

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 РАБОТЕ №2.1.
дисциплины «Основы кроссплатформенного программирования»

Выполнила:
Дудова Мира Сергеевна
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., канд. тех. наук, доцент,
доцент кафедры инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: основы языка Python

Цель работы: исследовать базовые возможности системы контроля версий Git для работы с локальными репозиториями.

Порядок выполнения работы:

1. Создала новый репозиторий со всеми необходимыми настройками.

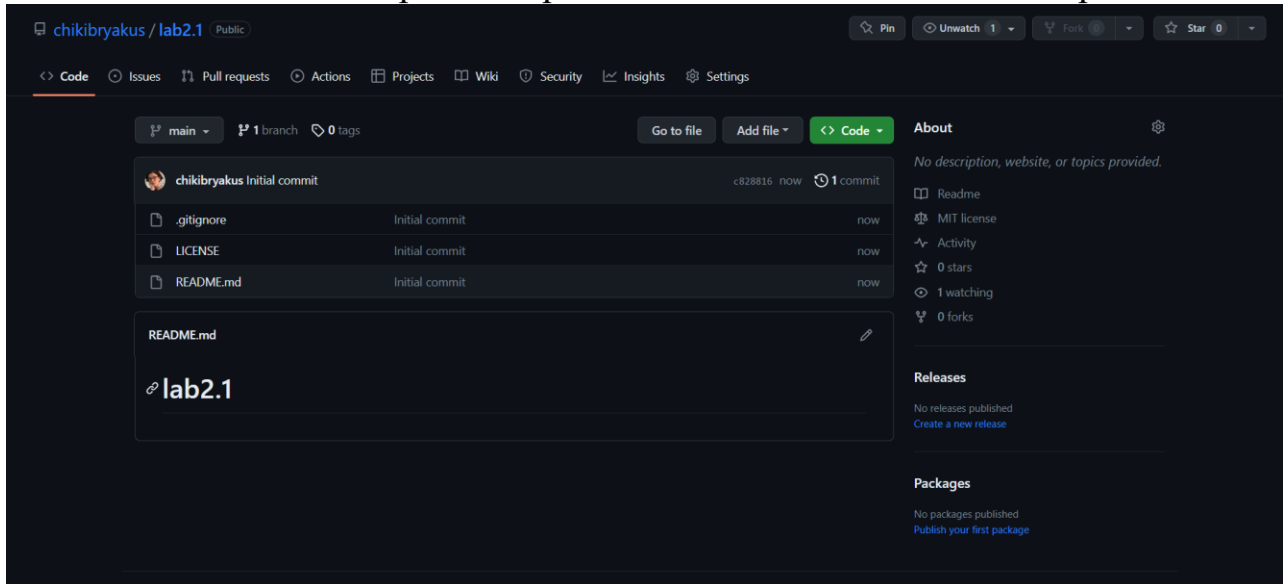


Рисунок 1. Новый репозиторий

2. Проклонировала свой репозиторий на компьютер.

Использовала git-flow.

```
C:\Users\Hp>git clone https://github.com/chikibryakus/lab2.1.git
Cloning into 'lab2.1'...
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\Hp>
```

Рисунок 2. Клон репозитория

```
C:\Users\Hp>cd C:\Users\Hp\lab2.1
C:\Users\Hp\lab2.1>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/Hp/lab2.1/.git/hooks]

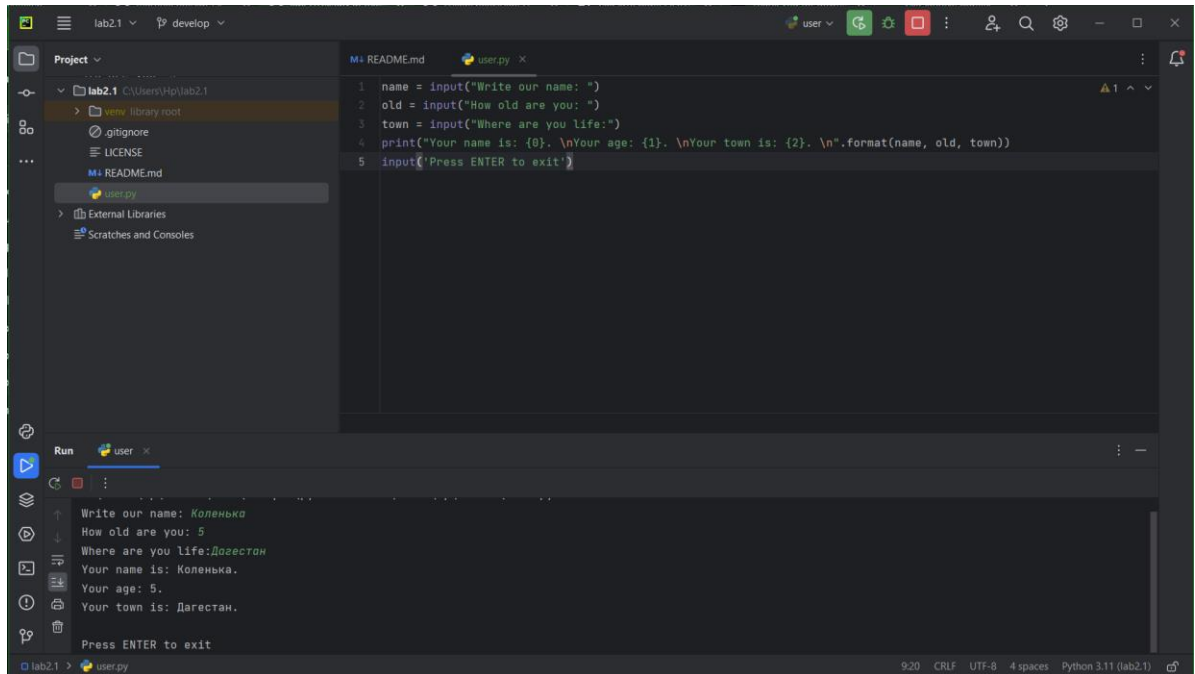
C:\Users\Hp\lab2.1>
C:\Users\Hp\lab2.1>
C:\Users\Hp\lab2.1>
C:\Users\Hp\lab2.1>git branch
* develop
main
```

Рисунок 3. Модель ветвления git-flow

В дальнейшей работе работала на новой ветке develop и в конце лабораторной работы слила ее с main.

3. Написала программу и проверила на работоспособность в файле *user.py*.

Необходимо, чтобы данная программа запрашивала у пользователя его имя, возраст, где он живет и выводила эти данные, написанные пользователем на экран.



The screenshot shows a code editor with a dark theme. The left sidebar displays a project structure for 'lab2.1' with files like 'README.md' and 'user.py'. The main editor window shows the code in 'user.py':

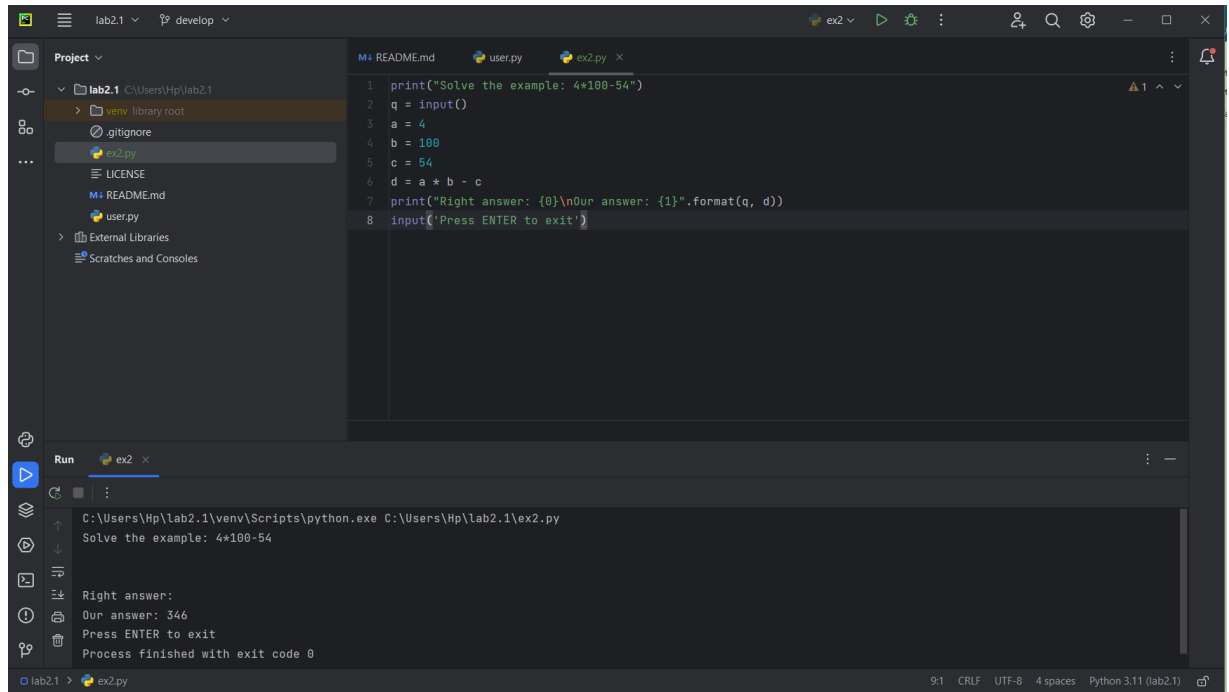
```
1 name = input("Write our name: ")
2 old = input("How old are you: ")
3 town = input("Where are you life:")
4 print("Your name is: {0}. \nYour age: {1}. \nYour town is: {2}. \n".format(name, old, town))
5 input('Press ENTER to exit')
```

Below the editor, the 'Run' console shows the program's execution with user input in Russian: 'Коленька' for name, '5' for age, and 'Дагестан' for town. The output matches the format string in the code.

Рисунок 4. Программа и ее результат

4. Создала новый файл *ex2.py* и написала программу в develop ветке.

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



The screenshot shows a code editor with a project named 'lab2.1'. The file explorer on the left shows the project structure, including 'ex2.py'. The main editor window displays the following Python code:

```
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(q, d))
8 input("Press ENTER to exit")
```

The 'Run' panel at the bottom shows the execution output:

```
C:\Users\Hp\lab2.1\venv\Scripts\python.exe C:\Users\Hp\lab2.1\ex2.py
Solve the example: 4*100-54
Right answer:
Our answer: 346
Press ENTER to exit
Process finished with exit code 0
```

Рисунок 5. Программа и ее результат

5. Создала новый файл *ex3.py* Написала программу и проверила на работоспособность в файле *ex3.py*.

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).

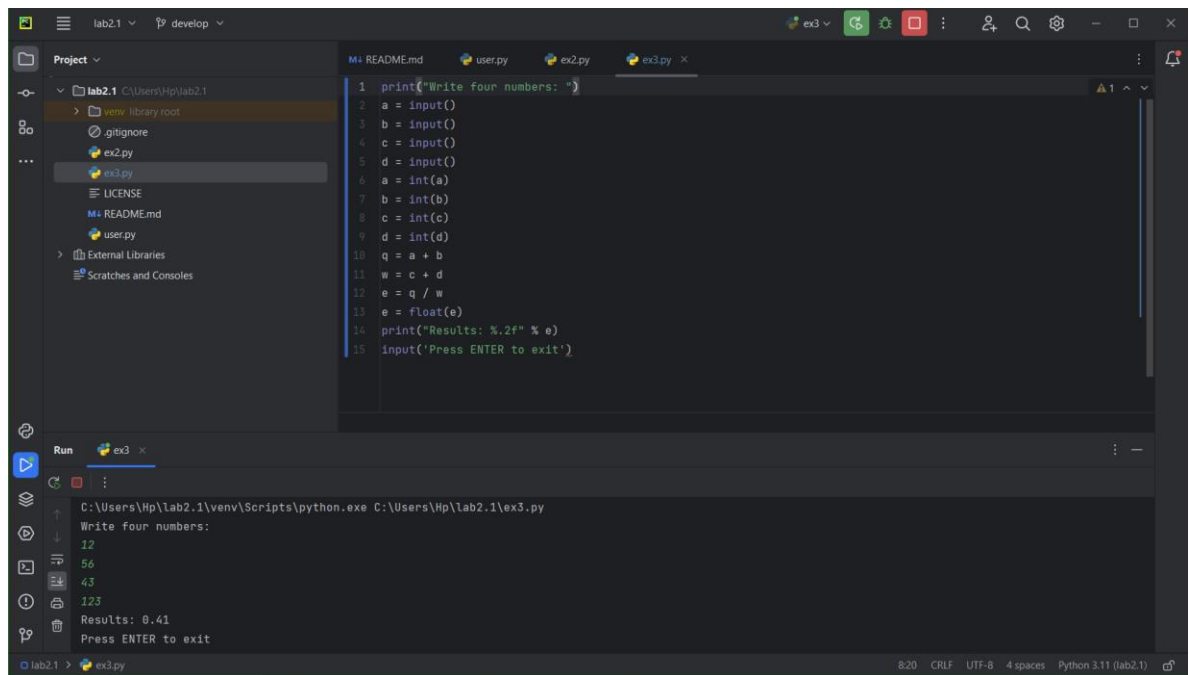


Рисунок 6. Программа и ее результат

6. Сделала индивидуальное в соответствии со списком задание в ветке develop.

8. Даны основания равнобедренной трапеции и угол при большем основании. Найти площадь трапеции.

Рисунок 7. Условие задания

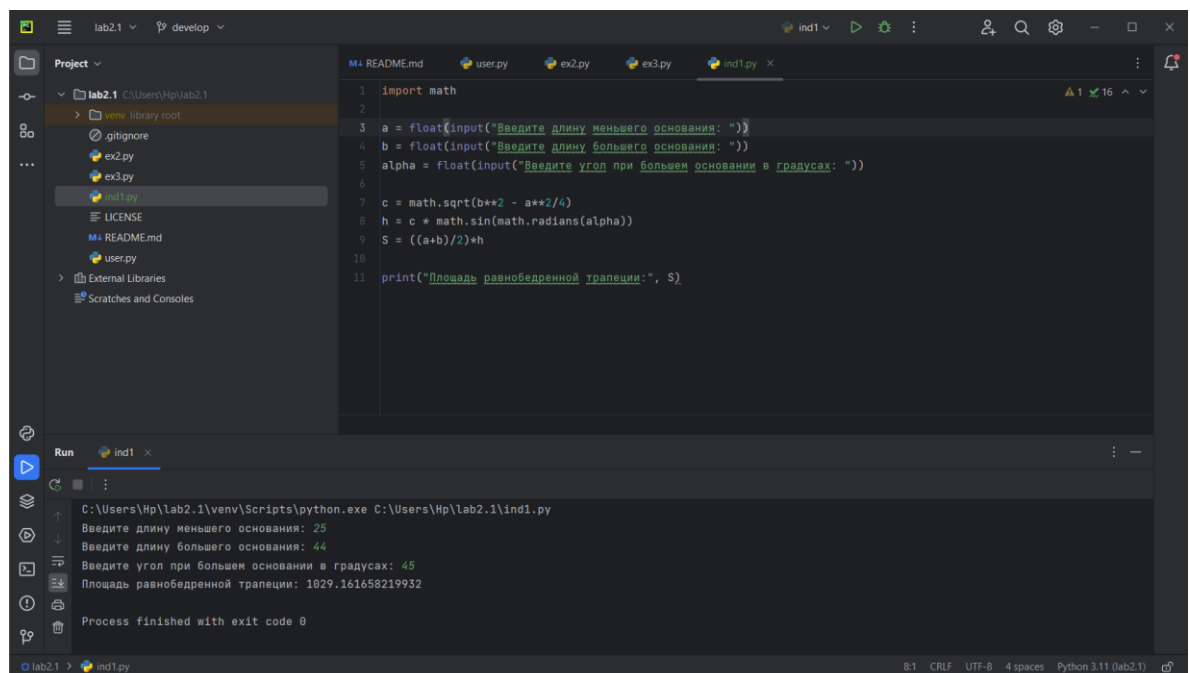


Рисунок 8. Программа и ее результат

7. Слила ветку develop с веткой main.

```
C:\Users\Hp\lab2.1>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

C:\Users\Hp\lab2.1>git merge develop
Updating c828816..d11dc01
Fast-forward
 ex2.py | 8 ++++++++
 ex3.py | 15 ++++++++
 ind1.py | 11 ++++++++
 user.py | 5 +++++
4 files changed, 39 insertions(+)
create mode 100644 ex2.py
create mode 100644 ex3.py
create mode 100644 ind1.py
create mode 100644 user.py
```

Рисунок 9. Слияние веток

8. Отправила все на удаленный репозиторий.

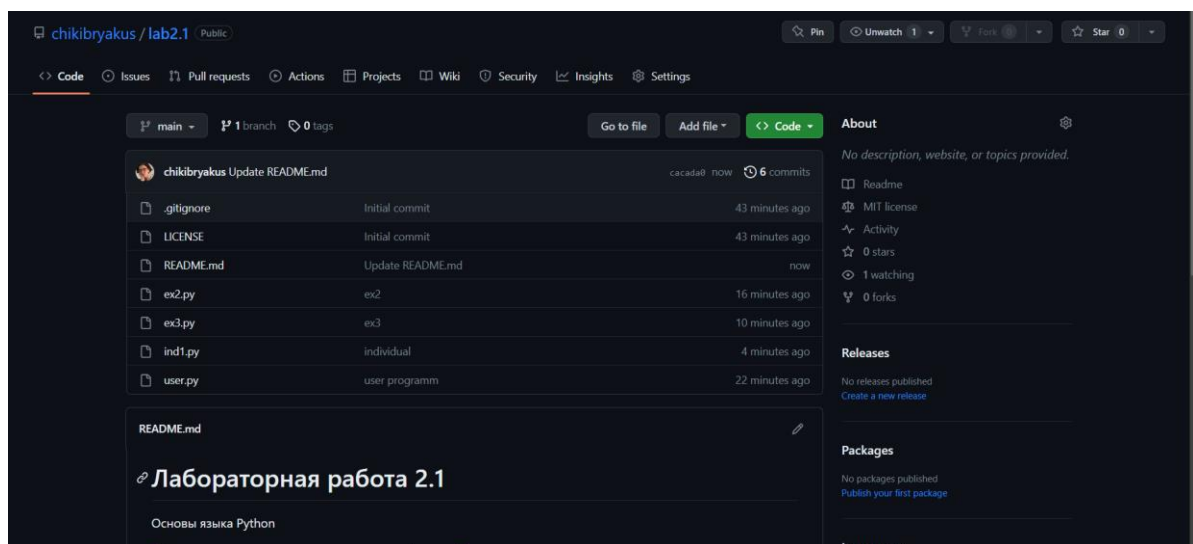


Рисунок 10. Удаленный репозиторий

Ссылка: <https://github.com/chikibryakus/lab2.1>

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

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: я исследовал процесс установки и базовые возможности языка Python3.