# **Лабораторная работа № 1**

*Начало работы с Python*

Python – это интерпретируемый язык программирования высокого уровня, созданный Гвидо ван Россумом и впервые выпущенный в 1991 году.

Python поддерживает структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное программирование. Основные архитектурные черты — динамическая типизация, автоматическое управление памятью, механизм обработки исключений, поддержка многопоточных вычислений, высокоуровневые структуры данных.

Python может служить скриптовым языком для создания веб-приложений. Веб-фреймворки, такие как Django, Pylons, Pyramid, TurboGears, web2py, Tornado, Flask, Bottle и Zope, помогают разработчикам в разработке и сопровождении сложных приложений.

Python — активно развивающийся язык программирования, новые версии с добавлением/изменением языковых свойств выходят примерно раз в два с половиной года.

Установка Python 3.7.X (неважно, какая цифра в версии будет стоять третьей):

Необходимо скачать с официального сайта https://www.python.org/downloads/ последнюю версию интерпретатора Python для нужной операционной системы, затем запустить установочный файл.

Узнать точную установленную версию Python3 можно введя данную команду в командной строке: python3 -V

Создайте директорию для лабораторных работ (в названии директории не используйте кириллицу, т.к. при выполнении команд может произойти ошибка). В созданной папке создайте файл «helloworld.py». Используя редактор кода IDLE или любой другой добавьте в созданный файл следующий код: print("Hello world!")

Для запуска программы необходимо открыть командную строку (cmd – в Windows, Terminal – в macOS), перейти в консоли в папку с лабораторными работами с помощью команды cd + адрес директории. Например: cd Users/admin/Desktop/labs\_python. После перехода в нужную директорию, нужно запустить скрипт с помощью команды: python3 helloworld.py

*Пример:* 

Рисунок 1. Пример работающего скрипта

Таким образом, первая программа на Python3 запущена.

Затем в той же директории создайте файл «mygroup.py», в котором будут находится данные о студентах вашей группы (достаточно 5-7 человек) в формате списка. Обратите внимание, что весь список сохраняется в одну переменную (например, groupmates), для каждого студента должны быть заполнены поля: имя, фамилия, список экзаменов, оценки.

*Пример:*

groupmates = [

{

"name": "Александр",

"surname": "Иванов",

"exams": ["Информатика", "ЭЭиС", "Web"],

"marks": [4, 3, 5]

},

{

"name": "Иван",

"surname": "Петров",

"exams": ["История", "АиГ", "КТП"],

"marks": [4, 4, 4]

},

{

"name": "Кирилл",

"surname": "Смирнов",

"exams": ["Философия", "ИС", "КТП"],

"marks": [5, 5, 5]

}

]

**Задание.** Вам необходимо написать функцию фильтрации студентов по средней оценке, так, чтобы на экран выводился список студентов, средний балл которых выше заданного. Средний балл, по которому будет проводиться фильтрация, вводится пользователем с клавиатуры.

*Пример функции вывода списка всех студентов:*

def print\_students(students):

print(u"Имя".ljust(15), u"Фамилия".ljust(10), u"Экзамены".ljust(30), u"Оценки".ljust(20))

for student in students:

print(student["name"].ljust(15), student["surname"].ljust(10), str(student["exams"]).ljust(30), str(student["marks"]).ljust(20))

print\_students(groupmates)

Функция в Python – это объект, принимающий аргументы и возвращающий значение. Разберем функцию вывода подробнее.

def print\_students(students): – объявление функции.

def – стандартная инструкция для объявлении функции.

print\_students – название функции (может быть любым).

(students) – принимаемое значение.

print (u"Имя".ljust(15), u"Фамилия".ljust(10), u"Экзамены".ljust(30), u"Оценки".ljust(20)) – вывод на экран «шапки» таблицы.

u"Имя" – префикс u обозначает, что данная строка будет выведена в кодировке Unicode.

.ljust(15) – метод строки, который добавляет нужное количество пробелов, чтобы длина строки стала равной передаваемому параметру. Используется для выравнивания строк.

for student in students:

print(student["name"].ljust(15), student["surname"].ljust(10), str(student["exams"]).ljust(30), str(student["marks"]).ljust(20)) – цикл, в котором построчно выводится информация о каждом студенте.

print\_students(groupmates) – вызов функции. В качестве передаваемого значения – список всех студентов (groupmates).

*Начало работы с Django*

Django – это свободный веб-фреймворк с открытым исходным кодом на основе Python, который следует архитектурному шаблону MVC (Model-View-Controller: Модель-Представление-Контроллер).

Установка Django:

Необходимо перейти на официальный сайт фреймворка Django https://www.djangoproject.com/download/, где будет написана команда для установки с указанием последней версии, например, на момент написания учебного пособия актуальной версией Django была версия 2.1.7. Пример установки фреймворка Django с помощью pip (системы управления пакетами): pip3 install Django==2.1.7

***Примечание:*** *команда pip3 используется только для операционных систем macOS и Linux, для Windows используется команда pip.*

После установки фреймворка приступим к созданию проекта. Для этого с помощью командной строки необходимо перейти в директорию, в которой находятся ваши лабораторные работы, и выполнить команду:

**django-admin.py startproject project\_name**

project\_name – имя вашего проекта

После выполнения данной команды в выбранной директории создается папка **project\_name**, в которой содержится:

* Файл manage.py – файл для командной строки (терминала), который позволяет взаимодействовать с проектом с помощью различных методов.
* Файл \_\_init\_\_.py – файл необходим для того, чтобы Python рассматривал данный каталог как пакет, т.е., как группу модулей. Это пустой файл.
* Файл settings.py – в нем содержатся настройки для текущего проекта.
* Файл urls.py – описания URL для текущего проекта. «Оглавление» сайта.
* Файл wsgi.py – файл для настроек текущего проекта. Не изменяется.

Для того, чтобы проверить, правильно ли был установлен фреймворк Django, необходимо запустить локальный web-сервер разработки. Для запуска сервера перейдите в директорию project\_name, в которой находится файл manage.py и выполните команду, которая запускает локальный сервер на порту 8000:

**Перейти в project\_name в консоли**

**python manage.py runserver** или python3 manage.py runserver

Если при установке не возникло ошибок, то в консоли появится следующее:

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2. Сообщение о правильном запуске сервера

Затем перейдите по адресу: <http://127.0.0.1:8000/> Если сервер удачно запущен, то будет выведено сообщение: The install worked successfully! Congratulations!

Для того, чтобы попасть в административный интерфейс Django, необходимо обладать правами суперпользователя. Для этого в свою очередь необходимо создать нужные таблицы в базе данных, в которой будут храниться все данные о пользователях.

**Задание.**

В файле settings.py измените параметр базы данных. Переменной DATABASES есть внутренний словарь с ключами ENGINE и NAME. Для параметра NAME установите следующее значение:

'NAME': os.path.join(BASE\_DIR, 'db\_project\_name')

**В нём же указать import os перед патчем**

– данный параметр является названием базы данных, так что он может быть любым.

Чтобы создать таблицы базы данных, необходимо перейти в директорию project\_name и через командную строку выполнить команду:

**python3 manage.py migrate**

Затем создаем суперпользователя (при создании необходимо ввести логин, email и пароль пользователя):

**python3 manage.py createsuperuser**

Теперь необходимо запустить сервер, если он не был запущен до этого, командой:

**python3 manage.py runserver**

и в браузере пройти по адресу http://127.0.0.1:8000/admin/.

После этого должно появиться окно входа, куда необходимо ввести данные, которые вы указали при создании суперпользователя. Когда вы успешно войдете в аккаунт суперюзера, перед вами откроется главная страница административной панели, через которую вы можете управлять вашими приложениями, редактируя существующие записи в базе данных или генерируя новые.

* Изучите интерфейс административного приложения django;
* Через интерфейс административного приложения создайте нового пользователя с правами суперпользователя;
* Через интерфейс административного приложения создайте нового пользователя без прав суперпользователя;
* Через интерфейс административного приложения «забаньте» одного из пользователей (сделайте пользователя «неактивным»);
* Загрузите ваш проект на любой гит-репозиторий (GitHub, GitLab, Google Code, Bitbucket и т.п.).