# **Main Terminal**

**ls** или **dir** выводит все файлы которые лежат в текущей директории

**cd ./my\_folder** идет на одну указаную директорию вперед

**cd ..** идет на одну директорию назад

**mkdir my\_folder** создание папки с названием

**copy con my\_file.txt -> Ctrl+Z -> Enter** создание файла в текущей директории (также echo)

**pwd** показывает путь к текущему каталогу

**D:** переход на другой диск

**.\my\_name.exe** выполнение файла

**clear** или **cls** очистка терминала

**exit** выход из терминала

# **Python**

**python -V** выводит версию python

**pip** пакетный менеджер пайтон

**pip list** выводит список установленых пакетов(модулей)

**pip freeze** выводить только сторонние пакеты, которые установил сам програмист

**pip list > my\_file.txt** создаем файл, куда и выводим результат выполнения команды

**pip install --upgrade pip** обновление пакетного менеджера пайтон

**rm my\_file.txt** или **rm my\_folder** удаление файлов или папок

**pip install name\_module** установка модуля

**pip uninstall name\_module** удаление модуля

**python** запуск интерпритатора python

**python my\_file.py** запус указаного файла в интерпритаторе

**python -i my\_file.py** запуск файла в фоновом(интерактивном режиме)

**exit( )** выход из интерпритатора python

# **Django**

**python -m venv my\_name** создание виртуального окружения для проекта Django и дача названия

**rm my\_name** удаление виртуального окружения

**.\my\_name\Scripts\activate** запуск виртуального окружения нашого проекта

**deactivate** выход из виртуального окружения

**pip install -r .\requirements.txt** установка всех пакетов из етого файла

**pip install django** установка самого фрейморка django

**django-admin** выводит все возможные команды django

**django-admin startproject my\_project** создание проекта

**python manage.py runserver** запуск web-сервера в качетсве нашего компьютера . Проверка через http://127.0.0.1:8000/ . Выйти Ctrl + C

**python manage.py runserver** **4000** запуск сервера с указанием порта

**python manage.py startapp my\_app** создания приложения

# **Project Files**

**requirements.txt** содержит внутри пакеты(модули), которые нужно установить для работы с проектом

**manage.py** отвечает за запуск web-сервера

**setting.py -> INSTALLED\_APPS = [ ‘my\_app.apps.NameConfig’ ]** хранить подключеные приложения. После создания приложений нужно сюда добавлять

**urls.py** -> **urlpatterns = [ ]** указывають url запросы от пользователя и что они делают

# **Applications Files**

**migrations** хранятса переносы баз данных

**\_\_init\_\_.py** инициализация самого приложения

**admin.py** настройка админ панели нашего сайта

**apps.py** настройка текущего приложения

**models.py** хранить ORM модели для взаемодействия с базами данных

**tests.py** хранит тестирующие процедуры

**views.py** хранит педставление, которые отвечают пользователю