

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное
учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ»

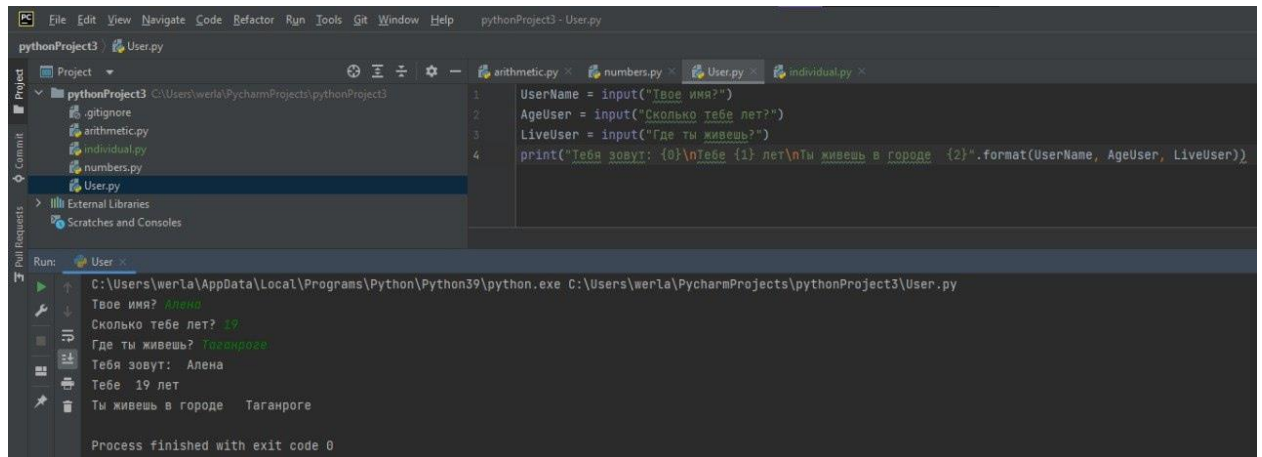
«Исследование основных возможностей Git и GitHub»

ОТЧЕТ
по лабораторной работе
«2.1 Основы языка Python»

Выполнил:
Гладкова Елена
2 курс, группа ППЖ-б-о-21-1.
Проверил:
Воронкин Роман Александрович

Ставрополь, 2022 г.

8. Напишите программу (файл user.py)



```
pythonProject3 - User.py
1  UserName = input("Ваше имя?")
2  AgeUser = input("Сколько тебе лет?")
3  LiveUser = input("Где ты живешь?")
4  print("Тебя зовут: {0}\nТебе {1} лет\nТы живешь в городе {2}".format(UserName, AgeUser, LiveUser))
```

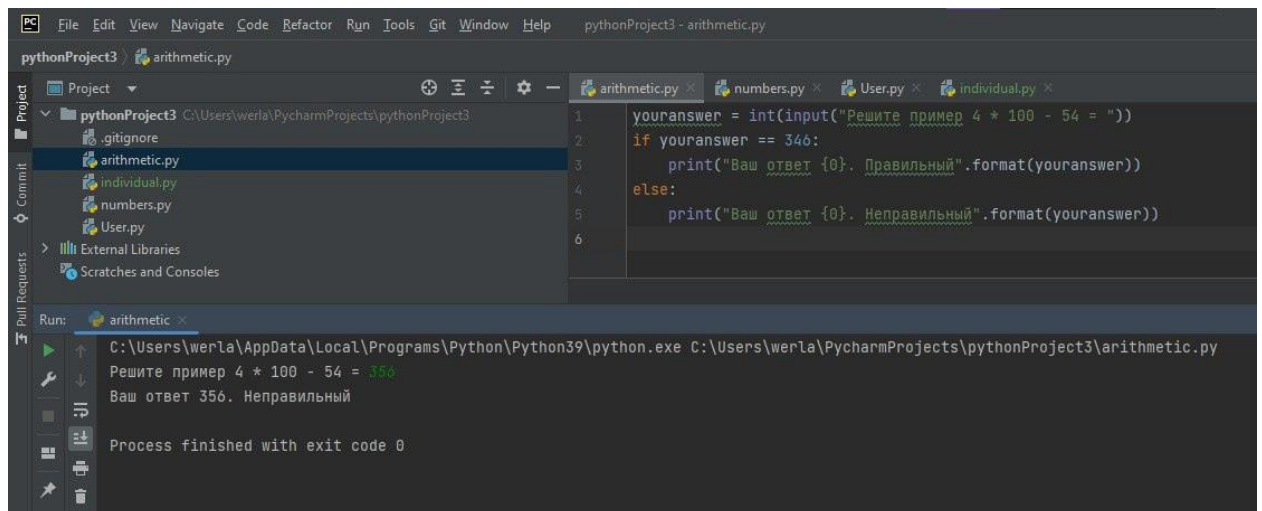
Run: User

C:\Users\werla\AppData\Local\Programs\Python\Python39\python.exe C:\Users\werla\PycharmProjects\pythonProject3\User.py

Ваше имя? Алена
Сколько тебе лет? 19
Где ты живешь? Таганрог
Тебя зовут: Алена
Тебе 19 лет
Ты живешь в городе Таганрог

Process finished with exit code 0

9. Напишите программу (файл arithmetic.py)



```
pythonProject3 - arithmetic.py
1  youranswer = int(input("Решите пример 4 * 100 - 54 = "))
2  if youranswer == 346:
3      print("Ваш ответ {0}. Правильный".format(youranswer))
4  else:
5      print("Ваш ответ {0}. Неправильный".format(youranswer))
6
```

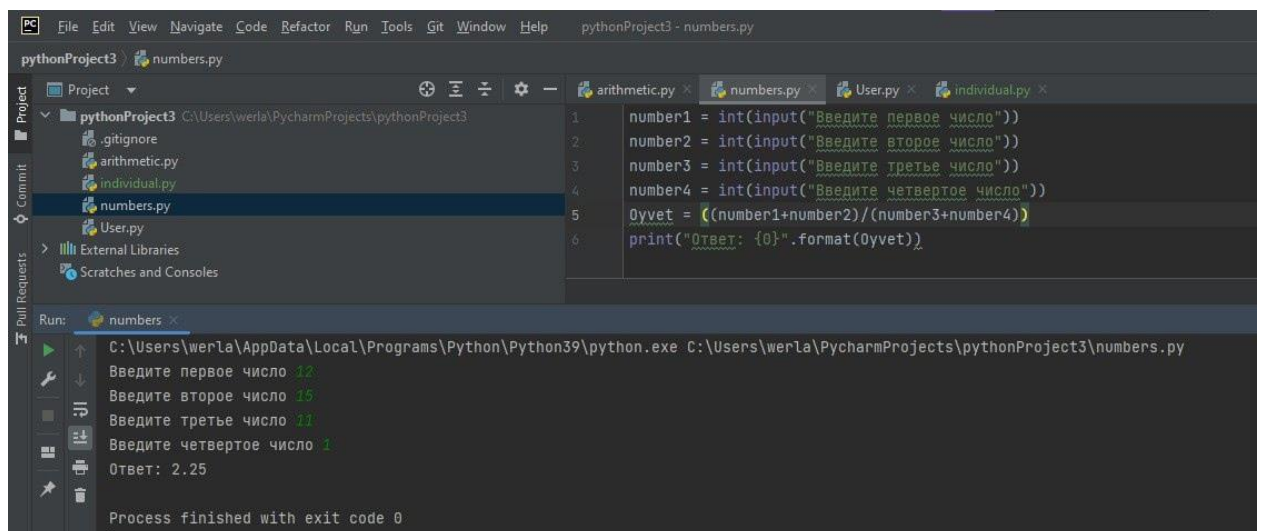
Run: arithmetic

C:\Users\werla\AppData\Local\Programs\Python\Python39\python.exe C:\Users\werla\PycharmProjects\pythonProject3\arithmetic.py

Решите пример 4 * 100 - 54 = 356
Ваш ответ 356. Неправильный

Process finished with exit code 0

10. Запросите у пользователя четыре числа (файл numbers.py)



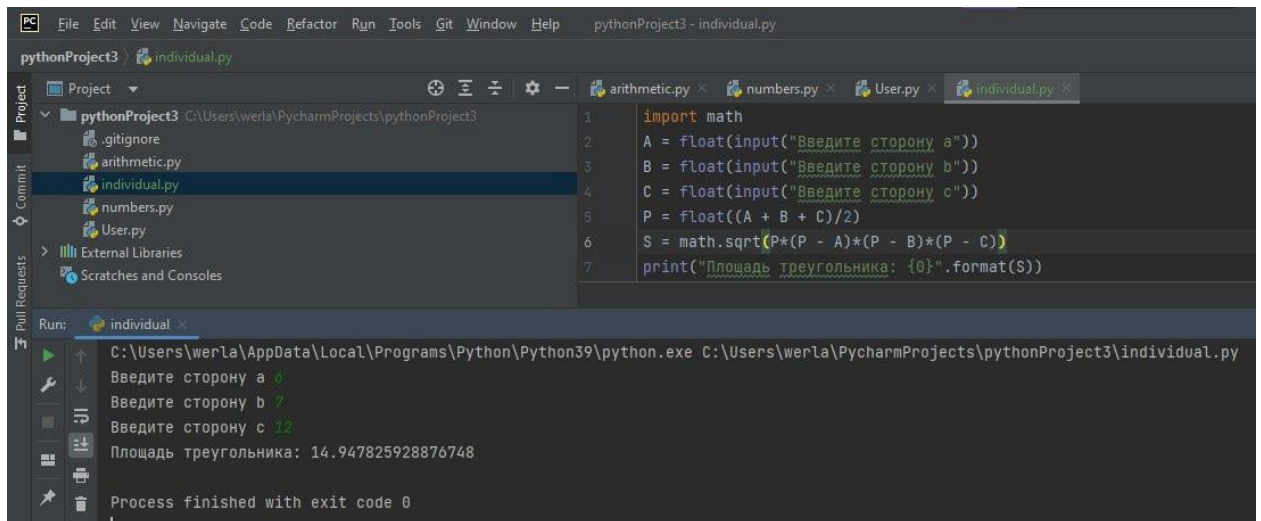
```
pythonProject3 - numbers.py
1  number1 = int(input("Введите первое число"))
2  number2 = int(input("Введите второе число"))
3  number3 = int(input("Введите третье число"))
4  number4 = int(input("Введите четвертое число"))
5  Oyvvet = ((number1+number2)/(number3+number4))
6  print("Ответ: {0}".format(Oyvvet))
```

Run: numbers

C:\Users\werla\AppData\Local\Programs\Python\Python39\python.exe C:\Users\werla\PycharmProjects\pythonProject3\numbers.py

Введите первое число 12
Введите второе число 15
Введите третье число 11
Введите четвертое число 1
Ответ: 2.25

Process finished with exit code 0



Ссылка на репозиторий:

<https://github.com/PuGaLo1337/git>

Вопросы для защиты работы

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

1.1 Установка Python в Windows

Для операционной системы Windows дистрибутив распространяется либо в виде исполняемого файла (с расширением exe), либо в виде архивного файла (с расширением zip). Если вы используете Windows 7, не забудьте установить Service Pack 1!

Порядок установки:

1.1.1. Запустите скачанный установочный файл.

1.1.2. Выберет способ установки.

В данном окне предлагается два варианта Install Now и Customize installation. При выборе Install Now, Python установится в папку по указанному пути. Помимо самого интерпретатора будет установлен IDLE (интегрированная среда разработки), pip (пакетный менеджер) и документация, а также будут созданы соответствующие ярлыки и установлены связи файлов, имеющие расширение .py с интерпретатором Python. Customize installation – это вариант настраиваемой установки. Опция Add python 3.5 to PATH нужна для того, чтобы появилась возможность запускать интерпретатор без указания полного пути до исполняемого файла при работе в командной строке

1.1.3. Отметьте необходимые опции установки (доступно при выборе Customize installation)

На этом шаге нам предлагается отметить дополнения, устанавливаемые вместе с интерпретатором Python. Рекомендуется выбрать все опции. Documentation – установка документаций. pip – установка пакетного менеджера pip. tcl/tk and IDLE – установка интегрированной среды разработки (IDLE) и библиотеки для построения графического интерфейса (tkinter).

1.1.4. Выберите место установки (доступно при выборе Customize installation)

Помимо указания пути, данное окно позволяет внести дополнительные изменения в процесс установки с помощью опций: Install for all users – Установить для всех пользователей. Если не выбрать данную опцию, то будет предложен вариант инсталляции в папку пользователя, устанавливающего интерпретатор. Associate files with Python – Связать файлы, имеющие расширение .py, с Python. При выборе данной опции будут внесены изменения в Windows, позволяющие запускать Python скрипты по двойному щелчку мыши. Create shortcuts for installed applications – Создать ярлыки для запуска приложений. Add Python to environment variables – Добавить пути до интерпретатора Python в переменную PATH. Precompile standard library – Провести прекомпиляцию стандартной библиотеки. Последние два пункта связаны с загрузкой компонентов для отладки, их мы устанавливать не будем.

1.1.5. После успешной установки вас ждет сообщение об установке.

1.2 Установка Python в Linux

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

```
$ python или $ python3
```

В первом случае, вы запустите Python 2 во втором – Python 3. В будущем, скорее всего, во всех дистрибутивах Linux, включающих Python, будет входить только третья версия. Если у вас, при попытке запустить Python, выдается сообщение о том, что он не установлен, или установлен, но не тот, что вы хотите, то у вас есть два пути: а) собрать Python из исходников; б) взять из репозитория. Для установки из репозитория в Ubuntu воспользуйтесь командой

```
$ sudo apt-get install python3
```

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

Anaconda - это не что иное, как дистрибутив python и R. Если вы работаете в области машинного обучения или науки о данных, то вы найдете anaconda очень полезной. Таким образом, при установке anaconda также будут установлены python, conda (который является менеджером пакетов в anaconda), множество сторонних пакетов python, IDE (например, spyder), jupyter notebook (что очень полезно для написания кодов, визуализации результатов и выполнения кодов по ячейкам). ячейка за ячейкой). Однако, если вы только новичок, установки только python будет достаточно. Python будет иметь определенные стандартные библиотеки, которые будут установлены вместе с ним. И когда вам нужны новые пакеты, вы можете использовать pip для их установки.

3. Как осуществить проверку работоспособности пакета Anaconda?

Необходимо Перейти в папку Scripts и введите в командной строке: input notebook. Если вы находитесь в Windows и открыли папку C:\Anaconda3\Scripts через проводник, то для запуска интерпретатора командной строки для этой папки в поле адреса введите cmd. В результате запустится веб-сервер и среда разработки в браузере. Создайте ноутбук для разработки, для этого нажмите на кнопку New (в правом углу окна) и в появившемся списке выберете Python.

В результате будет создана новая страница в браузере с ноутбуком. Введите в первой ячейке команду `print("Hello word")` и нажмите Alt+Enter на клавиатуре. Ниже ячейки должна появиться соответствующая надпись.

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

Для того чтобы запустить скрипт необходимо передать его интерпретатору Python. Для этого просто откройте терминал с помощью сочетания клавиш Ctrl + Alt + T, перейдите в папку со скриптом и выполните: `python script.py`. Если вы хотите, чтобы после выполнения скрипта открылась консоль, в которой можно интерактивно выполнять команды языка Python используйте опцию -i: `python -i script.py`.

5. Как осуществить запуск программы с помощью IDE PyCharm?

Начните с открытия одного из существующих проектов, хранящихся на компьютере. Вы можете выбрать один из них в списке последних проектов на экране приветствия или нажать кнопку открыть, В противном случае можно создать проект для существующих исходных файлов. Выберите команду открыть в меню Файл и укажите

каталог, в котором существуют источники. Затем PyCharm создаст для вас проект из ваших исходных кодов.

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

Чтобы выполнить программу, достаточно запустить интерпретатор Python, указав имя файла в параметрах:> python sum.py 5050. Интерпретатор выполняет команды одну за одной и в конце завершает свою работу. Такой режим работы интерпретатора называют пакетным. Таким образом, интерпретатор может работать в двух режимах. Интерактивный режим (REPL) помогает нам проверять идеи и придумывать решение задач. Пакетный режим выполняет готовую программу.

7. Почему язык программирования Python называется языком динамической типизации?

«Динамическая» означает, что типы объектов определяются в процессе исполнения программы (runtime). Поэтому питонисты могут не указывать типы переменных. Переменные в Python — это всего лишь указатели на объекты, они не содержат информации о типе.

8. Какие существуют основные типы в языке программирования Python?

В языке Python выделяют несколько типов данных: целые числа, числа с плавающей точкой (вещественные), строки, логический тип. Тип каждой переменной может динамически изменяться по ходу выполнения программы. Определить, какой тип имеет переменная, можно с помощью команды type () Целые числа.

1. None (неопределенное значение переменной)

2. Логические переменные (Boolean Type)

3. Числа (Numeric Type)

3.1. int – целое число

3.2. float – число с плавающей точкой

3.3. complex – комплексное число

4. Списки (Sequence Type)

4.1. list – список

4.2. tuple – кортеж

4.3. range – диапазон

5. Строки (Text Sequence Type)

5.1. str

6. Бинарные списки (Binary Sequence Types)

6.1. bytes – байты

6.2. bytearray – массивы байт

6.3. memoryview – специальные объекты для доступа к внутренним данным объекта через protocol buffer

7. Множества (Set Types)

7.1. set – множество

7.2. frozenset – неизменяемое множество

8. Словари (Mapping Types)

8.1. 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 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?

Имя переменной не должно совпадать с ключевыми словами интерпретатора 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(). Пример использования приведен ниже.

12. Что такое изменяемые и неизменяемые типы в Python?

В Python существуют изменяемые и неизменяемые типы.

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?

Чтобы выполнить деление на цело в Python, можно воспользоваться целочисленным делением. В этом случае результатом будет целое число, без остатка. Целочисленное деление в Python обозначается двумя косыми чертами `//`.

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.

`Cmath` - математические функции для работы с комплексными числами.

Модуль `math` предоставляет доступ к функциям, которые могут выполнять математические действия над комплексными числами.

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

Что такое функция в программировании, узнаем позже. Пока будем считать, что `print()` – это такая команда языка Python, которая выводит то, что в ее скобках на экран.

В `print()` предусмотрены дополнительные параметры. Например, 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. Обычно, если `end` используется, то не в интерактивном режиме, а в скриптах, когда несколько выводов подряд надо разделить не переходом на новую строку, а, скажем, запятыми. Сам переход на новую строку обозначается комбинацией символов `"\n"`. Если присвоить это значение параметру `end`, то никаких изменений в работе функции `print()` вы не увидите, так как это значение и так присвоено по-умолчанию.

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 (`{}`).

До сих пор мы сталкивались с двумя способами записи значений: операторами выражений и функцией `print()`. (Третий способ заключается в использовании метода `write()` файловых объектов; на стандартный выходной файл можно ссылаться как `sys.stdout`. Дополнительные сведения об этом см. в справочнике по библиотеке.)

Часто вам потребуется больше контроля над форматированием выходных данных, чем просто печать значений, разделенных пространством. Существует несколько способов форматирования вывода.

Чтобы использовать форматированные строковые литералы, начните строку с `f` или `F` перед открывающей кавычкой или тройной кавычкой. Внутри этой строки

можно написать выражение Python между символами { и } может ссылаться на переменные или литеральные значения.

Метод строк `str.format()` требует больше ручных усилий. Вы по-прежнему будете использовать { и }, чтобы отметить, где переменная будет заменена, и можете предоставить подробные директивы форматирования, но вам также нужно будет предоставить информацию для форматирования.

Наконец, вы можете выполнить всю обработку строк самостоятельно, используя операции нарезки строк и конкатенации для создания любого макета, который вы можете себе представить. Тип `string` имеет некоторые методы, которые выполняют полезные операции для заполнения строк до заданной ширины столбца.

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

обычно программа работает по такой схеме: получает входные данные → обрабатывает их → выдает результат. Ввод может поступать как непосредственно от пользователя через клавиатуру, так и через внешний источник (файл, база данных).

В стандартной библиотеке Python 3 есть встроенная функция `input()` (в Python 2 это `raw_input()`), которая отвечает за прием пользовательского ввода.