

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:  
Евдаков Евгений 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., доцент кафедры  
инфокоммуникаций

---

(подпись)

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

Ставрополь, 2023 г.

**Тема:** Основы языка Python.

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

### **Ход работы:**

**Задание 1.** Создал общедоступный репозиторий на GitHub, в котором использована лицензий MIT и язык программирования Python, также добавил файл .gitignore с необходимыми правилами. Клонировал свой репозиторий на свой компьютер.

```
C:\Users\Gaming-PC>git clone https://github.com/EvgenyEvdakov/Laba_2.1.git
Cloning into 'Laba_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.
```

Рисунок 1. Клонирование репозитория

**Задание 2.** Организовал свой репозиторий в соответствии с моделью ветвления git-flow.

```
C:\Users\Gaming-PC\Laba_2.1>git flow init
Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main] Branch name for "next release" development: [develop]
How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [C:/Users/Gaming-PC/Laba_2.1/.git/hooks]
```

Рисунок 2. Модель ветвления git-flow

**Задание 3.** Создал проект PyCharm в папке репозитория. Добавил новый файл user.py.

**Условие примера:** необходимо, чтобы данная программа запрашивала у пользователя его имя, возраст, где он живет и выводила эти данные,

написанные пользователем на экран.

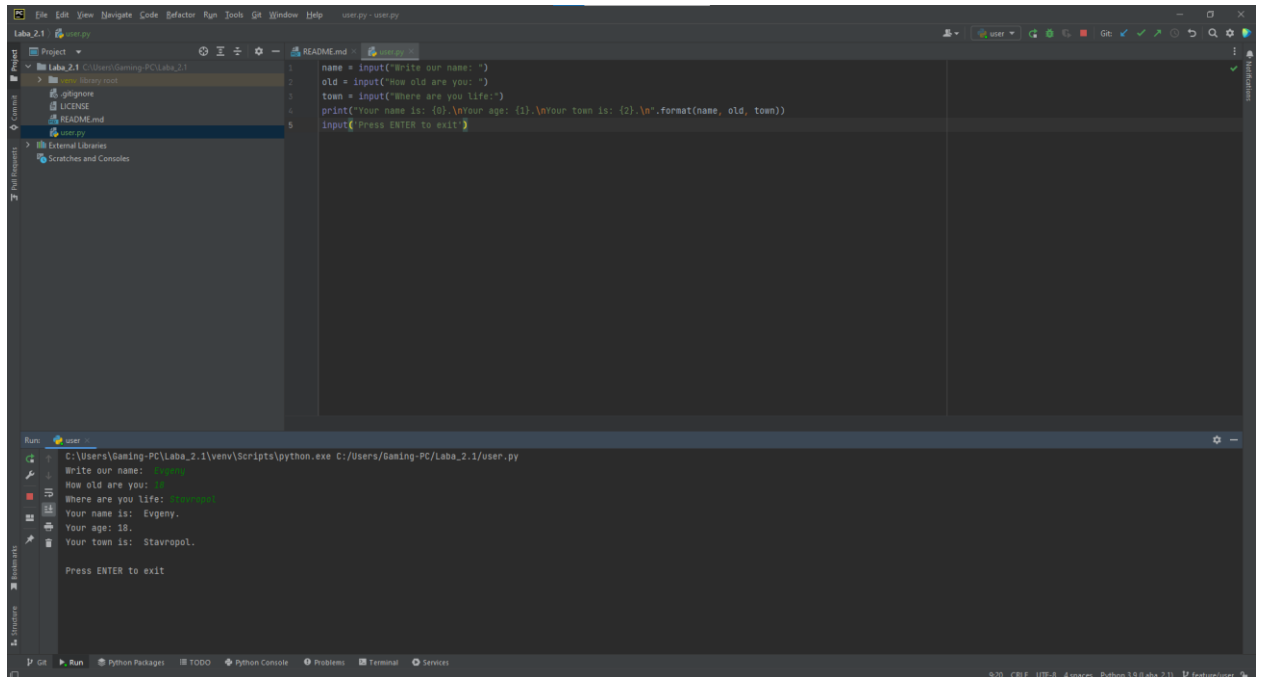


Рисунок 3. Программа user.py

Так как закончил написание программы в ветке feature/user, то необходимо закоммитить изменения и закончить работу с ней с помощью команды в Git: git flow feature finish user.

```
C:\Users\Gaming-PC\Laba_2.1>git add user.py
C:\Users\Gaming-PC\Laba_2.1>git commit -m "user.py"
[feature/user d0775b4] user.py
1 file changed, 5 insertions(+)
create mode 100644 user.py
C:\Users\Gaming-PC\Laba_2.1>git flow feature finish user
Switched to branch 'develop'
Updating 8044b25..d0775b4
Fast-forward
 user.py | 5 +++++
1 file changed, 5 insertions(+)
create mode 100644 user.py
Deleted branch feature/user (was d0775b4).

Summary of actions:
- The feature branch 'feature/user' was merged into 'develop'
- Feature branch 'feature/user' has been locally deleted
- You are now on branch 'develop'
```

Рисунок 4. Завершение работы с feature/user

Написал программу и проверил на работоспособность в файле arithmetic.py. В новой ветке feature/arithmetic от develop.

```
C:\Users\Gaming-PC\Laba_2.1>git flow feature start arifhmetic
Switched to a new branch 'feature/arifhmetic'

Summary of actions:
- A new branch 'feature/arifhmetic' was created, based on 'develop'
- You are now on branch 'feature/arifhmetic'

Now, start committing on your feature. When done, use:

git flow feature finish arifhmetic
```

Рисунок 5. Создание новой ветки

**Задание 4.** Создал проект PyCharm в папке репозитория. Добавил новый файл arifmetic.py.

**Условие примера:** необходимо, чтобы данная программа предлагала бы пользователю решить пример  $4 * 100 - 54$ . Потом выводила бы на экран правильный ответ и ответ пользователя.

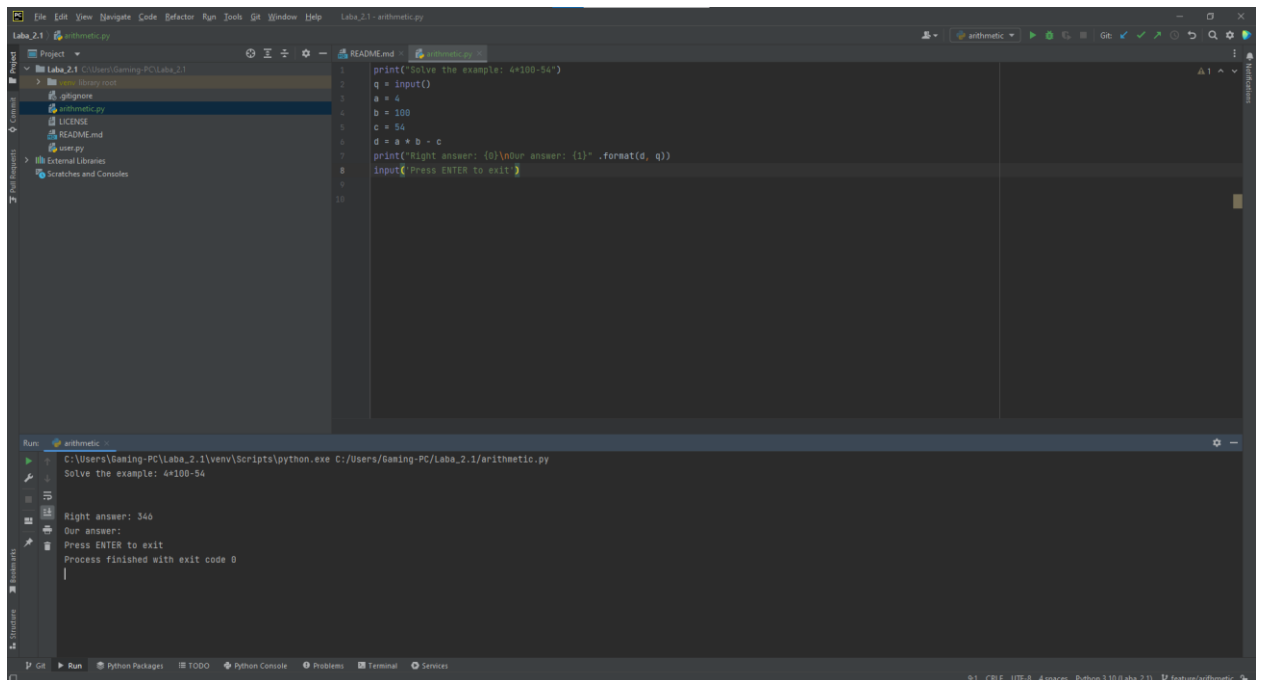


Рисунок 6. Программа arifmetic.py

**Задание 5.** Создал новую ветку `feature/numbers` на основании ветки `develop` и в данной ветке написал программу. Создал новый файл `numbers.py`.

```
C:\Users\Gaming-PC\Laba_2.1>git flow feature start numbers
Switched to a new branch 'feature/numbers'

Summary of actions:
- A new branch 'feature/numbers' was created, based on 'develop'
- You are now on branch 'feature/numbers'

Now, start committing on your feature. When done, use:

git flow feature finish numbers
```

Рисунок 7. Создание ветки feature/numbers

**Условие примера:** Необходимо, чтобы данная программа запросила у пользователя четыре числа, далее отдельно складывала первые два числа и вторые два числа, разделила первую сумму на вторую и вывела результат на экран (ответ должен содержать 2 цифры после запятой).

The screenshot shows a code editor with the following Python code in `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)
14 print("Results: %.2f" % e)
15 input("Press ENTER to exit ")
16
```

The Run console at the bottom shows the execution output:

```
Write four numbers:
Results: 0.50
Press ENTER to exit
Process finished with exit code 0
```

Рисунок 8. Программа numbers.py

## Задание 6. Создал новую ветку feature/individual

```
C:\Users\Gaming-PC\Laba_2.1>git merge feature/arithmetric
Already up to date.

C:\Users\Gaming-PC\Laba_2.1>git flow feature start individual
Switched to a new branch 'feature/individual'

Summary of actions:
- A new branch 'feature/individual' was created, based on 'develop'
- You are now on branch 'feature/individual'

Now, start committing on your feature. When done, use:

git flow feature finish individual
```

Рисунок 9. Создание ветки feature/individual

## Индивидуальное задание

## Вариант 9 (по списку группы)

### Условие задания:

9. Треугольник задан координатами своих вершин. Найти периметр и площадь треугольника.

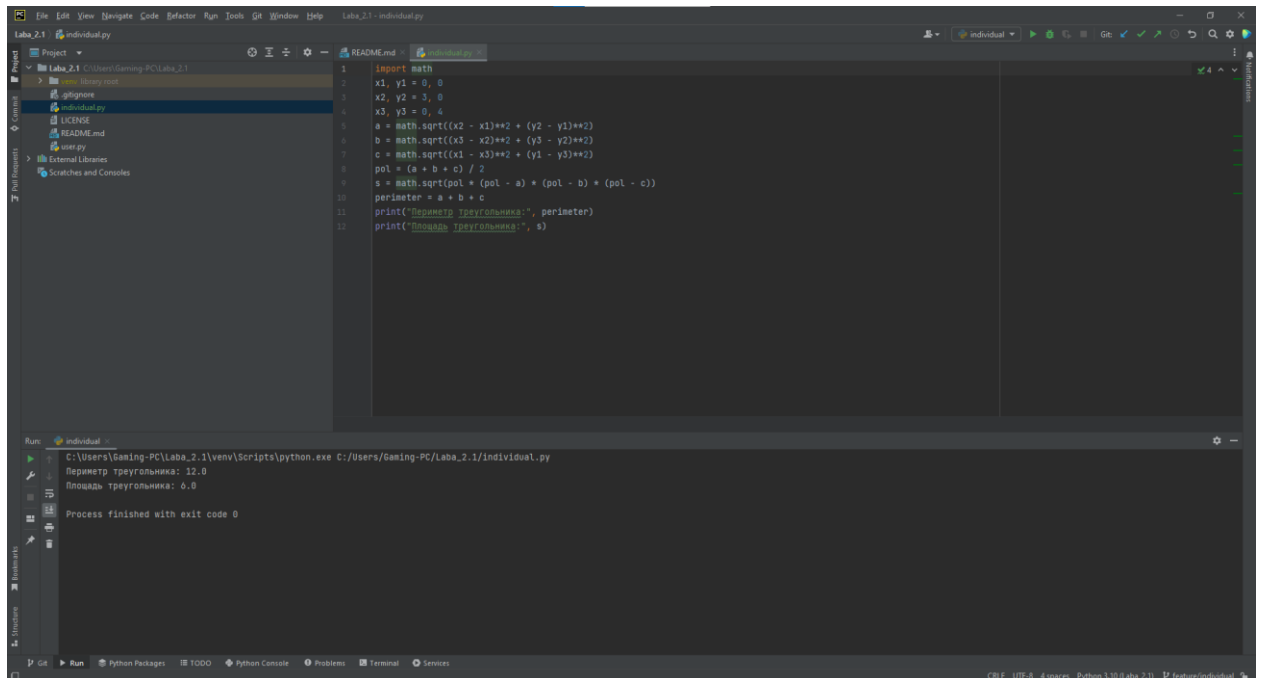


Рисунок 10. Программа individual.py

### Задание 7. Создал новую ветку feature/hardtask

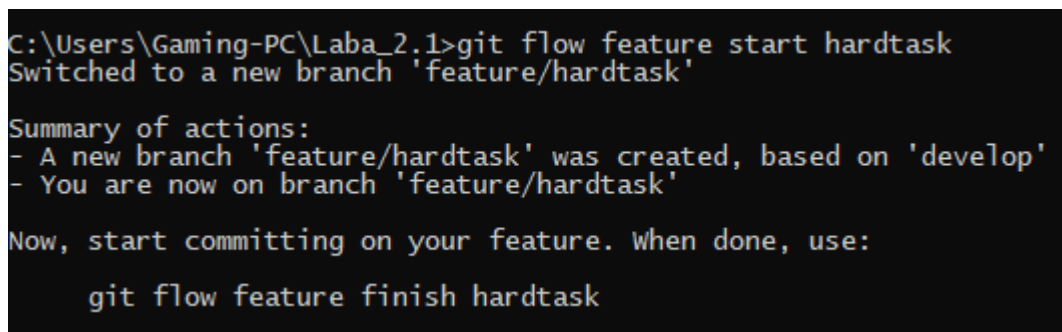


Рисунок 11. Создание ветки feature/hardtask

## Усложненное индивидуальное задание

### Вариант 7 (по списку подгруппы т.к там всего 8 заданий)

### Условие задания:

7. Часовая стрелка образует угол  $\gamma$  с лучом, проходящим через центр и через точку, соответствующую 12 часам на циферблате,  $0 < \gamma \leq 2\pi$ . Определить значение угла для минутной стрелки, а также количество полных часов и полных минут.

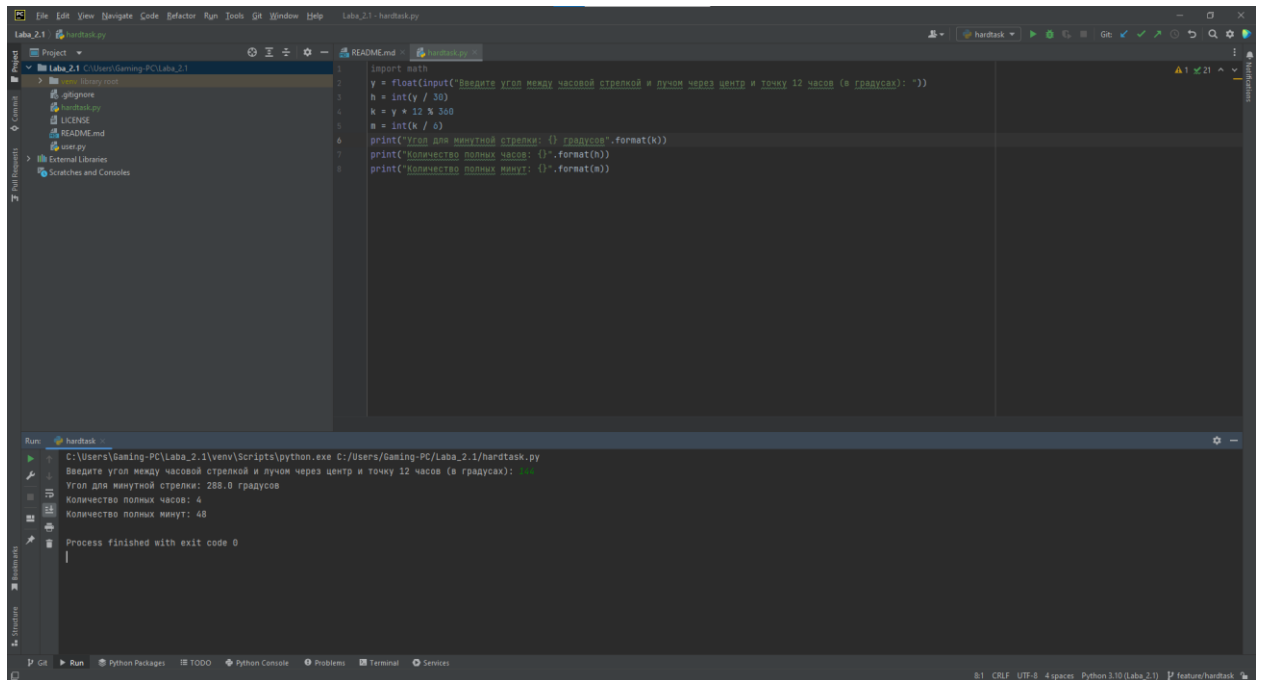


Рисунок 12. Программа hardtask.py

Далее слил все ветки в main и пропустил, тем самым отправил изменения на удаленный репозиторий.

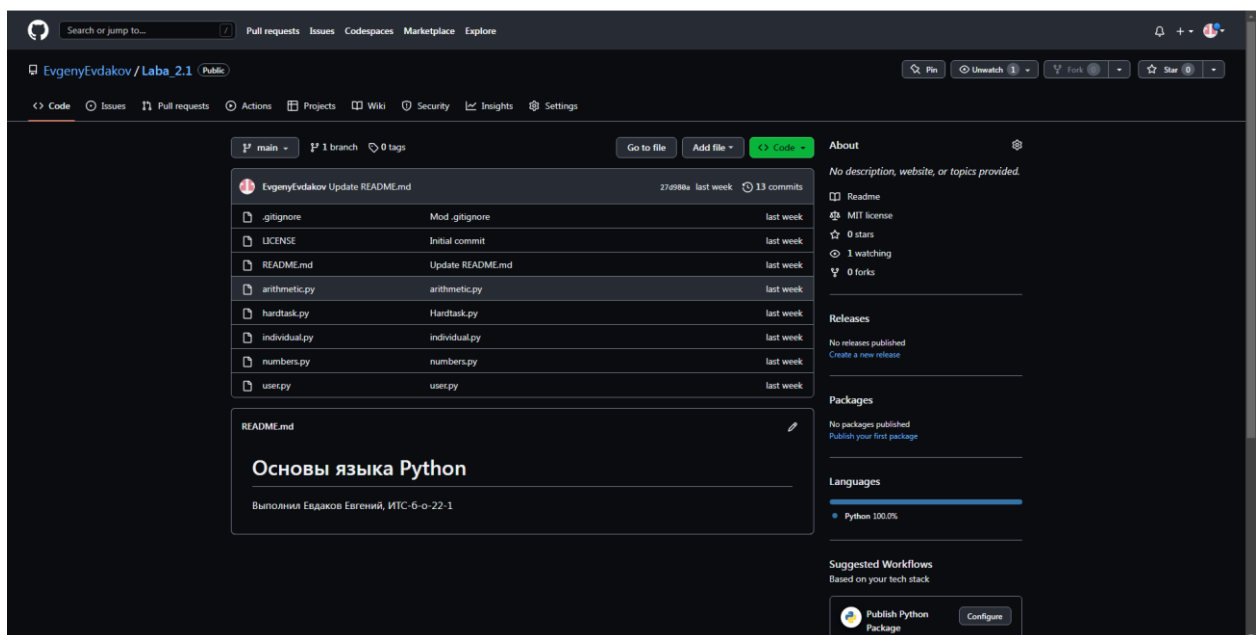


Рисунок 13. Удаленный репозиторий

Ссылка: [https://github.com/EvgenyEvdakov/Laba\\_2.1](https://github.com/EvgenyEvdakov/Laba_2.1)

### 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 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?**

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
- 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. 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 ввести целочисленное значение можно использовать код: `A = input()` `A = int()`

Если необходимо ввести 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:** исследовал процесс установки и базовые возможности языка Python3.