

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 Python3

Выполнила студентка группы

ИТС-б-о-20-1 (2)

Попов Д.В. «        » \_\_\_\_\_ 20\_\_ г.

Подпись студента \_\_\_\_\_

Работа защищена «    » \_\_\_\_\_ 20\_\_ г.

Проверил к.т.н., доцент

Кафедры инфокоммуникаций

Воронкин Р.А.

---

(подпись)

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

Порядок выполнения работы:

1) Создадим общедоступный репозиторий на GitHub, решим задачи с помощью языка программирования Python3, отправим их на GitHub.

Задача 1.

7. Напишите программу (файл *user.py*), которая запрашивала бы у пользователя:

- его имя (например, "What is your name?")
- возраст ("How old are you?")
- место жительства ("Where are you live?")

После этого выводила бы три строки:

```
"This is `имя`"  
"It is `возраст`"  
"(S)he live in `место_жительства`"
```

Рисунок 1. Задача 1.

```
1 print("What is ur name?")  
2 nameUser = input()  
3 print("How old r u?")  
4 ageUser = input()  
5 print("Where r u live?")  
6 cityUser = input()  
7 print("This is {0}.".format(nameUser))  
8 print("It is {0}.".format(ageUser))  
9 print("S(he) live in {0}.".format(cityUser))  
10
```

Рисунок 2. Код для задачи 1.

```
What is ur name?  
Danil  
How old r u?  
18  
Where r u live?  
Stavropol  
This is Danil.  
It is 18.  
S(he) live in Stavropol.
```

Рисунок 3. Работоспособность кода для первой задачи.

Задача 2. Напишите программу (файл arithmetic.py), которая предлагала бы пользователю решить пример  $4 * 100 - 54$ . Потом выводила бы на экран правильный ответ и ответ пользователя. Подумайте, нужно ли здесь преобразовывать строку в число.

```
1 userAnswr= input("Calc this 4*100-54")
2 print("correct is -> ", 346)
3 print("Your answer -> ", userAnswr)
4 |
```

Рисунок 4. Код для второй задачи.

```
Calc this 4*100-54250
correct is -> 346
Tour answer -> 250

Process finished with exit code 0
```

Рисунок 4. Работоспособность кода для второй задачи.

Задача 3. Запросите у пользователя четыре числа (файл numbers.py). 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.

```
1 print('Введите 4 числа:')
2 a=int(input())
3 b=int(input())
4 c=int(input())
5 d=int(input())
6 summ1=a+b
7 summ2=c+d
8 x=summ1/summ2
9 print('%.2f' % x)
10 |
```

Рисунок 5. Код для третьей задачи.

```
Введите 4 числа:  
3  
5  
9  
1  
0.80
```

Рисунок 6. Работоспособность кода для третьей задачи.

Задача 4. Напишите программу (файл individual.py) для решения индивидуального задания. Вариант индивидуального задания уточните у преподавателя.

Вариант 16.

16. Известно значение температуры по шкале Цельсия. Найти соответствующее значение температуры по шкале:

- Фаренгейта;
- Кельвина.

Для пересчета по шкале Фаренгейта необходимо исходное значение температуры умножить на 1,8 и к результату прибавить 32, а по шкале Кельвина абсолютное значение нуля соответствует  $-273,15$  градуса по шкале Цельсия.

```
print("Введите значение температуры в Цельсиях: ")  
a = int(input())  
b = a*1.8+32  
c = a-273.15  
print("по шкале Фаренгейта: ", b)  
print("по шкале Кельвина: ", c)  
|
```

Рисунок 7. Код для индивидуальной задачи.

```
Введите значение температуры в Цельсиях:  
30  
по шкале Фаренгейта: 86.0  
по шкале Кельвина: -243.14999999999998
```

Рисунок 8. Работоспособность кода для четвертой задачи.

Задача повышенной сложности. Вариант 8.

8. Даны два целых числа  $a$  и  $b$ . Если  $a$  делится на  $b$  или  $b$  делится на  $a$ , то вывести 1, иначе – любое другое число. Условные операторы и операторы цикла не использовать.

```

a = int(input('a='))
b = int(input('b='))
c = a % b
v = b % a
print(c*v+1)

```

Рисунок 9. Код для задачи повышенной сложности.

```

a=3
b=3
1

```

Рисунок 10. Работоспособность кода для задачи повышенной сложности.









 12W300 Add files via upload		6fb74cd now	 6 commits
 LICENSE	Initial commit	28 days ago	
 arithmetic.py	Add files via upload	22 minutes ago	
 individual.py	Add files via upload	5 minutes ago	
 individual2.py	Add files via upload	now	
 numbers.py	Add files via upload	15 minutes ago	
 user.py	Add files via upload	31 minutes ago	

Рисунок 11. Страница GitHub с выполненными задачами

Контрольные вопросы:

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

Основные этапы установки Python в Windows и Linux:

- 1) Скачивание дистрибутива с сайта python.org;
- 2) Запуск установочного файла;
- 3) Выбор необходимых опций установки;
- 4) Указание пути установки;
- 5) Установка Python.

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

Пакет Anaconda содержит в себе интерпретатор языка Python 2 и 3 версии, набор наиболее часто используемых библиотек и удобную среду разработки и исполнения.

3. Как осуществить проверку работоспособности пакета Anaconda?

Для проверки работоспособности пакета Anaconda нужно запустить программу Anaconda Prompt, после в появившейся командной строке ввести «jupyter notebook». После этих действий отобразится процесс загрузки веб-среды Jupyter Notebook. В открывшейся среде нужно создать новый ноутбук, после в поле для кода ввести пробный код, например «print(“Hello, World!”)». Если после запуска кода на экране появилась надпись «Hello, World!», то Anaconda установлен правильно.

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

Используемый интерпретатор языка Python в IDE PyCharm задается при создании нового проекта в соответствующей строке.

5. Как осуществить запуск программы с помощью IDE PyCharm?

Для запуска программы с помощью IDE PyCharm нужно открыть код программы, после чего в правом верхнем углу нажать кнопку «Run».

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

В интерактивном режиме работы Python ожидает ввода команд пользователя. При вводе команды интерпретатор выполнит строку и отобразит строкой ниже результат своей работы.

В пакетном режиме работы Python будет только выполнять уже написанный код. Для этого нужно набрать в командной строке "python «название файла».py".

7. Почему язык программирования Python называется языком динамической типизации?

Язык Python называется языком динамической типизации потому, что тип переменной определяется непосредственно при выполнении программы, а не на этапе компиляции, как в языках статической типизации.

8. Какие существуют основные типы в языке программирования Python?

Основные типы данных:

- None (неопределенное значение переменной);
- логические переменные;
- числа;
- списки;
- строки;
- бинарные списки (байты, массивы байт);
- множества;
- словари.

9. Как создаются объекты в памяти? Каково их устройство? В чем заключается процесс объявления новых переменных и работа операции присваивания?

Объект – это абстракция для представления данных, данные – это числа, списки, строки и т.д. Для создания объекта нужно написать его имя, потом поставить знак равенства и значение, с которым объект будет создан.

10. Как получить список ключевых слов в Python?

Для получения списка ключевых слов в Python нужно подключить модуль keyword и воспользоваться командой keyword.kwlist.

11. Каково назначение функций id() и type()?

Функции id() и type() задают идентификатор объекту, тип переменной.

12. Что такое изменяемые и неизменяемые типы в Python.

Изменяемые типы данных – типы данных, которые могут изменяться в процессе выполнения кода программы. К ним относятся списки (list), множества (set) и словари (dict).

Неизменяемые типы данных – типы данных, неизменяемые в процессе выполнения кода программы. К ним относятся целые числа (int), числа с плавающей точкой (float), комплексные числа (complex), логические переменные (bool), кортежи (tuple), строки (str) и неизменяемые множества (frozen set).

13. Чем отличаются операции деления и целочисленного деления?

Деление возвращает частное с его дробной частью (при наличии).

Целочисленное деление возвращает целую часть частного, а дробная часть отбрасывается.

14. Какие имеются средства в языке Python для работы с комплексными числами?

Для работы с комплексными числами используются функции complex(a, b), x.real, x.imag, x.conjugate().

15. Каково назначение и основные функции библиотеки (модуля) math?

По аналогии с модулем math изучите самостоятельно назначение и основные функции модуля cmath.

Библиотека math содержит в себе большое количество часто используемых математических функций, например math.celi(x), math.fabs(x), math.factorial(x) и т.д.

Основные функции библиотеки: math.celi(x), math.fabs(x), math.factorial(x), math.floor(x), math.exp(x), math.log2(x), math.log10(x),



`math.log(x[, base])`, `math.pow(x,y)`, `math.sqrt(x)`, `math.sin(x)`, `math.cos(x)`,  
`math.tan(x)`, `math.acos(x)`, `math.asin(x)`, `math.atan(x)`, `math.pi`, `math.e`.

Модуль `cmath` предоставляет функции для работы с комплексными числами.

Основные функции модуля `cmath`: `cmath.phase(x)`, `cmath.polar(x)`,  
`cmath.rect(x)`, `cmath.exp(x)`, `cmath.log(x[, base])`, `cmath.acosh(x)`, `cmath.asinh(x)`,  
`cmath.atanh(x)`.

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

Параметр `sep` указывает отличный от пробела разделитель строк.

Параметр `end` указывает, что делать после вывода строки (поставить знак, сделать перенос на несколько строк).

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()` применяется к строке и позволяет подставлять типы данных в неё (`%s`, `%d`, `%f`), указывать количество знаков после запятой для чисел (`%lf`).

F-строки позволяют форматировать строки схожим способом, как `format()`, но с некоторыми отличиями. Они позволяют подставить значение в строку, лишь указав имя переменной в фигурных скобках, использовать расширенное форматирование чисел, форматировать дату без вызова метода `strftime()`, совершать базовые арифметические операции прямо в строке, обращаться к значениям списков по индексу и элементам словаря по ключу, и т.д.

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

Ввод с консоли значения переменной производится при помощи функции `input()`.

Вывод: Исследован процесс установки и базовые возможности языка Python3, написано несколько задач.

<https://github.com/12W300/second>