

2024-09-04

Программирование (Python) — оргвопросы

Трек 1: Хочу выучить Питон

1. Посещение занятий
2. Сдача лабораторных работ
3. Выполнение практик
4. Подготовка и сдача итогового задания

Трек 2: Хочу сдать дисциплину, при этом Питон знаю или выучу самостоятельно

1. Работа с преподавателем в режиме консультации
2. Использование Python как основного языка разработки программного продукта (MVP), которым предполагается отчитаться по дисциплине
3. Итоговый проект должен отвечать ряду требований:
 1. должно быть ТЗ (по ЕСПД) или хотя бы функциональные требования в отдельном документе;
 2. должен быть разработан проект с описанием архитектуры, API (если предусмотрено), желательно в какой-либо нотации, например, UML или BPMN
 3. разработка строго с использованием системы контроля версий с майлстоунами каждые 2-4 недели
 4. должны быть написаны автотесты
 5. проект должен быть подготовлен к развертыванию через средства CI/CD (минимум — Docker + Docker Compose)

И итоговый проект, и итоговое задание должны быть защищены в конце курса по аналогии с тем, как защищаются заявки на конкурсах поддержки от ФСИ и аналогичных фондов

Подготовка

1. Установить Python версии 3.11 или 3.12:
 1. через менеджер пакетов дистрибутива на Линуксе
 2. через winget или EXE на Windows (не забыть галочку про PATH !)
 3. через brew на macOS
 4. ссылка: [Download Python | Python.org](https://www.python.org/downloads/)
2. через pip поставить Jupyter Notebook: `pip install jupyter notebook`
3. научиться создавать и активировать виртуальные среды:
`python -m venv ./venv`, см. [venv — Creation of virtual environments — Python 3.12.5 documentation](https://docs.python.org/3.12.5/using/venv.html)

Шпаргалки по языку

[python_basics.pdf](#)

[Comprehensive Python Cheatsheet](#)

[Python Cheatsheet - Python Cheatsheet](#)

Примерная структура курса

1. Введение в Python. История языка, экосистемы и настройка средств разработки

2. Императивное программирование. Типы данных и коллекции
3. Принципы и правила оформления кода. PEP. Видимость переменных
4. Объектно-ориентированное программирование. Имитация функции
5. Декларативное программирование. Включение (comprehensions). Функции высшего порядка. Частичное применение функций. Анонимные функции.
6. Параллельное программирование. Multiprocessing. Multithreading и GIL. Событийно-ориентированное программирование.
7. Научные вычисления. Машинное обучение (введение). Робототехника. Анализ данных.
8. Веб-программирование. Фреймворки.



Практика №1

1. Сгенерировать, используя модуль псевдослучайных чисел `random`, или ввести с клавиатуры список целых чисел.

Вывести в консоль, затем перевернуть его и снова вывести в консоль.

2. Сгенерировать, используя модуль псевдослучайных чисел `random`, или ввести с клавиатуры два списка целых чисел.

Вывести их в консоль. Создать новый пустой список.

Добавить в него все четные (по индексу) элементы первого списка и все нечетные (по индексу) элементы второго списка. Вывести третий список в консоль.

3. Сгенерировать, используя модуль псевдослучайных чисел `random`, или ввести с клавиатуры список произвольных элементов (целые числа, числа с плавающей точкой, строки).

Вывести в консоль. Убрать из него все дубликаты через приведение типов. Вывести в консоль.

4. Сгенерировать, используя модуль псевдослучайных чисел `random`, или ввести с клавиатуры словарь, где ключом является строка, значением — целое число или число с плавающей точкой. Вывести в консоль. Для всех уникальных значений создать кортеж, где первым элементом будет значение, вторым — список связанных с ним ключей. Собрать эти кортежи в список, вывести его в консоль.

5. Сгенерировать, используя модуль псевдослучайных чисел `random`, или ввести с клавиатуры два словаря, где ключом является строка, значением — целое число или число с плавающей точкой. Вывести в консоль. Найти пересечения множеств значений словарей. Создать новый словарь,

содержащий только те пары ключ-значение, значения из которых входят в пересечение. Вывести в консоль.

Лабораторная работа №1: Базовые типы данных и коллекции Python

1. Установить интерпретатор Python версии 3.11+ (<https://python.org>)
2. Изучить установку внешних пакетов через `pip` (<https://pypi.org>)
3. Установить Jupyter Notebook (если используете VS Code, то он нужен для работы интерактивной оболочки)
4. Изучить основные типы данных, операнды, структуры данных (списки, словари, кортежи, множества)
5. Описать структуру Московского зоопарка при помощи встроенных типов и структур данных.
6. Написать код, который возвращает соседей манула Тимофея (т.е. просто соседние с манулами вольеры)