

# Задание 1. Основы Python

## Описание задачи 6

Курс по методам машинного обучения, 2022-2023, Мишустина Маргарита

## 1 Характеристики задания

- **Длительность:** 1 неделя
- **Юнит-тестирование:** 22 балла; можно сдавать после дедлайна со штрафом в 40%; Публичная часть
- **Почта:** ml.cmc@mail.ru
- **Темы для писем на почту:** BMK.ML[Задание 1][unit-tests]

## 2 Описание задания

Прежде, чем приступить к заданию

- Убедитесь, пожалуйста, что у вас стоит виртуальное окружение с Python3 и установленными библиотеками нужных версий [отсюда](#), а также стоит jupyter-lab или jupyter-notebook.
- Убедитесь, что у вас есть доступ к Google Colab

### Внимание!

К данному заданию приложен очень информативный ноутбук (ipynb), содержащий в себе большое количество информации про python.

- Скачать его можно вот [отсюда по ссылке](#)
- Или в проверяющей системе, нажав на «Дополнительные файлы для решения».

Кроме того, в данном задании 7 задач; задачи 1-5, 6 и 7 находятся в *отдельных вкладках*. Их условия будут продублированы в данном файле.

### Как можно локально протестировать ваше решение

Для выполнения задания нужно скачать из *проверяющей системы*:

1. Скрипт для тестирования `run.py` (вкладка «скрипт для тестирования»)
2. Архив с тестами (вкладка «публичные тесты»)
3. Архив с шаблонами решения (вкладка «шаблон решения»)

### Внимание!

Для каждого из заданий (1-5, 6, 7) эти файлы **разные!** Кладите файлы из разных заданий в разные папки, чтобы их не перепутать!

### Для 6го задания

Зайдите в задание Введение в Python.6. Разархивируйте архив с тестами в папку `python_intro_public_test` и рядом с этой папкой положите тестовый скрипт `run.py`. Также Разархивируйте архив с шаблонами решения и положите их в так же рядом с `run.py`. В файлах с шаблонами решений необходимо написать необходимые функции, а затем, после успешного локального тестирования, сдать их в проверяющую систему. Для запуска тестов вам понадобится библиотека `pytest`.

Таким образом, в произвольной директории (назовем ее `Python_task6`) должны находиться файлы `run.py` и решения задач `task6.py` и тд. Там же должна быть создана директория `python_intro_public_test`,

содержимое которой должно представлять собой распакованный архив с публичными тестами, то есть содержать 01\_unittest\_task6\_input и тд:

```
> Python_task6
> run.py
> task6.py
> python_intro_public_test
> 01_unittest_task6_input
```

Как запускать тесты — будет расписано ниже в каждой задаче.

**Замечание:** Запрещается пользоваться библиотеками, импорт которых не объявлен в файле с шаблонами функций.

**Замечание:** Задания, в которых есть решения, содержащие в каком-либо виде взлом тестов, дополнительные импорты и прочие нечестные приемы, будут автоматически оценены в 0 баллов без права пересдачи задания.

### 3 Задача 6 (3 балла)

**Формулировка:** Когда студент прочитал учебник по линейной алгебре и аналитической геометрии, ему стало интересно, сколько слов и в каком количестве встречается в этой книге.

Требуется написать функцию `check(s, filename)`, которая принимает на вход строку – последовательность слов, разделенных пробелом и имя файла; слова состоят из строчных и прописных букв латинского алфавита, а разделяются пробельными символами (ввод считать корректным). Функция должна вывести в файл для каждого уникального слова в этой строке число его повторений (без учёта регистра) в формате "слово количество" (см. пример вывода). Гарантируется, что строка содержит минимум 1 слово.

Слова выводить нужно по алфавиту, каждое уникальное слово должно выводиться только один раз.

**Пояснение:** Решение должно содержать файл `task6.py` с функцией `check(s, filename)`.

**Пример работы:** В многострочном комментарии после вызова функции указано содержимое файла `file.txt`.

```
from task6 import check

check("a aa abC aa ac abc bcd a", "file.txt")
"""
a 2
aa 2
abc 2
ac 1
bcd 1
"""
```

**Запуск тестов:** Следующую команду нужно ввести в командной строке из директории `Python_task6`, содержащей файл `run.py` и `task6.py`.

```
$ python run.py python_intro_public_test
```