

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования «СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Институт цифрового развития

ОТЧЁТ

по лабораторной работе №2.1

Дисциплина: «Основы кроссплатформенного программирования»

Тема: «Основы языка Python»

Выполнила: студентка 1 курса,
группы ИВТ-б-о-21-1
Диченко Дина Алексеевна

Ставрополь 2022

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

Выполнение работы:

1. Создала общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и язык программирования Python.

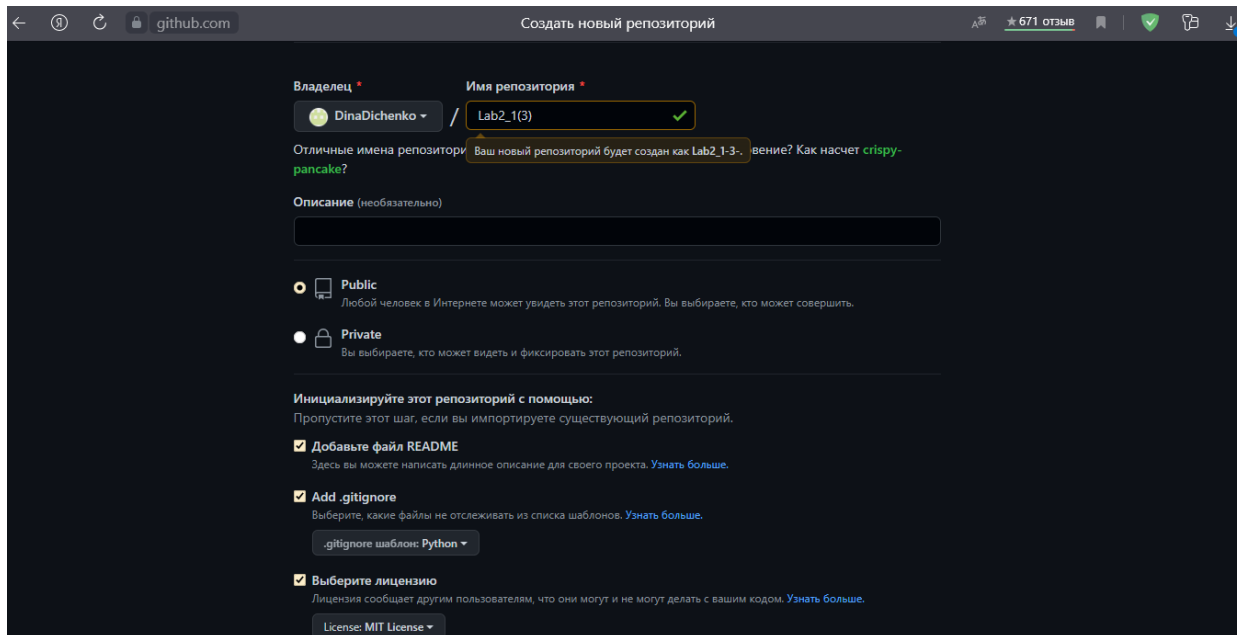


Рисунок 1. Создание репозитория

2. Клонировала репозиторий.

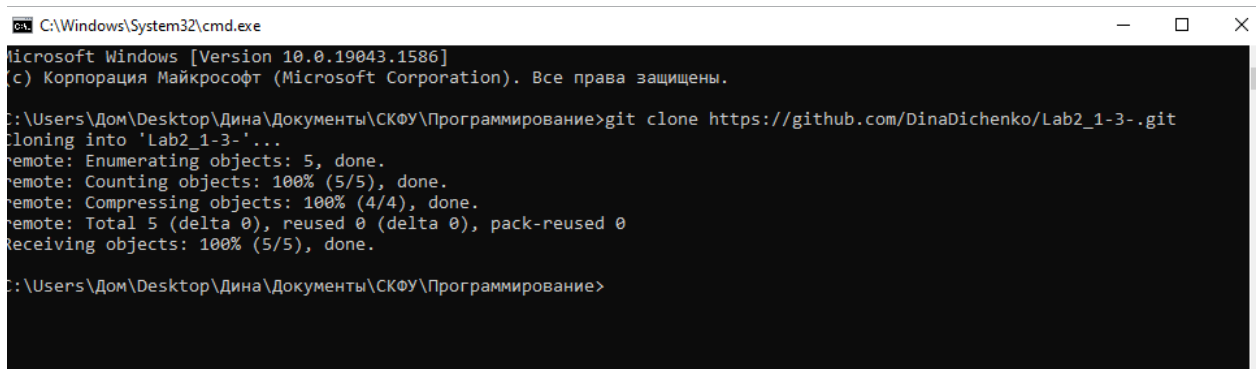


Рисунок 2. Клонирование репозитория

```

C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование>git flow init
Initialized empty Git repository in C:/Users/Дом/Desktop/Дина/Документы/СКФУ/Программирование/.git/
No branches exist yet. Base branches must be created now.
Branch name for production releases: [master]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]

```

Рисунок 3. Организация репозитория в соответствии с моделью ветвления git-flow

3. Написала программу (файл user.py), которая запрашивала бы у пользователя имя, возраст и место жительства.

The screenshot shows an IDE with a project named 'prog' containing files 'arithmetic.py', 'main.py', and 'user.py'. The 'user.py' file is open, showing the following code:

```

1 n = input("Write your name: ")
2
3 a = input("Write your age: ")
4
5 t = input("Write your town: ")
6
7 print("This is ", n)
8 print("It is ", a)
9 print("(S)he live in ", t)

```

Below the code editor, the 'Run' window shows the execution of 'main.py' (which calls 'user.py'). The output is as follows:

```

D:\python\python.exe C:/Users/дом/desktop/дина/документы/скфу/программирование/Lab2_1-3-/prog/main.py
Write your name: Dina
Write your age: 18
Write your town: Stavropol
This is Dina
It is 18
(S)he live in Stavropol
Process finished with exit code 0

```

Рисунок 4. Программа user

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

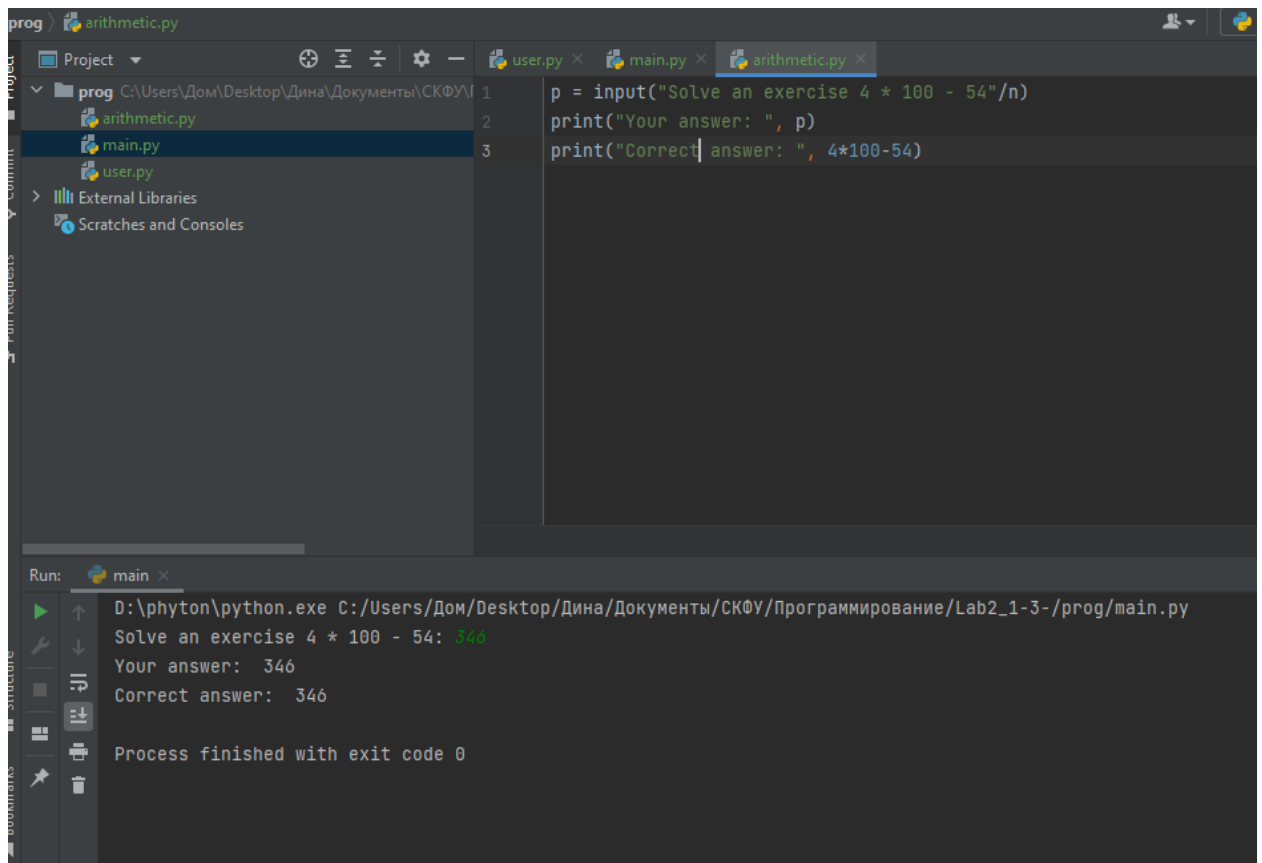


Рисунок 5. Программа arithmetic

5. Написала программу numbers.py (запросите у пользователя четыре числа. 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)

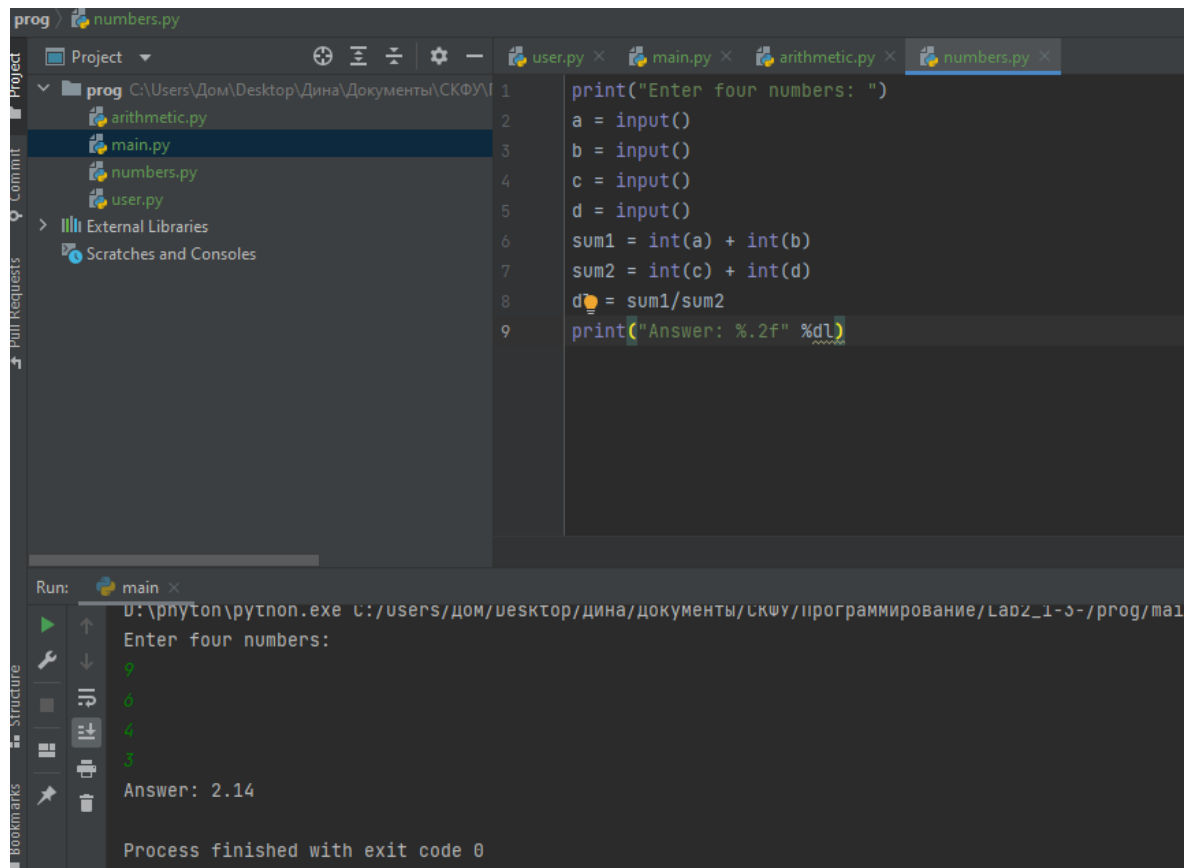


Рисунок 6. Программа numbers

6. Написала программу (файл individual.py) для решения индивидуального задания.

Вариант 7.

Даны основания и высота равнобедренной трапеции. Найти периметр трапеции.

```
1 #Даны основания и высота равнобедренной трапеции. Найти периметр трапеции.
2 import math
3
4 h = float(input("Enter the height of the trapezoid: "))
5 a = float(input("Enter a larger trapezoid base: "))
6 b = float(input("Enter a smaller trapezoid base: "))
7 c = 2*math.sqrt(pow(h,2) + pow_(((a-b)/2),2))+a+b
8 print("The perimeter of the trapezoid: ", c)
```

```
D:\phyton\python.exe C:/Users/Дом/Desktop/Дина/Документы/СКФУ/Программирование/Lab2_1-3-/prog/individual.py
Enter the height of the trapezoid: 4
Enter a larger trapezoid base: 12
Enter a smaller trapezoid base: 4
The perimeter of the trapezoid: 28.0
Process finished with exit code 0
```

Рисунок 7. Программа individual

Задача повышенной сложности:

Часовая стрелка образует угол y с лучом, проходящим через центр и через точку, соответствующую 12 часам на циферблате. Определить значение угла для минутной стрелки, а также количество полных часов и полных минут.

```
1 #Часовая стрелка образует угол y с лучом, проходящим через центр и через точку,
2 #соответствующую 12 часам на циферблате. Определить значение угла для
3 #минутной стрелки, а также количество полных часов и полных минут.
4 y = int(input("Enter the angle that the beam and the hour hand form: "))
5 ch = y//30
6 min = y%30
7 print("Minute hand angle: ", min*12)
8 print("TIME: ", ch, "hours", min, "minutes")
9
```

```
D:\phyton\python.exe C:/Users/Дом/Desktop/Дина/Документы/СКФУ/Программирование/Lab2_1-3-/prog/difficult.py
Enter the angle that the beam and the hour hand form: 45
Minute hand angle: 180
TIME: 1 hours 15 minutes
Process finished with exit code 0
```

Рисунок 8. Программа задачи повышенной сложности

```

C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab2.1>git commit -m "add"
develop f69f6a7] add
11 files changed, 69 insertions(+)
create mode 100644 prog/.idea/.gitignore
create mode 100644 prog/.idea/inspectionProfiles/profiles_settings.xml
create mode 100644 prog/.idea/misc.xml
create mode 100644 prog/.idea/modules.xml
create mode 100644 prog/.idea/prog.iml
create mode 100644 prog/.idea/vcs.xml
create mode 100644 prog/arithmetic.py
create mode 100644 prog/difficult.py
create mode 100644 prog/individual.py
create mode 100644 prog/numbers.py
create mode 100644 prog/user.py

C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab2.1>git branch
* develop
  main

C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab2.1>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

```

Рисунок 9. Коммит изменений в ветку develop

```

C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab2.1>git merge develop
Updating 85b20cb..f69f6a7
Fast-forward
 prog/.idea/.gitignore          | 3 +++
 prog/.idea/inspectionProfiles/profiles_settings.xml | 6 ++++++
 prog/.idea/misc.xml           | 4 ++++
 prog/.idea/modules.xml        | 8 ++++++++
 prog/.idea/prog.iml           | 8 ++++++++
 prog/.idea/vcs.xml            | 6 ++++++
 prog/arithmetic.py            | 3 +++
 prog/difficult.py             | 8 ++++++++
 prog/individual.py            | 8 ++++++++
 prog/numbers.py               | 9 ++++++++
 prog/user.py                  | 6 ++++++
11 files changed, 69 insertions(+)
create mode 100644 prog/.idea/.gitignore
create mode 100644 prog/.idea/inspectionProfiles/profiles_settings.xml
create mode 100644 prog/.idea/misc.xml
create mode 100644 prog/.idea/modules.xml
create mode 100644 prog/.idea/prog.iml
create mode 100644 prog/.idea/vcs.xml
create mode 100644 prog/arithmetic.py
create mode 100644 prog/difficult.py
create mode 100644 prog/individual.py
create mode 100644 prog/numbers.py
create mode 100644 prog/user.py

```

Рисунок 10. Слияние ветки develop с веткой main

```
C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab2.1>git push
Enumerating objects: 17, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 2 threads
Compressing objects: 100% (14/14), done.
Writing objects: 100% (16/16), 2.47 KiB | 158.00 KiB/s, done.
Total 16 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/DinaDichenko/Lab2.1.git
85b20cb..f69f6a7 main -> main
```

Рисунок 11. Пуш коммитов на удаленный репозиторий

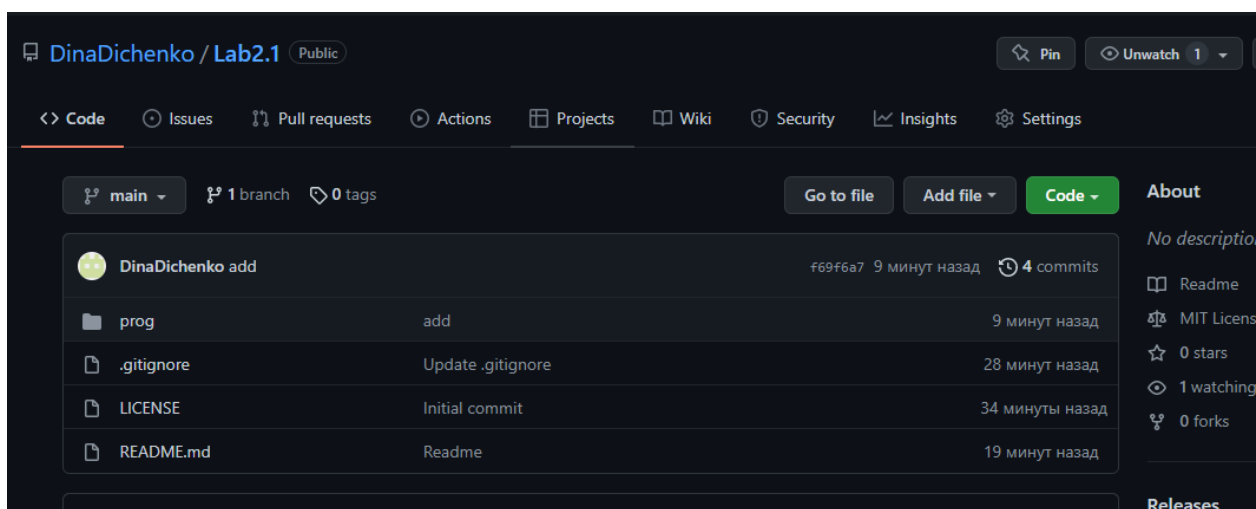


Рисунок 12. Изменения на удаленном репозитории

Ответы на вопросы:

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

Порядок установки Python в Windows:

- 1) запустите скачанный установочный файл;
- 2) выберите способ установки;
- 3) отметьте необходимые опции установки (доступно при выборе Customize installation);
- 4) выберите место установки (доступно при выборе Customize installation);
- 5) после успешной установки вас ждет следующее сообщение.

Порядок установки Python в Linux: чаще всего интерпретатор Python уже входит в состав дистрибутива.

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

Для удобства запуска примеров и изучения языка Python, настоятельно рекомендуется установить на свой ПК пакет Anaconda. Этот пакет включает в себя интерпретатор языка Python (есть версии 2 и 3), набор наиболее часто используемых библиотек и удобную среду разработки и исполнения, запускаемую в браузере.

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 после чего запустится веб-сервер и среда разработки в браузере.

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

Указать путь до интерпретатора в настройках IDE.

5. Как осуществить запуск программы с помощью IDE PyCharm?

Сочетанием клавиш Shift+F10.

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

Интерактивный: Python можно использовать как калькулятор для различных вычислений, а если дополнительно подключить необходимые математические библиотеки, то по своим возможностям он становится практически равным таким пакетам как Matlab, Octave и т.п.

Пакетный: сначала записывается вся программа, потом она выполняется полностью.

7. Почему язык программирования Python называется языком динамической типизации?

В языке программирования Python 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);
- 4) Списки (Sequence Type);
- 5) строки (Text Sequence Type);
- 6) бинарные списки (Binary Sequence Types);
- 7) множества (Set Types);
- 8) словари (Mapping Types).

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.

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?

При целочисленном делении дробная часть отбрасывается, при обычном – нет.

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 $a + bj$.

Комплексные числа можно складывать, вычитать, умножать, делить и возводить в степень.

У комплексного числа можно извлечь действительную и мнимую части (`x.real` и `x.imag`).

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.

Основные функции:

`math.ceil(x)` - возвращает ближайшее целое число большее, чем x .

`math.fabs(x)` - возвращает абсолютное значение числа.

`math.factorial(x)` - вычисляет факториал x .

`math.floor(x)` - возвращает ближайшее целое число меньшее, чем x .

`math.exp(x)` - вычисляет $e^{**}x$.

`math.log2(x)` - логарифм по основанию 2.

`math.log10(x)` - логарифм по основанию 10.

`math.log(x[, base])` - по умолчанию вычисляет логарифм по основанию e , дополнительно можно указать основание логарифма.

`math.pow(x, y)` - вычисляет значение x в степени y .

`math.sqrt(x)` - корень квадратный от x .

`math.cos(x)` - косинус от x .

`math.sin(x)` - синус от x .

`math.tan(x)` - тангенс от x .

`math.acos(x)` - арккосинус от x .

`math.asin(x)` - арксинус от x .

`math.atan(x)` - арктангенс от x .

`math.pi` - число π .

`math.e` - число e .

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

Через параметр `sep` можно указать отличный от пробела разделитель строк.

Параметр `end` позволяет указывать, что делать, после вывода строки. По умолчанию происходит переход на новую строку. Однако это действие можно отменить, указав любой другой символ или строку.

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.

Форматирование может выполняться в так называемом старом стиле или с помощью строкового метода `format`. Старый стиль также называют Си-стилем, так как он схож с тем, как происходит вывод на экран в языке C.

Буквы `s`, `d`, `f` обозначают типы данных – строку, целое число, вещественное число. Если бы требовалось подставить три строки, то во всех случаях использовалось бы сочетание `%s`.

К строке применяется метод `format()`. В его скобках указываются сами данные (можно использовать переменные). На нулевое место подставится первый аргумент метода `format()`, на место с номером 1 – второй и т. д.

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

Указать перед `input()` тип данных: `int(input())`.

Вывод: в результате выполнения работы был исследован процесс установки и базовые возможности языка Python версии 3.x.