Листок 07. Функции в Python

Замечания

- для решение приведенных ниже упражнений требуется создавать (определять) пользовательские функции.
- при решении упражнений полученные результаты выведите на экран с помощью функции print().
- позаботесь о том, чтобы выводимый на экран результат был снабжен информацией о нем, там где это необходимо.
- не забывайте писать комментарии к вашему коду

Упражнения
Функции с позиционными аргументами

1. Напишите функцию **sum_two_numbers**, принимающую 2 аргумента – числа a и b – и возвращающую 1 значение: их сумму.
2. Напишите функцию **is_even**, принимающую 1 аргумент – число n – и возвращающую 1 значение: **True**, если число чётное, и **False** в противном случае.
3. Напишите функцию **max_of_three**, принимающую 3 аргумента – числа **a**, **b** и **c** – и возвращающую 1 значение: наибольшее из них. **Не используйте встроенную функцию max()**.
4. Напишите функцию **is_vowel**, принимающую 1 аргумент – символ **ch** – и возвращающую 1 значение: **True**, если он является гласной, и **False** в противном случае.
5. Напишите функцию **count_vowels**, принимающую 1 аргумент – строку **s** – и возвращающую 1 значение: количество гласных в этой строке.
6. Напишите функцию **factorial**, принимающую 1 аргумент – число **n** – и возвращающую 1 значение: его факториал. **Решите задачу двумя способами: рекурсивно и итеративно.**
7. Напишите функцию **reverse_string**, принимающую 1 аргумент – строку **s** – и возвращающую 1 значение: её в обратном порядке.
8. Напишите функцию **is_palindrome**, принимающую 1 аргумент – строку **s** – и возвращающую 1 значение: **True**, если строка является палиндромом, и **False** в противном случае.
9. Напишите функцию **is_palindrome_ignore_spaces**, принимающую 1 аргумент – строку **s** – и возвращающую 1 значение: **True**, если строка является палиндромом, игнорируя регистр и пробелы, и **False** в противном случае.
10. Напишите функцию **remove_duplicates**, принимающую 1 аргумент – список **lst** – и возвращающую 1 значение: новый список без повторяющихся элементов.

11. Напишите функцию **fibonacci**, принимающую 1 аргумент – число **n** – и возвращающую 1 значение: n-ое число Фибоначчи. **Решите задачу двумя способами: рекурсивно и итеративно.**

12. Напишите функцию **sum_list**, принимающую 1 аргумент – список чисел **lst** – и возвращающую 1 значение: их сумму. **При решении не используйте встроенные функции.**

13. Напишите функцию **find_min**, принимающую 1 аргумент – список чисел **lst** – и возвращающую 1 значение: минимальное значение в списке. **При решении не используйте встроенные функции.**

14. Напишите функцию **find_max**, принимающую 1 аргумент – список чисел **lst** – и возвращающую 1 значение: максимальное значение в списке. **При решении не используйте встроенные функции.**

15. Напишите функцию **is_prime**, принимающую 1 аргумент – число n – и возвращающую 1 значение: **True**, если число простое, и **False** в противном случае.

16. Напишите функцию **sort_list**, принимающую 1 аргумент – список чисел lst – и возвращающую 1 значение: список, отсортированный по возрастанию. **При решении не используйте встроенные функции.**

17. Напишите функцию **average**, принимающую 1 аргумент – список чисел **lst** – и возвращающую 1 значение: их среднее значение.

18. Напишите функцию **merge_lists**, принимающую 2 аргумента – списки **lst1** и **lst2** – и возвращающую 1 значение: объединённый список.

19. Напишите функцию **capitalize_words**, принимающую 1 аргумент – строку **s** – и возвращающую 1 значение: строку, где каждое слово начинается с заглавной буквы.

20. Напишите функцию **count_occurrences**, принимающую 2 аргумента – список **lst** и элемент **elem** – и возвращающую 1 значение: количество вхождений элемента **elem** в список **lst**. **При решении не используйте встроенные функции и методы.**

21. Напишите функцию **remove_vowels**, принимающую 1 аргумент – строку **s** – и возвращающую 1 значение: строку без гласных букв.

22. Напишите функцию **power**, принимающую 2 аргумента – число **base** и степень **exp** – и возвращающую 1 значение: результат возведения числа **base** в степень **exp**. **При решении не используйте встроенные функции.**

23. Напишите функцию **swap_first_last**, принимающую 1 аргумент – список **lst** – и возвращающую 1 значение: список с первым и последним элементами, поменянными местами.

24. Напишите функцию **count_characters**, принимающую 1 аргумент – строку **s** – и возвращающую 1 значение: количество символов в строке.

25. Напишите функцию **to_uppercase**, принимающую 1 аргумент – строку **s** – и возвращающую 1 значение: строку в верхнем регистре.

26. Напишите функцию **to_lowercase**, принимающую 1 аргумент – строку **s** – и возвращающую 1 значение: строку в нижнем регистре.

27. Напишите функцию **find_second_largest**, принимающую 1 аргумент – список чисел **lst** – и возвращающую 1 значение: второе по величине число.

28. Напишите функцию **sum_of_squares**, принимающую 1 аргумент – список чисел **lst** – и возвращающую 1 значение: сумму их квадратов.

29. Напишите функцию **find_longest_word**, принимающую 1 аргумент – строку **s** – и возвращающую 1 значение: самое длинное слово в строке.

30. Напишите функцию **count_words**, принимающую 1 аргумент – строку **s** – и возвращающую 1 значение: количество слов в строке.

31. Напишите функцию **remove_negatives**, принимающую 1 аргумент – список **lst** – и возвращающую 1 значение: новый список без отрицательных чисел.

32. Напишите функцию **multiply_list**, принимающую 1 аргумент – список чисел **lst** – и возвращающую 1 значение: произведение всех элементов списка.

33. Напишите функцию **is_anagram**, принимающую 2 аргумента – строки **s1** и **s2** – и возвращающую 1 значение: **True**, если одна строка является анаграммой другой, и **False** в противном случае.

34. Напишите функцию **unique_elements**, принимающую 1 аргумент – список **lst** – и возвращающую 1 значение: новый список только с уникальными элементами.

35. Напишите функцию **is_sublist**, принимающую 2 аргумента – списки **lst1** и **lst2** – и возвращающую 1 значение: **True**, если **lst1** является подсписком **lst2**, и **False** в противном случае.

36. Напишите функцию **find_median**, принимающую 1 аргумент – список чисел **lst** – и возвращающую 1 значение: медиану списка.

37. Напишите функцию **remove_evens**, принимающую 1 аргумент – список **lst** – и возвращающую 1 значение: новый список без чётных чисел.

38. Напишите функцию **largest_negative**, принимающую 1 аргумент – список **lst** – и возвращающую 1 значение: наибольшее отрицательное число в списке.

39. Напишите функцию **count_multiples_of_three**, принимающую 1 аргумент – список **lst** – и возвращающую 1 значение: количество чисел, кратных трём, в списке.

40. Напишите функцию **join_with_comma**, принимающую 1 аргумент – список строк **lst** – и возвращающую 1 значение: строку с элементами, разделёнными запятой.

41. Напишите функцию **power_list**, принимающую 2 аргумента – список чисел **lst** и степень **exp** – и возвращающую 1 значение: новый список с элементами, возведёнными в степень **exp**.

42. Напишите функцию **abs_negatives**, принимающую 1 аргумент – список **lst** – и возвращающую 1 значение: список с абсолютными значениями отрицательных чисел.

43. Напишите функцию **fahrenheit_to_celsius**, принимающую 1 аргумент – температуру **f** в градусах Фаренгейта – и возвращающую 1 значение: температуру в градусах Цельсия.

44. Напишите функцию **is_alpha**, принимающую 1 аргумент – строку **s** – и возвращающую 1 значение: **True**, если строка содержит только буквы, и **False** в противном случае.

45. Напишите функцию **remove_spaces**, принимающую 1 аргумент – строку **s** – и возвращающую 1 значение: строку без пробелов.

46. Напишите функцию **fibonacci_sequence**, принимающую 1 аргумент – число **n** – и возвращающую 1 значение: список из первых n чисел Фибоначчи.

47. Напишите функцию **all_even**, принимающую 1 аргумент – список **lst** – и возвращающую 1 значение: **True**, если все элементы списка чётные, и **False** в противном случае.

48. Напишите функцию **square_all**, принимающую 1 аргумент – список чисел **lst** – и возвращающую 1 значение: новый список с квадратами всех элементов.

49. Напишите функцию **count_long_words**, принимающую 2 аргумента – строку **s** и число **n** – и возвращающую 1 значение: количество слов в строке длиной больше n.

50. Напишите функцию **squares_dict**, принимающую 1 аргумент – число **n** – и возвращающую 1 значение: словарь, где ключи – числа от 1 до n, а значения – их квадраты.

51. Напишите функцию **count_above_average**, принимающую 1 аргумент – список чисел **lst** – и возвращающую 1 значение: количество чисел в списке, которые больше среднего значения списка.

52. Напишите функцию **find_all_indices**, принимающую 2 аргумента – список **lst** и элемент **elem** – и возвращающую 1 значение: список всех индексов элемента **elem** в списке **lst**.

53. Напишите функцию **flip_signs**, принимающую 1 аргумент – список чисел **lst** – и возвращающую 1 значение: список с противоположными знаками элементов.

54. Напишите функцию **acronym**, принимающую 1 аргумент – строку **s** – и возвращающую 1 значение: строку, состоящую из первых букв каждого слова в s.

55. Напишите функцию **filter_even_length_words**, принимающую 1 аргумент – строку **s** – и возвращающую 1 значение: список слов чётной длины.

56. Напишите функцию **cube_all**, принимающую 1 аргумент – список чисел **lst** – и возвращающую 1 значение: новый список с кубами элементов.

57. Напишите функцию **is_substring**, принимающую 2 аргумента – строки **s1** и **s2** – и возвращающую 1 значение: **True**, если **s1** является подстрокой **s2**, и **False** в противном случае.

58. Напишите функцию **remove_spaces**, принимающую 1 аргумент – строку **s** – и возвращающую 1 значение: строку без пробелов.

59. Напишите функцию **is_strictly_increasing**, принимающую 1 аргумент – список lst – и возвращающую 1 значение: **True**, если список является строго возрастающей последовательностью, и **False** в противном случае.

60. Напишите функцию **min_max**, принимающую 1 аргумент – список чисел **lst** – и возвращающую 1 значение: кортеж с минимальным и максимальным элементами списка.

61. Напишите функцию **alternate_case**, принимающую 1 аргумент – строку **s** – и возвращающую 1 значение: строку с чередующимся регистром символов.

62. Напишите функцию **flatten_list**, принимающую 1 аргумент – список списков **lst** – и возвращающую 1 значение: плоский список.

63. Напишите функцию **sum_of_digits**, принимающую 1 аргумент – число **n** – и возвращающую 1 значение: сумму его цифр.

64. Напишите функцию **common_elements**, принимающую 2 аргумента – списки **lst1** и **lst2** – и возвращающую 1 значение: список общих элементов.

65. Напишите функцию **second_smallest**, принимающую 1 аргумент – список чисел **lst** – и возвращающую 1 значение: второе по величине минимальное число.

66. Напишите функцию **is_pangram**, принимающую 1 аргумент – строку **s** – и возвращающую 1 значение: **True**, если строка содержит все буквы алфавита хотя бы один раз, и **False** в противном случае.

67. Напишите функцию **reverse_words**, принимающую 1 аргумент – строку **s** – и возвращающую 1 значение: строку с словами в обратном порядке.

68. Напишите функцию **sum_of_cubes**, принимающую 1 аргумент – список чисел **lst** – и возвращающую 1 значение: сумму их кубов.

69. Напишите функцию **replace_vowels**, принимающую 2 аргумента – строку **s** и символ **ch** – и возвращающую 1 значение: строку с гласными, заменёнными на **ch**.

70. Напишите функцию **has_equal_char_frequency**, принимающую 1 аргумент – строку **s** – и возвращающую 1 значение: **True**, если строка содержит одинаковое количество каждой буквы, и **False** в противном случае.

71. Напишите функцию **multiply_list**, принимающую 1 аргумент – список чисел **lst** – и возвращающую 1 значение: произведение всех элементов списка.

72. Напишите функцию **unique_words**, принимающую 1 аргумент – список строк **lst** – и возвращающую 1 значение: строку, содержащую только уникальные слова.

73. Напишите функцию **count_vowels_consonants**, принимающую 1 аргумент – строку **s** – и возвращающую 1 значение: кортеж с количеством гласных и согласных.

74. Напишите функцию **count_unique**, принимающую 1 аргумент – список **lst** – и возвращающую 1 значение: количество уникальных чисел в списке.

75. Напишите функцию **find_pairs_with_sum**, принимающую 2 аргумента – список чисел **lst** и число **sum** – и возвращающую 1 значение: список пар чисел, сумма которых равна **sum**.

76. Напишите функцию **can_sort_with_one_swap**, принимающую 1 аргумент – список чисел **lst** – и возвращающую 1 значение: **True**, если список можно упорядочить по возрастанию с одной заменой элемента, и **False** в противном случае.

77. Напишите функцию **filter_by_length**, принимающую 2 аргумента – список строк **lst** и число **n** – и возвращающую 1 значение: список слов длиной больше или равной n.

78. Напишите функцию **permutations**, принимающую 1 аргумент – строку s – и возвращающую 1 значение: список всех возможных перестановок символов строки.

79. Напишите функцию **all_sublists**, принимающую 1 аргумент – список lst – и возвращающую 1 значение: список всех возможных подсписков.

80. Напишите функцию **word_frequency**, принимающую 1 аргумент – строку s – и возвращающую 1 значение: словарь, где ключ – слово из строки, а значение – частота его появления.

81. Напишите функцию **remove_duplicates**, принимающую 1 аргумент – список lst – и возвращающую 1 значение: список без элементов, встречающихся более одного раза.

82. Напишите функцию **to_title_case**, принимающую 1 аргумент – строку s – и возвращающую 1 значение: строку в формате заголовка.

83. Напишите функцию **sentence_count**, принимающую 1 аргумент – строку **s** – и возвращающую 1 значение: количество предложений в строке.

84. Напишите функцию **all_strings_unique**, принимающую 1 аргумент – список строк **lst** – и возвращающую 1 значение: **True**, если все строки в списке уникальны, и **False** в противном случае.

85. Напишите функцию **count_less_than**, принимающую 2 аргумента – список чисел **lst** и число **n** – и возвращающую 1 значение: количество чисел в списке, которые меньше n.

86. Напишите функцию **multiples**, принимающую 2 аргумента – числа **n** и **m** – и возвращающую 1 значение: список из первых n чисел, кратных m.

87. Напишите функцию **filter_positive**, принимающую 1 аргумент – список чисел **lst** – и возвращающую 1 значение: список только с положительными числами.

88. Напишите функцию **unique_random_numbers**, принимающую 2 аргумента – числа **n** и **range_max** – и возвращающую 1 значение: список из n уникальных случайных чисел в диапазоне от 0 до **range_max**.

89. Напишите функцию **capitalize_alternate**, принимающую 1 аргумент – строку **s** – и возвращающую 1 значение: строку с каждой второй буквой заглавной.

90. Напишите функцию **rotate_list**, принимающую 2 аргумента – список **lst** и число **n** – и возвращающую 1 значение: список, циклически сдвинутый на n позиций.

91. Напишите функцию **divisors**, принимающую 1 аргумент – число **n** – и возвращающую 1 значение: список всех делителей числа n.

92. Напишите функцию **sum_of_digits_squared**, принимающую 1 аргумент – число **n** – и возвращающую 1 значение: сумму квадратов цифр числа.

93. Напишите функцию **square_properties**, принимающую 1 аргумент – сторону квадрата **a** – и возвращающую 3 значения (с помощью кортежа): периметр квадрата, площадь квадрата и диагональ квадрата.

94. Напишите функцию **arithmetic_progression**, принимающую 3 позиционных аргумента a, n и d: первый a - первый член арифметической прогрессии, второй n - номер члена арифметической прогрессии (n натуральное число), третий d - разность арифметической прогрессии, - и возвращающую 1 значение: n-ый член арифметической прогрессии.

95. Напишите функцию **geometric_progression**, принимающую 3 позиционных аргумента b, n и q: первый b - первый член геометрической прогрессии, второй n - номер члена геометрической прогрессии (n натуральное число), третий q - знаменатель геометрической прогрессии, - и возвращающую 1 значение: n-ый член геометрической прогрессии.

96. Напишите функцию **perform_operation**, принимающую 3 позиционных аргумента: первые два – числа **a** и **b**, третий – операция **op**, которая должна быть произведена над ними. Если третий аргумент ‘+‘, сложить их; если ‘-‘, то вычесть (из первого второе); ‘*‘ - умножить; ‘/‘ - разделить (первое на второе). В остальных случаях вернуть строку «Неизвестная операция».

97. Напишите функцию **compound_interest**, принимающую 2 аргумента – сумму вклада **a** и срок вклада **years** – и возвращающую 1 значение: сумму, которая будет на счету пользователя через years лет при годовой процентной ставке 10%.

Функции с именованными аргументами

1. Напишите функцию **arithmetic_progression_sum**, принимающую 3 именованных аргумента a, n и d: первый a - первый член арифметической прогрессии, второй n - число первых членов арифметической прогрессии (n натуральное число), третий d - разность арифметической прогрессии, - и возвращающую 1 значение: сумму n первых членов арифметической прогрессии. По умолчанию $d = 1$.

2. Напишите функцию **geometric_progression_sum**, принимающую 3 именованных аргумента b, n и q: первый b - первый член геометрической прогрессии, второй n - число первых членов геометрической прогрессии (n натуральное число), третий q - знаменатель геометрической прогрессии, - и возвращающую 1 значение: сумму n первых членов геометрической прогрессии. По умолчанию $q = 1$.

3. Напишите функцию **polynomial_value**, принимающую 7 именованных аргументов: x, a_0, a_1, a_2, a_3, a_4 и a_5, - и возвращающую 1 значение: значение многочлена пятой степени $p(x) = a_0x^5 + a_1x^4 + a_2x^3 + a_3x^2 + a_4x + a_5$ с действительными коэффициентами. По умолчанию $a_i = 1$ для всех i.

4. Напишите функцию **calculate_bonus**, принимающую 4 аргумента: два именованных аргумента **name** и **last_name** – имя и фамилию сотрудника, и два аргумента по умолчанию – **salary=120000** и **bonus=10** (оклад и премия), - и возвращающую 2 значения: размер новогодней премии для сотрудника и зарплату с учетом премии (см. примеры вызова).

Примеры вызова функции

```
calculate_bonus('Алина', 'Тимофеева', salary=150000, bonus=25)
calculate_bonus('Алексей', 'Ковалев', bonus=15)
calculate_bonus('Игорь', 'Ефимов')
calculate_bonus('Анастасия', 'Яковлева', salary=100000, bonus=20)
```

Вывод

```
Новогодняя премия сотрудника Алина Тимофеева: 37500.00 руб.
Оклад: 150000.00 руб.
Всего к выдаче: 187500.00 руб.

Новогодняя премия сотрудника Алексей Ковалев: 18000.00 руб.
Оклад: 120000.00 руб.
Всего к выдаче: 138000.00 руб.

Новогодняя премия сотрудника Игорь Ефимов: 12000.00 руб.
Оклад: 120000.00 руб.
Всего к выдаче: 132000.00 руб.

Новогодняя премия сотрудника Анастасия Яковлева: 20000.00 руб.
Оклад: 100000.00 руб.
Всего к выдаче: 120000.00 руб.
```

5. Напишите функцию **calculate_discount**, принимающую 3 аргумента: два именованных аргумента **price** и **discount** – цена товара и размер скидки, и один аргумент по умолчанию **quantity=1** (количество товаров), - и возвращающую 1 значение: общую стоимость покупки с учетом скидки.

6. Напишите функцию **calculate_bmi**, принимающую 2 именованных аргумента – вес (**weight**) и рост (**height**) человека – и возвращающую 1 значение: индекс массы тела (**BMI**).

7. Напишите функцию **calculate_interest**, принимающую 3 именованных аргумента – сумму вклада (principal), процентную ставку (**rate**) и срок вклада в годах (**years**) – и возвращающую 1 значение: сумму процентов, начисленных за этот срок.

8. Напишите функцию **calculate_discount**, принимающую 2 именованных аргумента – цену товара (**price**) и процент скидки (**discount**) – и возвращающую 1 значение: цену товара после применения скидки.

9. Напишите функцию **calculate_future_value**, принимающую 3 именованных аргумента – сумму вклада (**principal**), процентную ставку (**rate**) и срок вклада в годах (**years**) – и возвращающую 1 значение: будущую стоимость вклада.

10. Напишите функцию **calculate_age**, принимающую 1 именованных аргумент – дату рождения в формате 'YYYY-MM-DD' (**birthdate**) – и возвращающую 1 значение: возраст человека в годах.

11. Напишите функцию **calculate_days_between_dates**, принимающую 2 именованных аргумента – две даты (**date1** и **date2**) в формате 'YYYY-MM-DD' – и возвращающую 1 значение: количество дней между ними.

Функции с переменным числом аргументов

1. Напишите функцию **average_numbers**, принимающую произвольное количество целых чисел (позиционных аргументов) и возвращающую 1 значение: среднее арифметическое без использования встроенных функций `sum()` и `len()`.

2. Напишите функцию **sum_of_squares**, принимающую произвольное количество числовых аргументов и возвращающую 1 значение: сумму их квадратов.

3. Напишите функцию **to_uppercase_words**, принимающую произвольное количество строк (позиционных аргументов) и возвращающую список строк, где каждое слово преобразовано в верхний регистр.

4. Напишите функцию **concatenate_strings**, принимающую произвольное количество строк (позиционных аргументов) и возвращающую 1 значение: объединённую строку.

5. Напишите функцию **max_number**, принимающую произвольное количество чисел (позиционных аргументов) и возвращающую 1 значение: наибольшее из них.

6. Напишите функцию **min_number**, принимающую произвольное количество чисел (позиционных аргументов) и возвращающую 1 значение: наименьшее из них.

7. Напишите функцию **reverse_strings**, принимающую произвольное количество строк (позиционных аргументов) и возвращающую список строк, где каждая строка перевёрнута.

8. Напишите функцию **count_vowels**, принимающую произвольное количество строк (позиционных аргументов) и возвращающую список, где каждый элемент представляет количество гласных в соответствующей строке.

9. Напишите функцию **factorial_numbers**, принимающую произвольное количество целых чисел (позиционных аргументов) и возвращающую список, где каждый элемент представляет факториал соответствующего числа.

10. Напишите функцию **is_palindrome**, принимающую произвольное количество строк (позиционных аргументов) и возвращающую список, где каждый элемент представляет **True**, если строка является палиндромом, и **False** в противном случае.

11. Напишите функцию **remove_spaces**, принимающую произвольное количество строк (позиционных аргументов) и возвращающую список строк без пробелов.

12. Напишите функцию **sort_numbers**, принимающую произвольное количество чисел (позиционных аргументов) и возвращающую список, отсортированный по возрастанию.

13. Напишите функцию **unique_elements**, принимающую произвольное количество аргументов и возвращающую список уникальных элементов.

14. Напишите функцию **count_characters**, принимающую произвольное количество строк (позиционных аргументов) и возвращающую список, где каждый элемент представляет количество символов в соответствующей строке.

15. Напишите функцию **multiply_numbers**, принимающую произвольное количество чисел (позиционных аргументов) и возвращающую 1 значение: произведение всех чисел.

16. Напишите функцию **is_prime**, принимающую произвольное количество целых чисел (позиционных аргументов) и возвращающую список, где каждый элемент представляет **True**, если число простое, и **False** в противном случае.

17. Напишите функцию **join_with_hyphen**, принимающую произвольное количество строк (позиционных аргументов) и возвращающую 1 значение: строку, где все элементы соединены дефисом.

18. Напишите функцию **cube_numbers**, принимающую произвольное количество чисел (позиционных аргументов) и возвращающую список, где каждый элемент представляет куб соответствующего числа.

19. Напишите функцию **reverse_words**, принимающую произвольное количество строк (позиционных аргументов) и возвращающую список строк, где слова в каждой строке расположены в обратном порядке.