

МИНИСТЕРСТВО НАУКИ ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
Институт цифрового развития  
Кафедра инфокоммуникаций.

**ОТЧЁТ**  
**по лабораторной работе №4 (2.1)**  
Дисциплина: «Программирование на Python»  
Тема: «Основы языка Python»

Выполнил:  
студент 2 курса группы ИВТ-б-о-22-1

Бабенко Артём Тимофеевич

Проверил:  
Доцент кафедры инфокоммуникаций

Воронкин Р.А

Работа защищена с оценкой: \_\_\_\_\_

Ставрополь, 2023

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

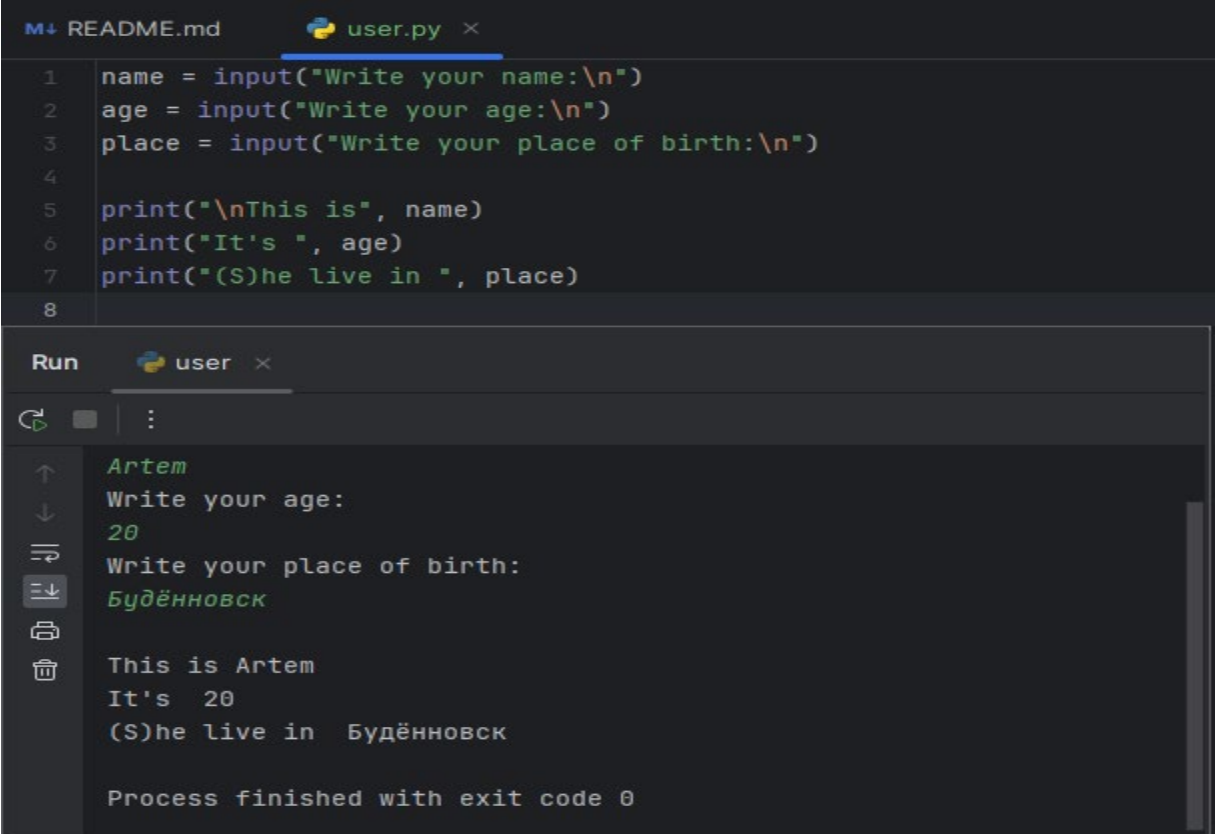
**Ход работы:**

```
>>> print("Hello, world!")  
Hello, world!  
>>>
```

Рисунок 1. Проверка установки программ.

```
D:\Gitlab\p\lab4>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? [D:/Gitlab/p/lab4/.git/hooks]
```

Рисунок 2. Организация репозитория согласно модели ветвления git-flow



The screenshot shows a code editor with two tabs: 'README.md' and 'user.py'. The 'user.py' tab is active, displaying a Python script that prompts the user for their name, age, and place of birth, and then prints the information. Below the code editor is a 'Run' window showing the execution of the script. The output shows the user's input: 'Artem', '20', and 'Будёновск', followed by the program's output: 'This is Artem', 'It's 20', and '(S)he live in Будёновск'. The process finished with exit code 0.

```
1 name = input("Write your name:\n")  
2 age = input("Write your age:\n")  
3 place = input("Write your place of birth:\n")  
4  
5 print("\nThis is", name)  
6 print("It's ", age)  
7 print("(S)he live in ", place)  
8  
  
Run user  
  
Artem  
Write your age:  
20  
Write your place of birth:  
Будёновск  
  
This is Artem  
It's 20  
(S)he live in Будёновск  
  
Process finished with exit code 0
```

Рисунок 3. Программа user

```
1 ans = input("Solve the example\n\t4*100-54=")
2 print("Your answer: ", ans)
3 print("Right answer: ", 4 * 100 - 54)
4
```

Run arithmetic x

D:\Gitlab\P\lab4\.venv\Scripts\python.exe D:\Gitlab\P\lab4\arithmetic

Solve the example  
4\*100-54=5  
Your answer: 5  
Right answer: 346

Process finished with exit code 0

Рисунок 4. Программа arithmetic.py

```
1 print("Введите любые 4 числа")
2 num1 = int(input())
3 num2 = int(input())
4 num3 = int(input())
5 num4 = int(input())
6 sum1 = num1 + num2
7 sum2 = num3 + num4
8 rez = sum1 / sum2
9 print(format(rez, ".2"))
10
```

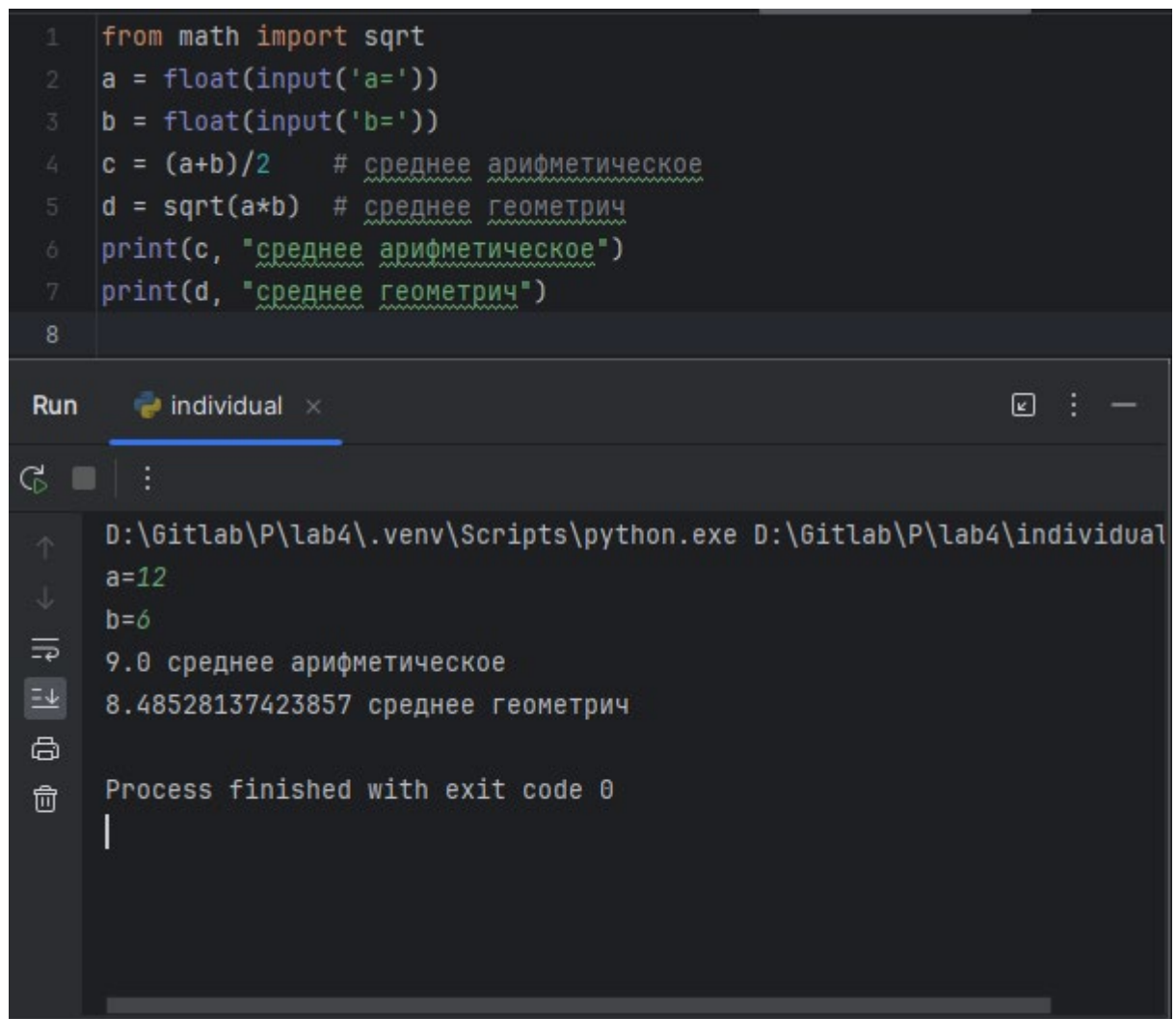
Run numbers x

D:\Gitlab\P\lab4\.venv\Scripts\python.exe D:\Gitlab\P\lab4\numbers.py

Введите любые 4 числа  
41  
1  
2  
5  
6.0

Process finished with exit code 0

Рисунок 5. Программа numbers.py



```
1 from math import sqrt
2 a = float(input('a='))
3 b = float(input('b='))
4 c = (a+b)/2 # среднее арифметическое
5 d = sqrt(a*b) # среднее геометрич
6 print(c, "среднее арифметическое")
7 print(d, "среднее геометрич")
8
```

Run individual x

D:\Gitlab\P\lab4\.venv\Scripts\python.exe D:\Gitlab\P\lab4\individual

a=12

b=6

9.0 среднее арифметическое

8.48528137423857 среднее геометрич

Process finished with exit code 0

Рисунок 6. Программа individual.py

**Вывод:** исследование процесса установки и базовых возможностей языка Python версии 3.x. было проведено успешно.

#### Контрольные вопросы:

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

Первое, что нужно сделать – это скачать дистрибутив. Загрузить его можно с официального сайта.

Порядок установки для Windows

- Запустите скачанный установочный файл.
- Выберет способ установки.

- Отметьте необходимые опции установки (доступно при выборе Customize installation)
- Выберите место установки (доступно при выборе Customize installation)

### Порядок установки для Linux

Чаще всего интерпретатор Python уже входит в состав дистрибутива. Это можно проверить набрав в терминале:

“\$ python”

В случае если он не установлен можно воспользоваться командой:

“\$ sudo apt-get install python3”

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

Anaconda 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

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

Запустите PyCharm и выберите Create New Project в появившемся окне.

Укажите путь до проекта Python и интерпретатор, который будет использоваться для запуска и отладки.

5. Как осуществить запуск программы с помощью IDE PyCharm?

Сочетанием клавиш Shift+F10

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

Интерактивный режим – прямое исполнение команд введенных в терминал.

Пакетный режим – выполнения пакетов команд, заранее сохранённых в системе.

7. Почему язык программирования Python называется языком динамической типизации?

Есть языки, в которых интерпретатор производит проверку типов, доступности функций и переменных во время выполнения кода. Такие языки называются динамически типизированными. Если мы случайно ошиблись и вызвали несуществующую функцию, то ошибка всплывет только во время вызова этой функции.

8. Какие существуют основные типы в языке программирования Python?

None (неопределенное значение переменной)

Логические переменные (Boolean Type)

Числа (Numeric Type)

- `int` – целое число
- `float` – число с плавающей точкой
- `complex` – комплексное число

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`

объявляет переменную `b` и присваивает ей значение `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 инициализации переменной, на уровне интерпретатора, происходит следующее: создается целочисленный объект `5` (можно представить, что в этот момент создается ячейка и `5` кладется в эту ячейку); данный объект имеет некоторый идентификатор, значение: `5`, и тип: целое число; посредством оператора “=” создается ссылка между переменной `b` и целочисленным объектом `5` (переменная `b` ссылается на объект `5`).

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`).

13. Чем отличаются операции деления и целочисленного деления?

В целочисленном делении, итоговое число “округляется” в меньшую сторону, до целого значения.

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(`x.real`) и мнимую части(`x.imag`).

Для получения комплексно сопряжённого числа необходимо использовать метод `conjugate()`.

15. Каково назначение и основные функции библиотеки (модуля) `math`? По аналогии с модулем `math` изучите самостоятельно назначение и основные функции модуля `cmath`.

Для выполнения математических операций необходим модуль `math`.

Осн. операции библиотеки `math`:

`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  $\pi$ . `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.

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`. Символы `%s` , `%d` , `%f` подставляются значения переменных. Буквы `s`, `d`, `f` обозначают типы данных – строку, целое число, вещественное число.

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

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