

Практическое задание.

Тема: Функции.

Теоретическая часть.

Функции в Python представляют собой блоки кода, которые могут быть вызваны для выполнения определенных задач.

Синтаксис объявления (создания) функции:

```
def name_func(arg):  
    code...
```

- `def` – ключевое слово для объявления функции
- `name_func` – произвольное имя функции
- `arg` – аргументы функции (имеют любые названия)
- `code` – код, который будет выполнять функция

Синтаксис объявления (создания) функции:

```
name_func(arg)
```

Вызов происходит по имени функции и аргументами (если они есть).

Например, создадим функцию, которая проверяет длину сообщения:

```
message = "I'm learning functions in Python"  
  
def exam(text):  
    return len(text)  
  
print(exam('random text'))  
print(exam(message))
```

На выходе получаем:

```
>>> 11  
>>> 32
```

Практическая часть.

Задание 1.

Создать функцию `upper` с атрибутом `t`, принимающий в себя строку. Вывести только заглавные буквы данной строки. Если их нет, то не выводить ничего.

Входные данные:

```
>>> upper('PriVet')
```

Выходные данные:

```
>>> P V
```

Для решения можно использовать цикл `for` внутри функции и метод `isupper()`.

Задание 2.

Реализовать функцию `punct` с атрибутом `txt`, принимающий в себя строку. Данная функция должна проверять строку на знаки: `! ? . , ()`

Если данные знаки есть в строке, то вывести их количество.

Входные данные:

```
>>> punct('(Как дела?'))
```

Выходные данные:

```
>>> 3
```

Задание 3.

Реализуйте функцию `create_cube` с двумя атрибутами, которая создаёт квадрат из символа звёздочки по оси `x` и `y`. `x` – длина квадрата, `y` – высота квадрата.

Входные данные:

```
>>> create_cube(5, 3)
```

Выходные данные:

```
>>> *****
```

```
>>> *****
```

```
>>> *****
```

Задание 4.

Создайте функцию `double`, с атрибутом, включающий в себя строку. Данная функция будет удваивать каждый символ строки.

Входные данные:

```
>>> double('строка')
```

Выходные данные:

```
>>> ссттррооккаа
```

Задание 5.

В готовый набор конструктора входит: 72 детали, 4 фигурки людей, 2 машины и 7 игрушечных деревьев.

Создать функцию, рассчитывающую количество готовых наборов конструктора, по переданным данным в атрибуты функции.

Пример входных данных 1:

```
>>> Constructor(144, 8, 4, 14)
```

Пример выходных данных 1:

```
>>> 2
```

Пример входных данных 2:

```
>>> Constructor(10000, 16, 6, 2)
```

Пример выходных данных 2:

```
>>> 0
```

Для выполнения задания, можно воспользоваться функцией `min` и передать ей минимальные значения каждого атрибута.

Задание 6.

Создайте функцию принимающая в себя 2 атрибута.

1 атрибут – длина списка.

2 атрибут – числа списка.

При вызове данная функция будет генерировать список чисел по атрибутам. Если второй атрибут (числа списка) отсутствует, то в списке выведутся нули.

Пример 1.

Входные данные:

```
>>> create_list(5, 3)
```

Выходные данные:

```
>>> [3, 3, 3, 3, 3]
```

Пример 2.

Входные данные:

```
>>> create_list(3)
```

Выходные данные:

```
>>> [0, 0, 0]
```

Задание 7.

Напишите функцию, которая на вход принимает список, с любыми типами данных, находит в этом списке все целые числа, отбирает из них все те, что делятся нацело на 7, суммирует их, а затем проверяет превышает ли эта сумма 83 или нет. Если нет, то вывести True, если да False.

Входные данные:

```
print(custom_filter([7, 10.5, 'txt', 14, 2, 56]))
```

Выходные данные:

```
>>> сумма: 77
```

```
>>> True
```

Задание 8.

Создайте функцию, принимающую на вход 2 числа (x, y) и выводящую плоскость, состоящую из этих чисел, обрамлённую определёнными символами.

Например:

```
square(2, 3)
```

На выходе получим:

```
  _ _ _  
|1||2||3|  
|1||2||3|  
  _ _ _
```

Над числами должен быть символ нижнего подчёркивания « », перед и после чисел, горизонтальная черта «|», а под ними верхнее подчёркивание « »