

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное автономное
образовательное учреждение 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

Потеев Антон Сергеевич

«___» _____ 2023г.

Подпись студента _____

Проверил: Доцент, к.т.н, доцент
кафедры инфокоммуникаций

Воронкин Роман Александрович

Работа защищена с оценкой:

(подпись)

Ставрополь, 2023 г.

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

Порядок выполнения работы:

1. Изучил теоретический материал
2. Создал общедоступный репозиторий на GitHub

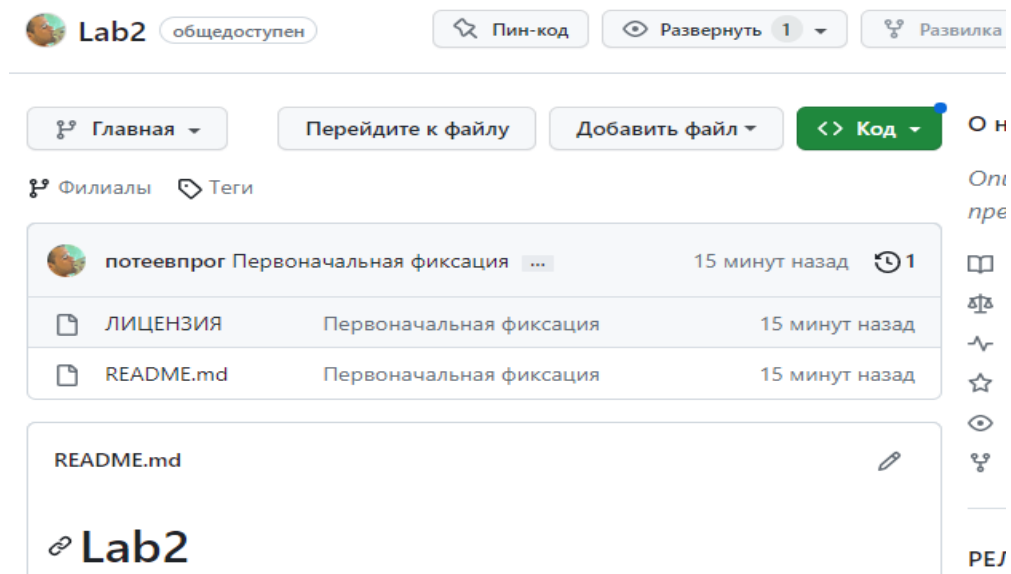


Рисунок 1 - Созданный репозиторий

3. Выполнил клонирование созданного репозитория.
4. Дополнил файл .gitignore и добавил в репозиторий.

```

MINGW64:/c:/Users/user/Desktop/учеба/1 курс/2 семестр/Основы кроссплатформенного программирования
user@WIN-I1P5V403487 MINGW64 ~/Desktop/учеба/1 курс/2 семестр/Основы кроссплатформенного программирования/Lab2 (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  gitignore.txt

nothing added to commit but untracked files present (use "git add" to track)

user@WIN-I1P5V403487 MINGW64 ~/Desktop/учеба/1 курс/2 семестр/Основы кроссплатформенного программирования/Lab2 (main)
$ git add .
warning: in the working copy of 'gitignore.txt', LF will be replaced by CRLF the next time Git touches it

user@WIN-I1P5V403487 MINGW64 ~/Desktop/учеба/1 курс/2 семестр/Основы кроссплатформенного программирования/Lab2 (main)
$ git commit -m "gitignore"
[main c4394f4] gitignore
1 file changed, 185 insertions(+)
create mode 100644 gitignore.txt

user@WIN-I1P5V403487 MINGW64 ~/Desktop/учеба/1 курс/2 семестр/Основы кроссплатформенного программирования/Lab2 (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1.74 KiB | 1.74 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/poteevprog/Lab2
   bd8fd2a..c4394f4  main -> main

user@WIN-I1P5V403487 MINGW64 ~/Desktop/учеба/1 курс/2 семестр/Основы кроссплатформенного программирования/Lab2 (main)
$ git push
Everything up-to-date

user@WIN-I1P5V403487 MINGW64 ~/Desktop/учеба/1 курс/2 семестр/Основы кроссплатформенного программирования/Lab2 (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

user@WIN-I1P5V403487 MINGW64 ~/Desktop/учеба/1 курс/2 семестр/Основы кроссплатформенного программирования/Lab2 (main)
$

```

Рисунок 2 - Клонирование и .gitignore.

5. Написать программу (файл user.py), которая запрашивала бы у пользователя: его имя, возраст, место жительства. После этого вывести три строки: «This is 'имя'», «It is 'возраст'», «(S)he live in 'город'».

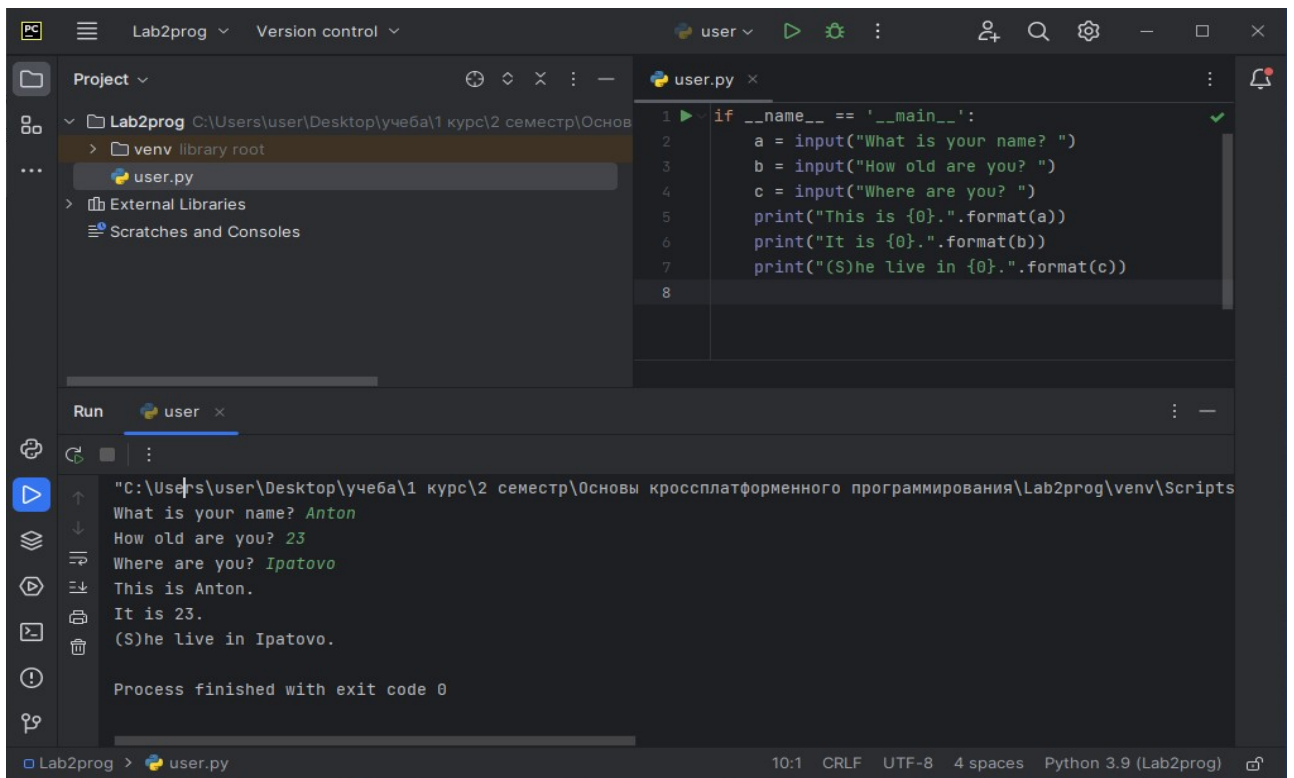


Рисунок 3 – Код и проверка первой задачи

6. Написать программу (файл arithmetic.py), которая предлагала бы пользователю решить пример $4 * 100 - 54$. Потом выводила бы на экран правильный ответ и ответ пользователя.

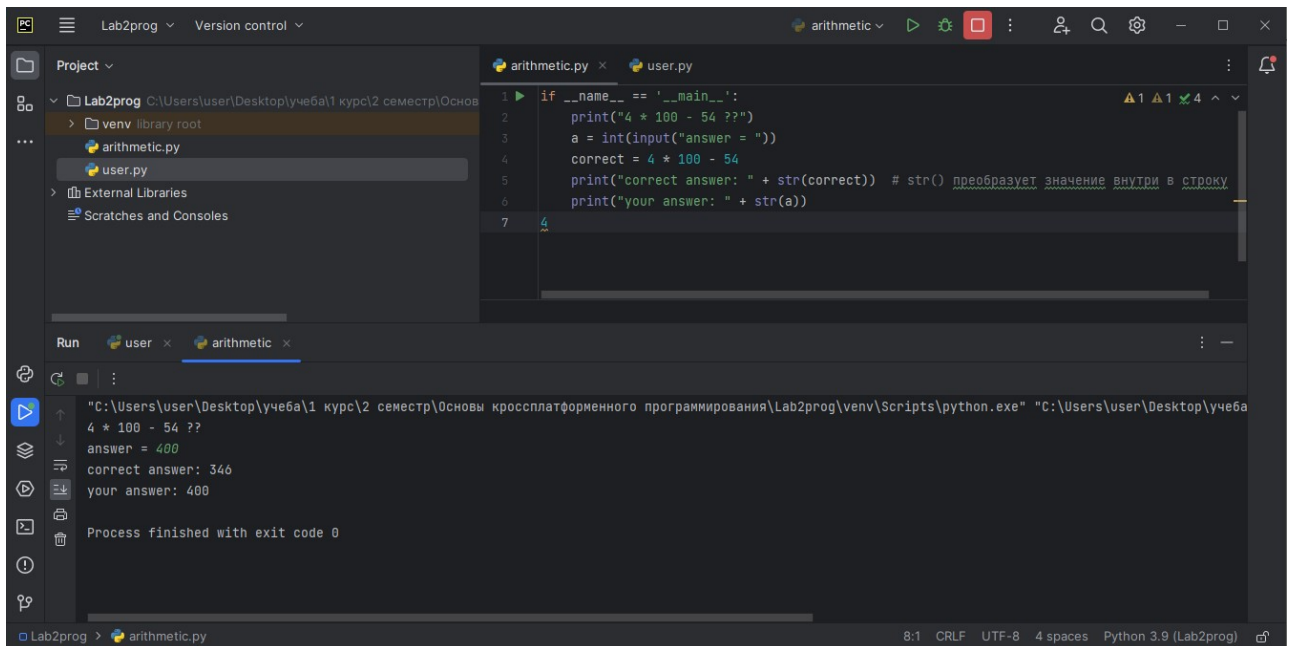
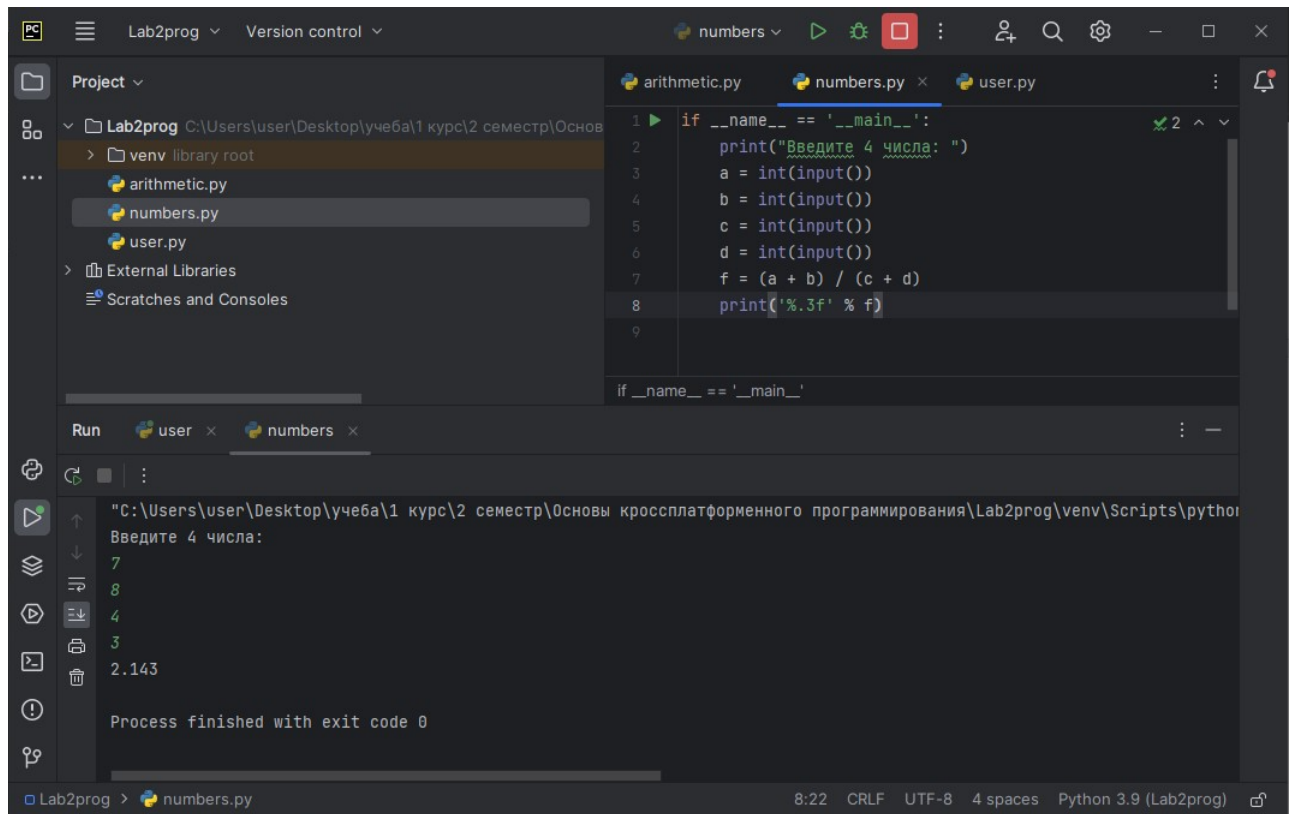


Рисунок 4 – Код и проверка второй задачи

7. Запросить у пользователя четыре числа (файл numbers.py). Отдельно сложить первые два и отдельно вторые два. Разделить первую сумму на вторую. Вывести результат на экран так, чтобы ответ содержал две цифры после запятой.



The screenshot shows an IDE window with a project named 'Lab2prog'. The file explorer on the left shows the project structure, including a 'venv' directory and files 'arithmetic.py', 'numbers.py', and 'user.py'. The main editor displays the code in 'numbers.py':

```
1 if __name__ == '__main__':
2     print("Введите 4 числа: ")
3     a = int(input())
4     b = int(input())
5     c = int(input())
6     d = int(input())
7     f = (a + b) / (c + d)
8     print('% .3f' % f)
9
```

Below the editor, the 'Run' tab shows the execution output:

```
"C:\Users\user\Desktop\учеба\1 курс\2 семестр\Основы кроссплатформенного программирования\Lab2prog\venv\Scripts\python
Введите 4 числа:
7
8
4
3
2.143
Process finished with exit code 0
```

The status bar at the bottom indicates the file is 'numbers.py' in the 'Lab2prog' project, using Python 3.9 with CRLF line endings, UTF-8 encoding, and 4 spaces for indentation.

Рисунок 5 - Код и проверка третьей задачи

8. Написать программу (файл individual.py) для решения индивидуального задания. Вариант индивидуального задания 12. Известна стоимость монитора, системного блока, клавиатуры и мыши. Сколько будут стоить 3 компьютера из этих элементов? N компьютеров?

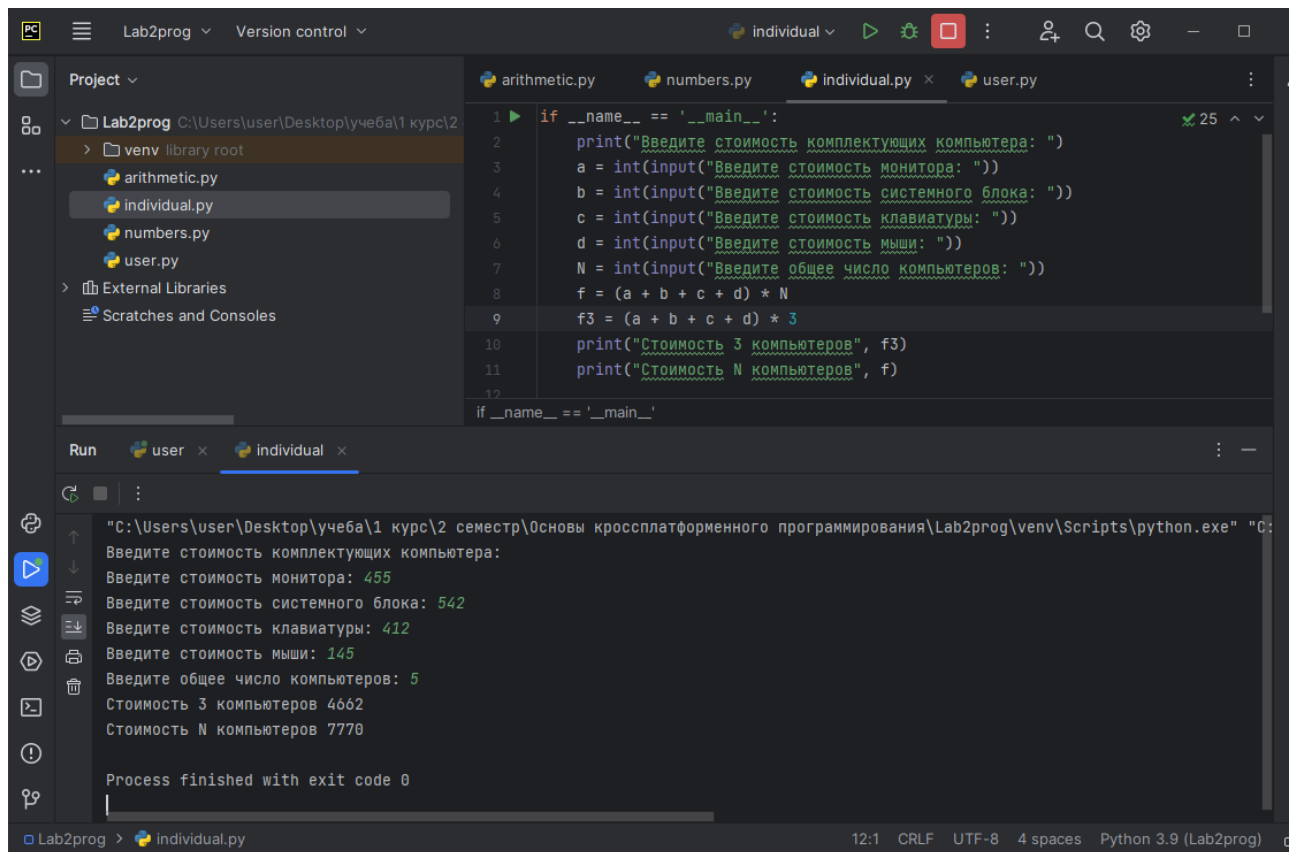


Рисунок 6 - Код и проверка индивидуальной задачи

9. Выполнил коммит всех разработанных файлов

Совершается 21 июня 2023 года

индивидуальный	ca771a4	<>
потеевпрог зафиксирован 3 минуты назад		
числа	16c56e7	<>
потеевпрог зафиксирован 22 минуты назад		

Совершается 20 июня 2023 года

прочитай меня	20f08cc	<>
потеевпрог зафиксирован 35 минут назад		
арифметика.py	b1a09df	<>
потеевпрог зафиксирован 40 минут назад		
арифметика.py	cc8902b	<>
потеевпрог зафиксирован 41 минуту назад		
пользователь.py	4a7fc49	<>
потеевпрог зафиксирован 1 час назад		
gitignore	c4394f4	<>
потеевпрог зафиксирован 1 час назад		
Первоначальная фиксация	Проверено bd8fd2a	<>
потеевпрог зафиксирован 1 час назад		

Рисунок 7 – Коммит файлов в репозиторий.

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 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?

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 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?

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.