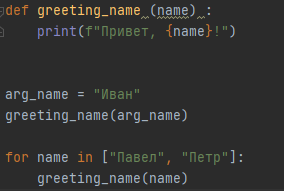
## Урок 1-2. Функции в Python. Полезные особенности

**Задание 1**

Напишите функцию «greeting\_name(name: str) -> None», которая в качестве единственного аргумента принимает имя «name» (являющееся строкой) и выводит строку «Привет, <name>!».

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| arg\_name = “Иван”  greeting\_name(arg\_name) | Привет, Иван! |
| for name in [“Павел”, “Пётр”]:  greeting\_name(name) | Привет, Павел!  Привет, Пётр! |



## Урок 3-4. Функции с возвращением значений

**Задание 2**

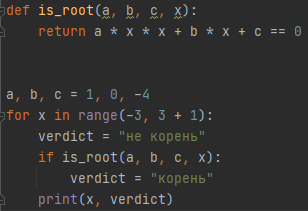
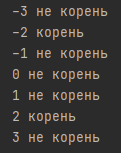
Необходимо написать функцию с сигнатурой   
«is\_root(a: int, b: int, c: int, x: int) -> bool», которая проверяла бы, является ли значение x корнем квадратного уравнения .

Напомним, что является корнем уравнения тогда и только тогда, когда уравнение является истинным.

Таким образом, написанная Вами функция «is\_root» должна принимать в качестве аргументов четыре целых числа, три из которых являются коэффициентами квадратного уравнения, а четвёртое – предполагаемым корнем. Функция должна возвращать логического значение True – если x является корнем, иначе – False.

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| a, b, c = 1, 0, -4  for x in range(-3, 3 + 1):  verdict = “не корень”  if is\_root(a, b, c, x):  verdict = “корень”  print(x, verdict) | -3 не корень  -2 корень  -1 не корень  0 не корень  1 не корень  2 корень  3 не корень |

*Пояснение*: в качестве коэффициентов уравнения и предполагаемого корня были взяты целые числа для упрощения производимых расчётов.

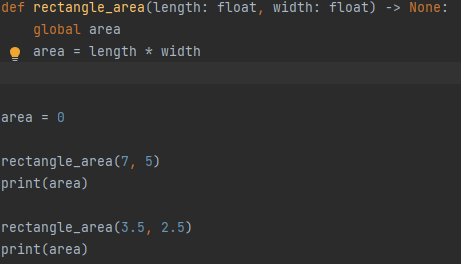
 

## Урок 7-8. Функции. Глобальные и локальные переменные

**Задание 3**

Напишите функцию «rectangle\_area(length: float, width: float) -> None», которая принимает в качестве аргументов длину length и ширину width прямоугольника и записывает значение его площадь в глобальную переменную area.

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| # описание функции  # ...  area = 0  rectangle\_area(7, 5)  print(area)  rectangle\_area(3.5, 2.5)  print(area) | 35  8.75 |

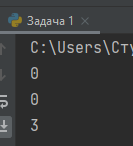
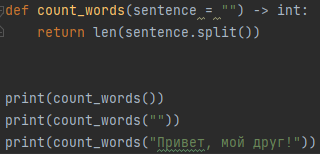
 

## Урок 9-10. Функции. Передача значений через аргументы

**Задание 1**

Напишите функцию «count\_words(sentence: str) -> int», которая принимает предложение sentence в качестве аргумента и возвращает количество слов в предложении, sentence по умолчанию является пустой строкой.

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| # описание функции  # ...  print(count\_words())  print(count\_words(""))  print(count\_words("Привет, мой друг!")) | 0  0  3 |

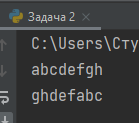
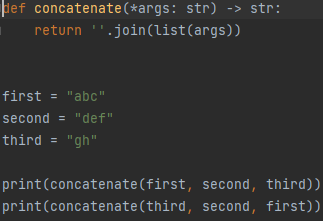


## Урок 11-12. Функции. Передача значений через переменное количество аргументов

**Задание 2**

Напишите функцию «concatenate(\*args: str) -> str», которая принимает произвольное количество строковых аргументов и возвращает строку, полученную путём склеивания (операции конкатенации) строк-аргументов в порядке их перечисления.

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| first = “abc”  second = “def”  third = “gh”  print(concatenate(first, second, third))  print(concatenate(third, second,first)) | abcdefgh  ghdefabc |

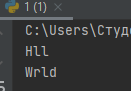
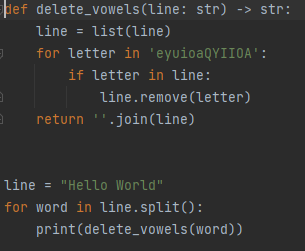


## Урок 15-16. Функции в Python. Решение задач

**Задание 1**

Напишите функцию «delete\_vowels(line: str) -> str», которая на основе строки-аргумента line, состоящей из английских бука, составляет новую строку, из согласных букв изначальной (удаляет все английские гласные буквы).

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| # описание функции  # ...  line = "Hello World"  for word in line.split():  print(delete\_vowels(word)) | Hll  Wrld |



## Урок 19-20. Функции как объект. Применение лямбда-функций

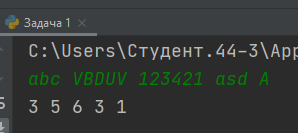
**Задание 1**

Напишите программу, которая принимает список слов, записанных в одной строке через пробел, и выводит последовательность длин этих слов.

В одной строке программа принимает последовательность слов, разделённых пробелом. Программа выводит в одной строке последовательность чисел – длин введённых слов.

*Примечание*: постарайтесь написать программа в одну строку, используя функцию map и методы строк.

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| abc VBDUV 123421 asd A | 3 5 6 3 1 |

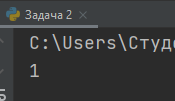
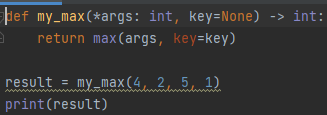


## Урок 21-22. Функции высших порядков

**Задание 2**

Напишите функцию «my\_max(\*args: int, key: function) -> int», которая принимает на вход произвольное количество целых чисел и функцию key – критерий выбора максимума (функция key должна быть функцией, которая принимает один аргумент и возвращает числовое значение, по умолчанию int). Функция my\_max должна возвращать одно из переданных чисел – максимум согласно ключу key.

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| result = my\_max(4, 2, 5, 1)  print(result) | 5 |
| result = my\_max(  4, 2, 5, 1,  key=lambda x: -1 \* x  )  print(result) | 1 |



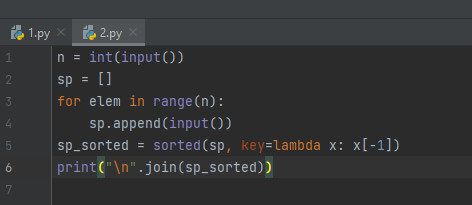
## Урок 23-24. Функции max/min/sorted

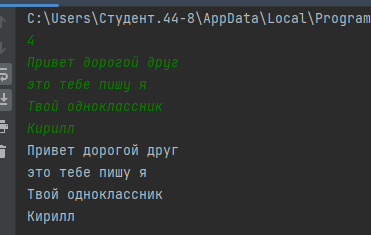
**Задание 2**

Напишите программу, которая принимает список строк и сортирует его в порядке возрастания последнего символа (лексикографически) в каждой строке с помощью функции sorted.

Программа принимает в первой строке натуральное число *n* – количество строк в последовательности. В следующих *n* строках вводится строки последовательности. Программа должна вывести *n* строк в соответствии с условием задачи.

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| 3  привет дорогой друг  это тебе пишу я  твой одноклассник  Кирилл | привет дорогой друг  твой одноклассник  Кирилл  это тебе пишу я |





## Урок 25-26. Функции all/any/filter

**Задание 2**

Напишите программу, которая принимает список чисел и использует функцию all, чтобы проверить, являются ли все элементы списка четными.

Программа принимает на вход последовательность целых чисел произвольной длины, разделённых пробелом. Программа должна вывести «YES», если среди введённых чисел все чётные, иначе – «NO».

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| -4 2 6 -8 144 | YES |
| 0 2 1 6 8 | NO |

