

## Домашнее задание

Дисциплина	Программирование на Python
Тема	Управляющие конструкции и коллекции
Форма проверки	<b>Самопроверка. Студент выполняет задание и самостоятельно проверяет его с помощью чек-листа</b>
Имя преподавателя	Константин Башевой
Время выполнения	2 часа
Цель задания	Приобрести базовые навыки работы с основными типами данных в Python
Инструменты для выполнения ДЗ	Jupyter Notebook или Google Colab
Правила приёма работы	Прикрепите ссылку на выполненное задание в Google Colab или GitHub, если вы использовали Jupyter Notebook. Важно: убедитесь в том, что по ссылке есть доступ в Google Colab (иногда в Colab нет доступа для другого логина)
Критерии оценки	<b>Задание считается выполненным, если:</b> <ul style="list-style-type: none"><li>• прикреплена ссылка на файл с готовым заданием;</li><li>• доступ к файлу открыт;</li><li>• создан репозиторий, содержащий файл с кодом и файл README.md с кратким описанием этого кода.</li></ul> <b>Задание не выполнено, если:</b> <ul style="list-style-type: none"><li>• файл с заданием не прикреплён или отсутствует доступ по ссылке;</li><li>• репозиторий не создан или в нём нет файла с кодом и файла README.md с кратким описанием этого кода.</li></ul>
Дедлайн	Срок сдачи — 7 дней после окончания вебинара

## Описание задания

Перед выполнением задания установите Jupyter Notebook или используйте Google Colab.

## **Задание 1**

Напишите код на Python в среде Jupyter Notebook или Google Colab для решения следующей задачи.

Дана переменная, в которой хранится слово из латинских букв. Напишите код, который выводит на экран:

- среднюю букву, если число букв в слове нечётное;
- две средних буквы, если число букв чётное.

Пример работы программы:

1. word ='test' Результат: es
2. word = 'testing' Результат: t

## **Задание 2.**

Напишите код на Python в среде Jupyter Notebook или Google Colab для решения следующей задачи.

Вы делаете MVP (минимально жизнеспособный продукт) dating-сервиса. У вас есть список юношей и девушек.

Выдвигаем гипотезу: лучшие рекомендации получатся, если просто отсортировать имена по алфавиту и познакомить людей с одинаковыми индексами после сортировки. Но вы не будете никого знакомить, если кто-то может остаться без пары.

Примеры работы программы:

```
boys = ['Peter', 'Alex', 'John', 'Arthur', 'Richard']
girls = ['Kate', 'Liza', 'Kira', 'Emma', 'Trisha']
```

Результат:

Идеальные пары:

Alex и Emma

Arthur и Kate

John и Kira

Peter и Liza

Richard и Trisha

```
boys = ['Peter', 'Alex', 'John', 'Arthur', 'Richard', 'Michael']
girls = ['Kate', 'Liza', 'Kira', 'Emma', 'Trisha']
```

Результат: Внимание, кто-то может остаться без пары!

## **Чек-лист самопроверки**

- Установлен Jupyter Notebook или используется Google Colab.
- Задание 1.

При вводе слова word = 'test' выдаётся 'es'.

При вводе слова word = 'testing' выдаётся 't'.

- Задание 2.

При следующих входных данных:

boys = ['Peter', 'Alex', 'John', 'Arthur', 'Richard']

girls = ['Kate', 'Liza', 'Kira', 'Emma', 'Trisha']

выводится на экран:

Идеальные пары:

Alex и Emma

Arthur и Kate

John и Kira

Peter и Liza

Richard и Trisha.

При следующих входных данных:

boys = ['Peter', 'Alex', 'John', 'Arthur', 'Richard', 'Michael']

girls = ['Kate', 'Liza', 'Kira', 'Emma', 'Trisha']

выводится на экран:

Внимание, кто-то может остаться без пары!

- На учебной платформе прикреплена ссылка на выполненное задание в Google Colab или GitHub, если вы использовали Jupyter Notebook.