Ответы на вопросы по проге, 14.03.2020

1.Аргументы функций на языке Python.

Вызывая функцию, мы можем передавать ей следующие типы аргументов:

**Обязательные аргументы** (Required arguments)

**Аргументы-ключевые слова** (Keyword argument)

**Аргументы по умолчанию** (Default argument)

**Аргументы произвольной длины** (Variable-length argumens)

В программировании, если нам нужно выполнять похожие действия, мы определяем функции для многоразового использования кода. Чтобы выполнить это действие, мы вызываем функцию с определённым значением — аргументом.

Предположим, у нас есть функция, которая складывает три числа. Во фрагменте кода выше у нас есть функция adder() с тремя аргументами: x, y и z. При передаче трёх значений этой функции на выходе мы получаем их сумму. Но что, если передать больше трёх аргументов в эту функцию?  
В Python можно передать переменное количество аргументов двумя способами:

\*args для неименованных аргументов;

\*\*kwargs для именованных аргументов.

Мы используем \*args и \*\*kwargs в качестве аргумента, когда заранее не известно, сколько значений мы хотим передать функции. Если поставить \* перед именем, это имя будет принимать не один аргумент, а несколько. Аргументы передаются как кортеж и доступны внутри функции под тем же именем, что и имя параметра, только без \*. По аналогии с \*args мы используем \*\*kwargs для передачи переменного количества именованных аргументов. Схоже с \*args, если поставить \*\* перед именем, это имя будет принимать любое количество именованных аргументов. Кортеж/словарь из нескольких переданных аргументов будет доступен под этим именем.

2.