



Базовые алгоритмы на Python

Привет! Мы продолжаем наше погружение в автоматизацию тестирования.

Сегодня у нас несколько задач.

Мы настроим рабочее окружение и на практике попробуем создавать переменные и вызывать их, а также создавать функции, используя Python.

В результате у вас получится ссылка на репозиторий, в котором будут все файлы с решением заданий.

Презентация к уроку:

[L2.pdf](#)

Задание 0. Приведение типов

1. Создайте файл `lesson_2_task_0.py`.
2. Создайте переменную `my_age`.
3. Задайте значение переменной через функцию `input`.
4. Запустите скрипт и введите в консоль ваш возраст.
5. Выведите на экран сообщение в формате «Ваш возраст: `my_age`».
6. Сохраните в переменную `my_age` эту же переменную + 1.

▼ Подсказка

```
my_age = my_age + 1
```

7. Выведите в консоль обновленный возраст.
8. Запустите скрипт и укажите ваш возраст.
9. Внимательно изучите текст ошибки в консоли.
10. Введенное значение является строкой. Необходимо преобразовать значение `my_age` к типу «число».

Используйте для этого функцию `int()`.

Пример: `my_age = int("23")` #запишется число 23, а не строка "23"

11. Перезапустите скрипт.

Задание 1. Работа со списками

1. Создайте файл `lesson_2_task_1.py`.
2. Дан список `lst = ['🍇', '🍎', '🍏', '🍊', '🍌', '🍎']`.
3. Выведите на экран первый и последний элементы.

Задание 2. Високосный год

1. Создайте файл `lesson_2_task_2.py`.
2. Создайте функцию `is_year_leap`, принимающую 1 аргумент — год (число) — и возвращающую `True`, если год високосный, и `False` — иначе.

Год високосный, если его номер делится на 4 без остатка.

Например, 2020 или 2008. 2009 или 2023 не делится на 4 без остатка, значит, год не високосный.

Оператор `%` возвращает **остаток** от деления.

Примеры использования оператора

`%` :

1. $12 \% 3 = 0$ (т.к. $12 : 3 = 4 + 0$)
2. $13 \% 3 = 1$ (т.к. $13 : 3 = 4 + 1$)
3. $14 \% 3 = 2$ (т.к. $14 : 3 = 4 + 2$)
4. $15 \% 3 = 0$ (т.к. $15 : 3 = 5 + 0$)
5. $11 \% 3 = 2$ (т.к. $11 : 3 = 3 + 2$)

3. В этом же файле напишите код, который вызывает функцию и передает в нее год (выберите любой).

4. Результат вызова функции должен сохраняться в переменную.

5. Выведите в консоль ответ: `год <номер года>: <True|False>`

▼ Пример

1. `год 2024: True`

2. `год 2023: False`

Задание 3. Площадь квадрата

1. Создайте файл `lesson_2_task_3.py`.

2. Напишите функцию `square`, принимающую 1 аргумент — сторону квадрата — и возвращающую площадь квадрата.

*Площадь квадрата = сторона * сторона.*

3. Если переданный аргумент был не целым, округлите результат вверх.

Задание 4. FizzBuzz. Задача с собеседования

1. Создайте файл `lesson_2_task_4.py`.

2. Напишите функцию `fizz_buzz`, которая принимает один аргумент — `n` (число).

3. Функция должна печатать числа от 1 до `n`. При этом:

a. если число делится на 3, печатать `Fizz` ;

b. если число делится на 5, печатать `Buzz` ;

c. если число делится на 3 и на 5, печатать `FizzBuzz` .

▼ Пример: `fizz_buzz(17)` →

1

2

Fizz

4

Buzz
Fizz
7
8
Fizz
Buzz
11
Fizz
13
14
FizzBuzz
16
17

Задание 5. Месяц — сезон

1. Создайте файл `lesson_2_task_5.py`.
2. Напишите функцию `month_to_season()`, которая принимает 1 аргумент — номер месяца — и возвращает название сезона, к которому относится этот месяц.
Например, передаем 2, на выходе получаем «Зима».

Задание 6. Фильтрация списка

1. Создайте файл `lesson_2_task_6.py`.
2. Дан список `lst = [11, 5, 8, 32, 15, 3, 20, 132, 21, 4, 555, 9, 20]`.
3. Необходимо вывести элементы, которые одновременно:
 - a. меньше 30,
 - b. делятся на 3 без остатка.

Задание 7. Сумма элементов списка

1. Создайте файл `lesson_2_task_7.py`.

2. Дан список `lst = [11, 5, 8, 32, 15, 3, 20, 132, 21, 4, 555, 9, 20]`.
3. Выведите сумму всех элементов списка.

Задание 8. Range

1. Изучите статью: <https://pythonchik.ru/osnovy/python-range>.
2. Создайте файл `lesson_2_task_8.py`.
3. Создайте список `[18, 14, 10, 6, 2]` с помощью функции `range()` и выведите его на экран.

Задание 9. Поменять значения местами

1. Создайте файл `lesson_2_task_9.py`.
2. Создайте переменные:
 - a. `var_1 = 37`
 - b. `var_2 = 99`
3. Напишите код, который меняет значение переменных местами (`var_1` должен быть равен 99 и `var_2` — 37).
4. Выведите обновленные переменные на экран.

▼ Подсказка (открыть в крайнем случае)

Можно создавать новые переменные.

Задание 10. Банковское приложение*

1. Создайте файл `lesson_2_task_10.py`.
2. Дано: пользователь делает вклад в размере `X` рублей сроком на `Y` лет под 10% годовых (каждый год размер его вклада увеличивается на 10%, эти деньги прибавляются к сумме вклада, и на них в следующем году тоже будут проценты).
3. Задача: написать функцию `bank`, принимающую аргументы `X` и `Y` и возвращающую сумму, которая будет на счету пользователя спустя `Y` лет.

**Это задача жизненная, у нее может быть много способов решения, единственно верного решения нет. Обсудите с коллегами создание наиболее эффективного алгоритма.*

P. S. Если вы хотите попрактиковаться дополнительно, можно параллельно выполнять задания по этой ссылке: <https://code-basics.com/ru/languages/python>.