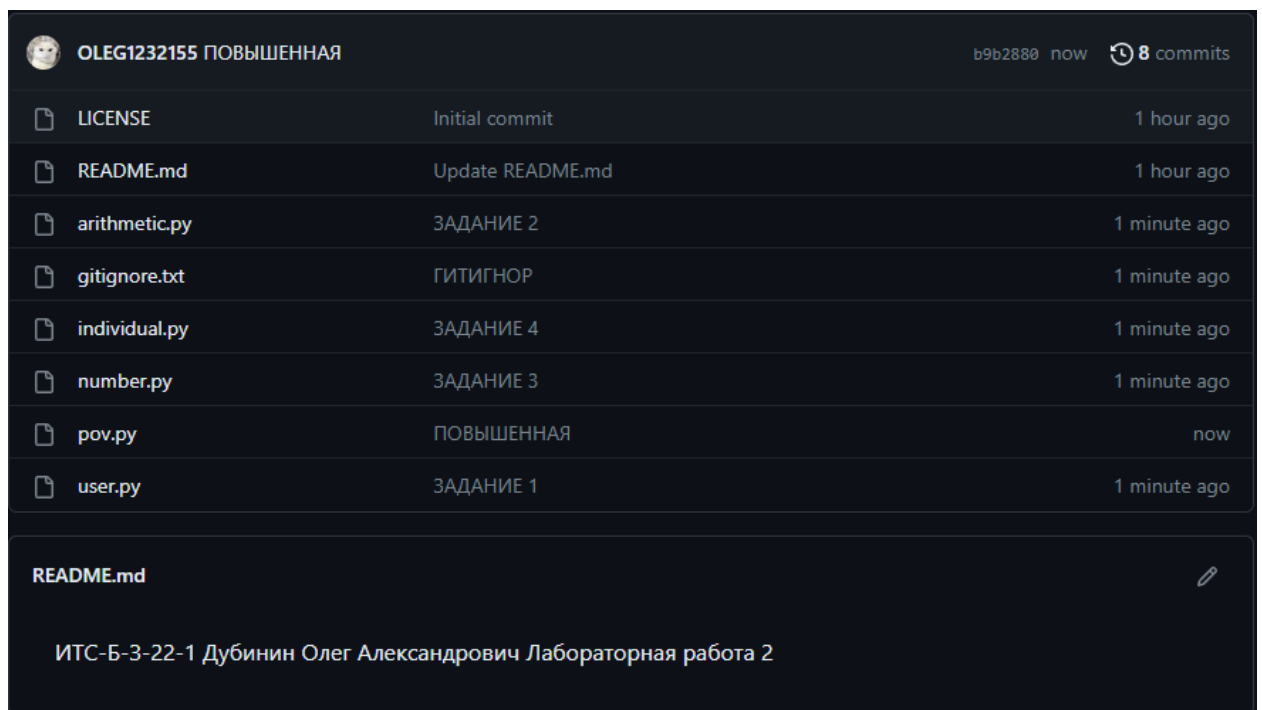
```
1 a = int(input("Число a = "))
2 b = int(input("Число b = "))
3
4 print((a % b) * (b % a) + 1)
5
6
```

Below the code editor, the execution output is displayed in a terminal window. It shows the user inputting '2' for 'a' and '4' for 'b', followed by the output '1'. The process finished with exit code 0.

```
C:\Users\n1c6\Desktop\Олег(кросс)\pycharm\2\venv\Scripts\python.exe
Число a = 2
Число b = 4
1

Process finished with exit code 0
```

Рисунок 5 – Окно кода для задания повышенной сложности



The screenshot shows a GitHub repository page for user OLEG1232155. The repository is named "ПОВЫШЕННАЯ" and has a commit hash of b9b2880. The page shows a list of files and their commit history:

File	Commit Message	Time
LICENSE	Initial commit	1 hour ago
README.md	Update README.md	1 hour ago
arithmetic.py	ЗАДАНИЕ 2	1 minute ago
gitignore.txt	ГИТИГНОР	1 minute ago
individual.py	ЗАДАНИЕ 4	1 minute ago
number.py	ЗАДАНИЕ 3	1 minute ago
pov.py	ПОВЫШЕННАЯ	now
user.py	ЗАДАНИЕ 1	1 minute ago

Below the file list, the README.md file is shown with the following content:

```
ИТС-Б-3-22-1 Дубинин Олег Александрович Лабораторная работа 2
```

Рисунок 6 – Страница GitHub с выполненными задачами

Ответы на контрольные вопросы:

1. Опишите основные этапы установки Python в Windows и Linux.

Для начала скачиваем дистрибутив по ссылке. Запускаем скачанный файл. Выбираем способ установки (install now или customize installation). Отмечаем необходимые опции для установки. Выбираем место установки (доступно при выборе Customize installation). successful

2. В чем отличие пакета Anaconda от пакета Python, скачиваемого с официального сайта?

Anaconda наиболее известна как дистрибутив Python со встроенным в него пакетным менеджером conda. Она позволяет изолировать окружение проекта от системной версии Python, который критически необходим для работы системы. Использование sudo pip считается плохой практикой. Также conda позволяет без проблем переносить окружение с одной машины на другую. Кроме того, если вы что-то сломаете, то с Anaconda вы всегда сможете откатиться на более старую версию окружения. Конечно, если вы позаботитесь о регулярных бэкапах. С системной версией Python это гораздо сложнее и может потребовать переустановки системы.

3. Как осуществить проверку работоспособности пакета Anaconda?

Проверка Anaconda

Для выполнения проверки работоспособности Anaconda необходимо вначале запустить командный процессор с поддержкой виртуальных окружений Anaconda. В Windows это можно сделать выбрав следующий пункт главного меню системы *Пуск → Anaconda3 (64-bit) → Anaconda Prompt*. В появившейся командной строке необходимо ввести

```
> jupyter notebook
```

в результате чего отобразится процесс загрузки веб-среды Jupyter Notebook

4. Как задать используемый интерпретатор языка Python в IDE PyCharm?

Запускаем программу. И он не содержит ни одного файла с текстом программы. Создадим его. Нажимаем правую кнопку мыши, выбираем «New» -> «Python File». Вводим имя файла, например, ex1 и этот файл автоматически

добавляется в наш проект. Здесь мы можем писать наши программы, на Питоне.

5. Как осуществить запуск программы с помощью IDE PyCharm?

Здесь при первом запуске необходимо создать новый проект. Нажимаем «Create New Project». В поле «Location» указывается расположение проекта и его имя.

6. В чем суть интерактивного и пакетного режимов работы Python?

Его суть в удобстве использования языка Питон, а именно его внешний вид и простота, сопровождаемая автоматикой действий, позволяющей пользователям работать с данными программами.

7. Почему язык программирования Python называется языком динамической типизации?

Потому что в Питоне тип переменной определяется непосредственно при выполнении программы.

8. Какие существуют основные типы в языке программирования Python?

В Python типы данных можно разделить на встроенные в интерпретатор (built-in) и не встроенные, которые можно использовать при импортировании соответствующих модулей.

К основным встроенным типам относятся:

1. None (неопределенное значение переменной)

2. Логические переменные (Boolean Type)

3. Числа (Numeric Type)

- int – целое число
- float – число с плавающей точкой
- complex – комплексное число

4. Списки (Sequence Type)

- list – список
- tuple – кортеж
- range – диапазон

5. Строки (Text Sequence Type)

- str

6. Бинарные списки (Binary Sequence Types)

- bytes – байты
- bytearray – массивы байт
- memoryview – специальные объекты для доступа к внутренним данным

объекта через protocol buffer

7. Множества (Set Types)

- set – множество
- frozenset – неизменяемое множество

8.Словари (Mapping Types)

- dict – словарь

9. Как создаются объекты в памяти? Каково их устройство? В чем заключается процесс объявления новых переменных и работа операции присваивания?

Для того, чтобы объявить и сразу инициализировать переменную необходимо написать её имя, потом поставить знак равенства и значение, с которым эта переменная будет создана. Например строка: `b=5` Целочисленное значение 5 в рамках языка Python по сути своей является объектом. Объект, в данном случае – это абстракция для представления данных, данные – это числа, списки, строки и т.п. При этом, под данными следует понимать как непосредственно сами объекты, так и отношения между ними (об этом чуть позже). Каждый объект имеет три атрибута – это идентификатор, значение и тип. Идентификатор – это уникальный признак объекта, позволяющий отличать объекты друг от друга, а значение – непосредственно информация, хранящаяся в памяти, которой управляет интерпретатор.

10. Как получить список ключевых слов в Python?

Список ключевых слов можно получить непосредственно в программе, для этого нужно подключить модуль `keyword` и воспользоваться командой `keyword.kwlist`.

11. Каково назначение функций `id()` и `type()`?

Для того, чтобы посмотреть на объект с каким идентификатором ссылается данная переменная, можно использовать функцию `id()`. Тип переменной можно определить с помощью функции `type()`.

12. Что такое изменяемые и неизменяемые типы в Python.

К неизменяемым (immutable) типам относятся: целые числа (`int`), числа с плавающей точкой (`float`), комплексные числа (`complex`), логические переменные (`bool`), кортежи (`tuple`), строки (`str`) и неизменяемые множества (`frozen set`). К изменяемым (mutable) типам относятся: списки (`list`), множества (`set`), словари (`dict`). Как уже было сказано ранее, при создании переменной, вначале создается объект, который имеет уникальный идентификатор, тип и значение, после этого переменная может ссылаться на созданный объект.

13. Чем отличаются операции деления и целочисленного деления?

Целочисленное деление (`div`) отличается от обычной операции деления тем, что возвращает целую часть частного, дробная часть отбрасывается. Перед выполнением операции оба операнда округляются до целых значений.

14. Какие имеются средства в языке Python для работы с комплексными числами?

Для создания комплексного числа можно использовать функцию `complex(a, b)`, в которую, в качестве первого аргумента, передается действительная часть, в качестве второго – мнимая. Либо записать число в виде $a + bj$.

15. Каково назначение и основные функции библиотеки (модуля) `math`? По аналогии с модулем `math` изучите самостоятельно назначение и основные функции модуля `cmath`.

В стандартную поставку Python входит библиотека `math`, в которой содержится большое количество часто используемых математических функций. Для работы с данным модулем его предварительно нужно импортировать.

16. Каково назначение именных параметров `sep` и `end` в функции `print()`?

`sep` – с помощью этого параметра вы можете указать разделитель строк. А по умолчанию в качестве разделителя используется пробел `end` – этот параметр позволяет указать, что нужно добавить после последней строки. По умолчанию добавляется управляющий символ `'\n'` (перевод строки)

17. Каково назначение метода `format()`? Какие еще существуют средства для форматирования строк в Python? Примечание: в дополнение к рассмотренным средствам изучите самостоятельно работу с f-строками в Python.

Метод `format()`, который определен у строк, позволяет форматировать строку, вставляя в нее на место плейсхолдеров определенные значения. Для вставки в строку используются специальные параметры, которые обрамляются фигурными скобками `{}`.

18. Каким образом осуществить ввод с консоли значения целочисленной и вещественной переменной в языке Python?

Для ввода нужно нажать Enter после завершения набора текста. Обычно Enter добавляет символ новой строки `(\n)`, но не в этом случае. Введенная строка просто будет передана приложению.

Вывод: исследовал процесс установки и базовый возможности языка Python3.