

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития  
Кафедра инфокоммуникаций

**ОТЧЕТ**  
**ПО ЛАБОРАТОРНОЙ РАБОТЕ №2.1**  
**дисциплины «Основы кроссплатформенного программирования»**

Выполнил:  
Хачатрян Владимир Владимирович  
1 курс, группа ИТС-б-о-22-1,  
11.03.02 «Инфокоммуникационные  
технологии и системы связи»,  
направленность (профиль)  
«Инфокоммуникационные системы и  
сети», очная форма обучения

---

(подпись)

Руководитель практики:  
Воронкин Р.А.

---

(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2023 г.

## Основы языка Python.

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

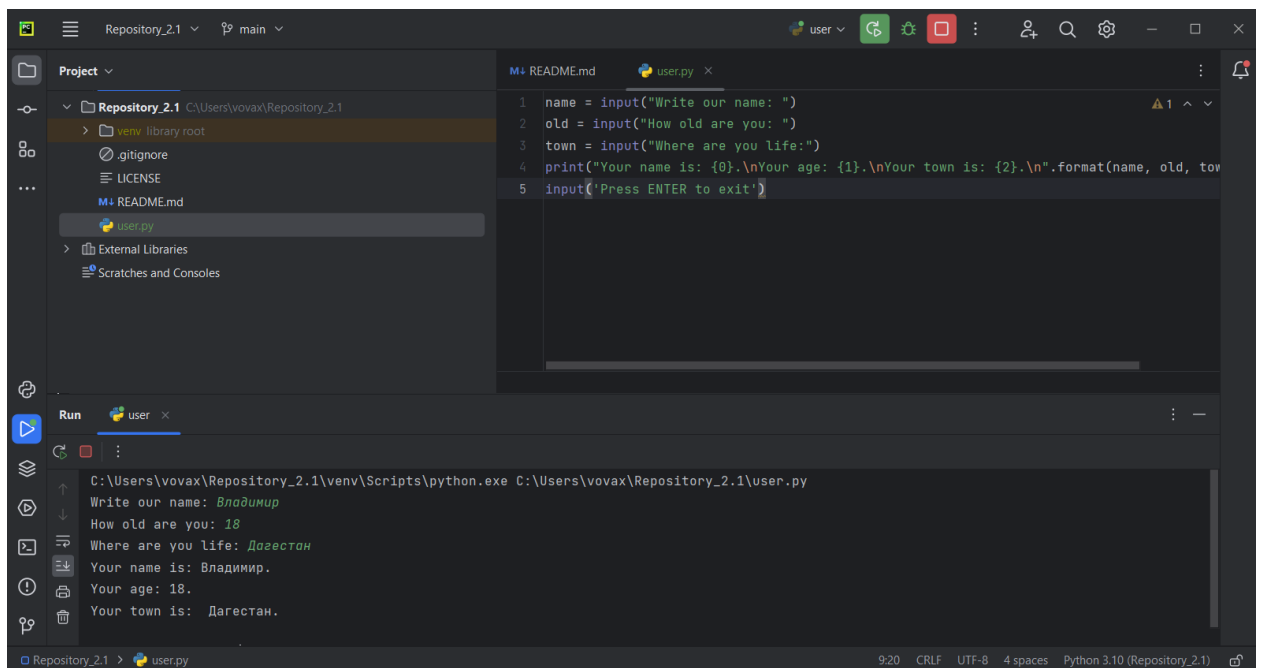
### Ход работы:

Создал репозиторий на GitHub, добавил файл .gitignore с необходимыми правилами. Клонировал репозиторий на свой компьютер.

```
C:\Users\vovax>git clone https://github.com/GitVolodya/Repository_2.1.git
Cloning into 'Repository_2.1'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.

C:\Users\vovax>cd C:\Users\vovax\Repository_2.1
```

### Программа user.py



The screenshot shows a code editor with a file named `user.py` open. The code in the editor is as follows:

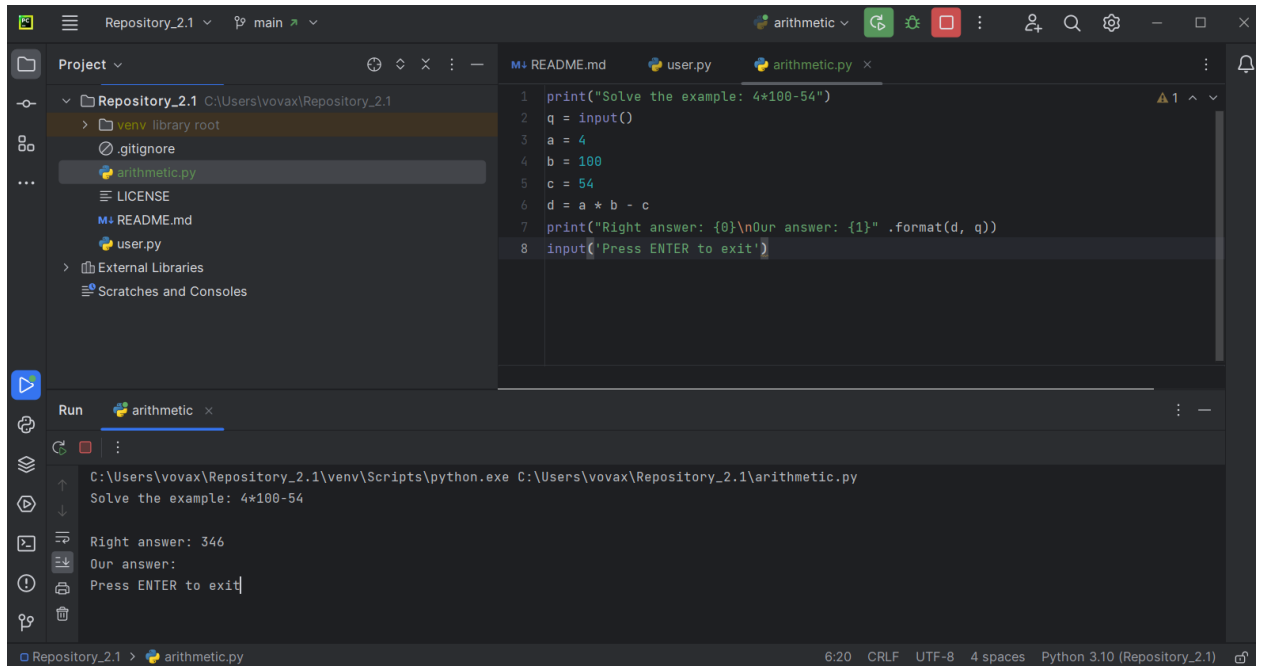
```
1 name = input("Write our name: ")
2 old = input("How old are you: ")
3 town = input("Where are you life:")
4 print("Your name is: {0}.\nYour age: {1}.\nYour town is: {2}.\n".format(name, old, town))
5 input('Press ENTER to exit')
```

Below the editor, the 'Run' console shows the output of the program. The user has entered 'Владимир' for the name, '18' for the age, and 'Дагестан' for the town. The output is:

```
C:\Users\vovax\Repository_2.1\venv\Scripts\python.exe C:\Users\vovax\Repository_2.1\user.py
Write our name: Владимир
How old are you: 18
Where are you life: Дагестан
Your name is: Владимир.
Your age: 18.
Your town is: Дагестан.
```

The status bar at the bottom of the IDE indicates the file is `Repository_2.1 > user.py`, the time is 9:20, and the encoding is UTF-8 with 4 spaces and Python 3.10 syntax.

## Программа arithmetic.py

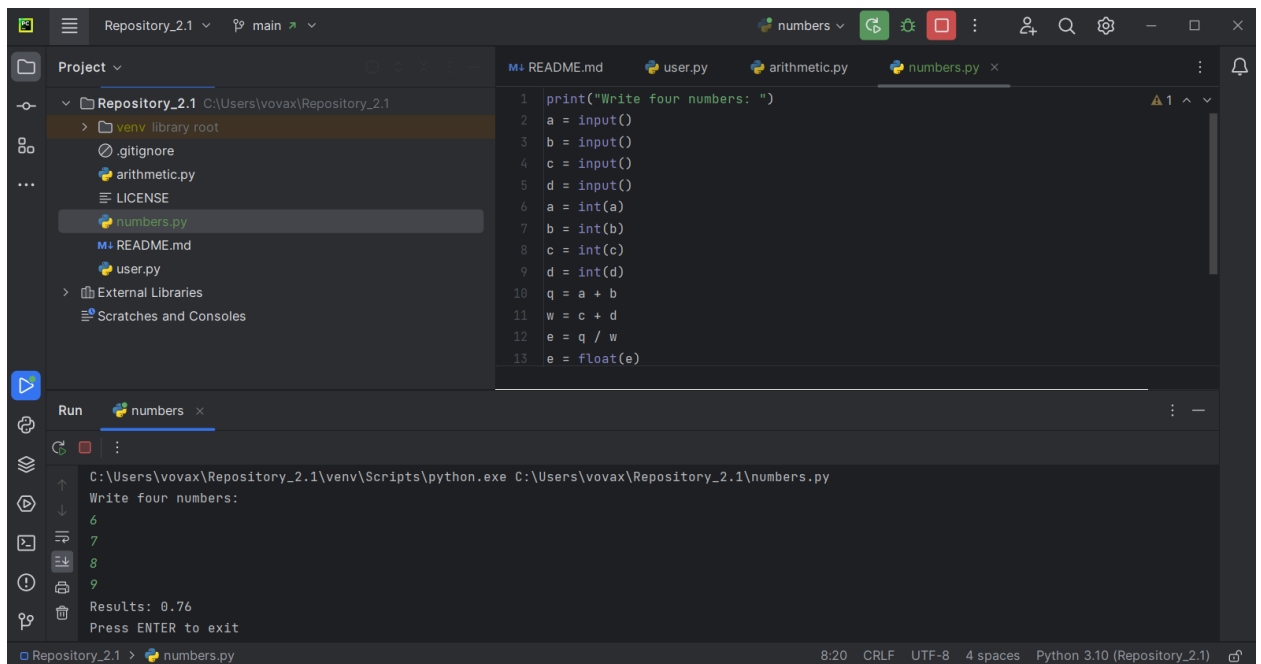


```
1 print("Solve the example: 4*100-54")
2 q = input()
3 a = 4
4 b = 100
5 c = 54
6 d = a * b - c
7 print("Right answer: {0}\nOur answer: {1}" .format(d, q))
8 input('Press ENTER to exit')
```

Run `arithmetic`

```
C:\Users\vovax\Repository_2.1\venv\Scripts\python.exe C:\Users\vovax\Repository_2.1\arithmetic.py
Solve the example: 4*100-54
Right answer: 346
Our answer:
Press ENTER to exit
```

## Программа numbers.py



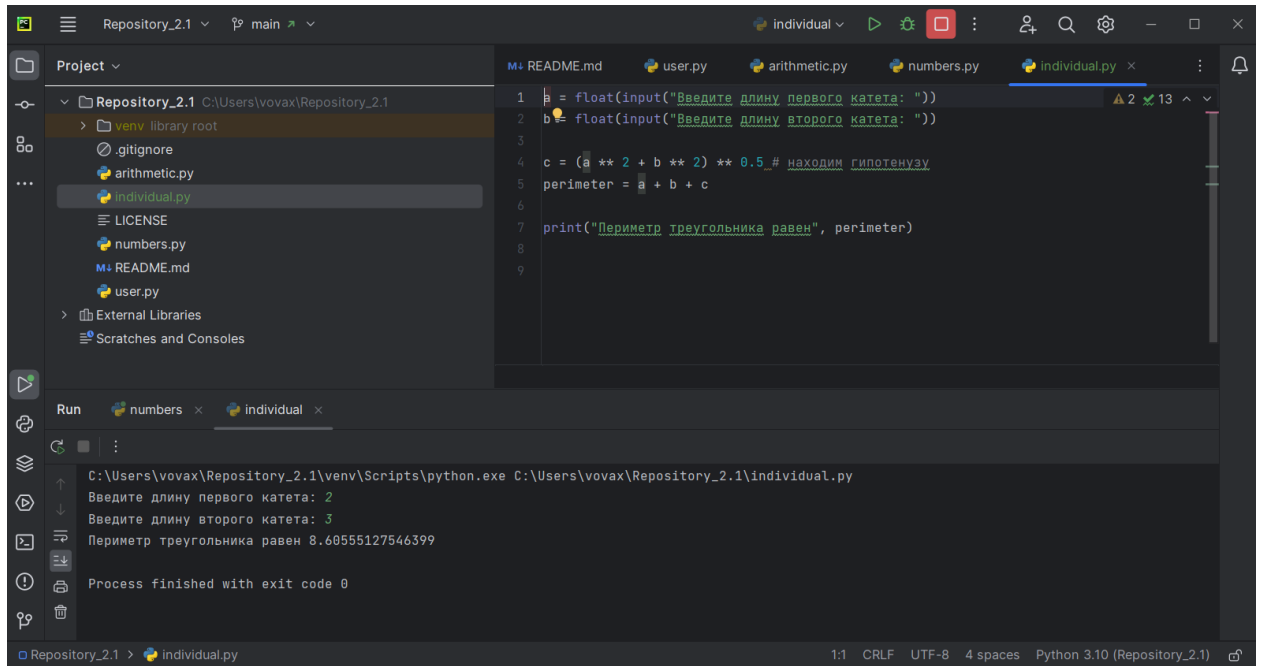
```
1 print("Write four numbers: ")
2 a = input()
3 b = input()
4 c = input()
5 d = input()
6 a = int(a)
7 b = int(b)
8 c = int(c)
9 d = int(d)
10 q = a + b
11 w = c + d
12 e = q / w
13 e = float(e)
```

Run `numbers`

```
C:\Users\vovax\Repository_2.1\venv\Scripts\python.exe C:\Users\vovax\Repository_2.1\numbers.py
Write four numbers:
6
7
8
9
Results: 0.76
Press ENTER to exit
```

## Индивидуальное задание 23 вариант (individual.py)

Даны катеты прямоугольного треугольника. Найти его периметр.



```
1 a = float(input("Введите длину первого катета: "))
2 b = float(input("Введите длину второго катета: "))
3
4 c = (a ** 2 + b ** 2) ** 0.5 # находим гипотенузу
5 perimeter = a + b + c
6
7 print("Периметр треугольника равен", perimeter)
8
9
```

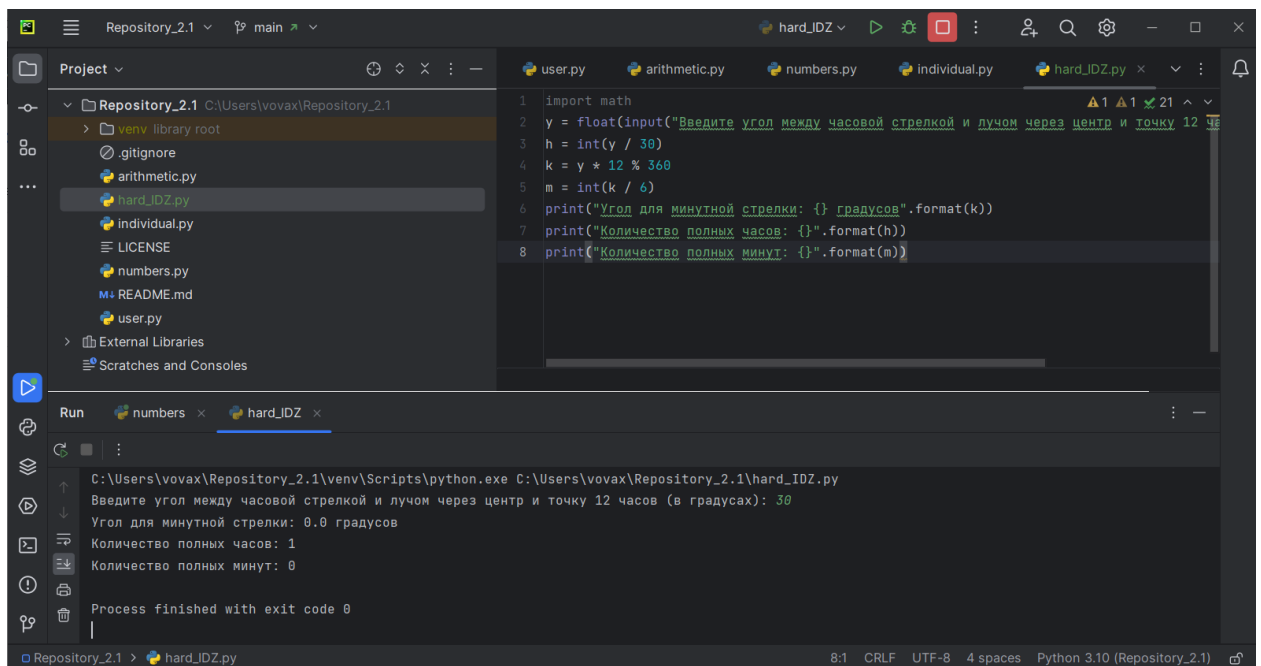
Run numbers x individual x

C:\Users\vovax\Repository\_2.1\venv\Scripts\python.exe C:\Users\vovax\Repository\_2.1\individual.py

Введите длину первого катета: 2  
Введите длину второго катета: 3  
Периметр треугольника равен 8.60555127546399

Process finished with exit code 0

## Задача повышенной сложности 7 вариант (hard\_IDZ)



```
1 import math
2 y = float(input("Введите угол между часовой стрелкой и лучом через центр и точку 12 часов: "))
3 h = int(y / 30)
4 k = y * 12 % 360
5 m = int(k / 6)
6 print("Угол для минутной стрелки: {} градусов".format(k))
7 print("Количество полных часов: {}".format(h))
8 print("Количество полных минут: {}".format(m))
```

Run numbers x hard\_IDZ x

C:\Users\vovax\Repository\_2.1\venv\Scripts\python.exe C:\Users\vovax\Repository\_2.1\hard\_IDZ.py

Введите угол между часовой стрелкой и лучом через центр и точку 12 часов (в градусах): 30  
Угол для минутной стрелки: 0.0 градусов  
Количество полных часов: 1  
Количество полных минут: 0

Process finished with exit code 0

Ссылка на репозиторий: [https://github.com/GitVolodya/Repository\\_2.1.git](https://github.com/GitVolodya/Repository_2.1.git)

## **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).

### *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.

### *3. Как осуществить проверку работоспособности пакета 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 отобразиться 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?

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
- str Строки (Text Sequence Type)
- str Бинарные списки (Binary Sequence Types)
- bytes – байты
- bytearray – массивы байт
- memoryview – специальные объекты для доступа к внутренним данным

Объекта через protocol buffer Множества (Set Types)

- set – множество

- frozenset – неизменяемое множество
- 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. 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 именованных 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?*

Если необходимо вести целочисленное значение можно использовать код: `A = input()` `A = int()`

Если необходимо вести вещественное значение можно использовать код: `A = input()` `A = float()`

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:** в ходе работы я исследовал процесс установки и базовые возможности языка Python3.