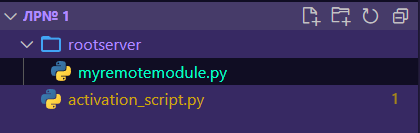
**Лабораторная работа №1**

1.Создаём файлы «myremotemodule.py» и «activation\_script.py», и получаем данную директорию:



Содержимое файлов:

«myremotemodule.py»:

def myfoo():

    author = "Artem Gnevnov"

    print(f"{author}'s module is imported")

«activation\_script.py»:

import re

import sys

from urllib.request import urlopen

from importlib.abc import PathEntryFinder

from importlib.util import spec\_from\_loader

def url\_hook(some\_str):

    if not some\_str.startswith(("http", "https")):

        raise ImportError

    with urlopen(some\_str) as page: *# requests.get()*

        data = page.read().decode("utf-8")

    filenames = re.findall("[a-zA-Z\_][a-zA-Z0-9\_]\*.py", data)

    modnames = {name[:-3] for name in filenames}

    return URLFinder(some\_str, modnames)

sys.path\_hooks.append(url\_hook)

print(sys.path\_hooks)

class URLLoader:

    def create\_module(self, target):

        return None

    def exec\_module(self, module):

        with urlopen(module.\_\_spec\_\_.origin) as page:

            source = page.read()

        code = compile(source, module.\_\_spec\_\_.origin, mode="exec")

        exec(code, module.\_\_dict\_\_)

class URLFinder(PathEntryFinder):

    def \_\_init\_\_(self, url, available):

*self*.url = url

*self*.available = available

    def find\_spec(self, name, target=None):

        if name in *self*.available:

            origin = "{}/{}.py".format(*self*.url, name)

            loader = URLLoader()

            return spec\_from\_loader(name, loader, origin=origin)

        else:

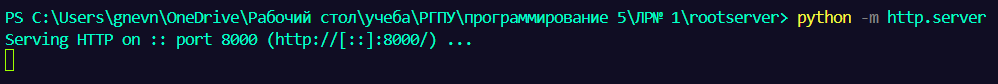
            return None

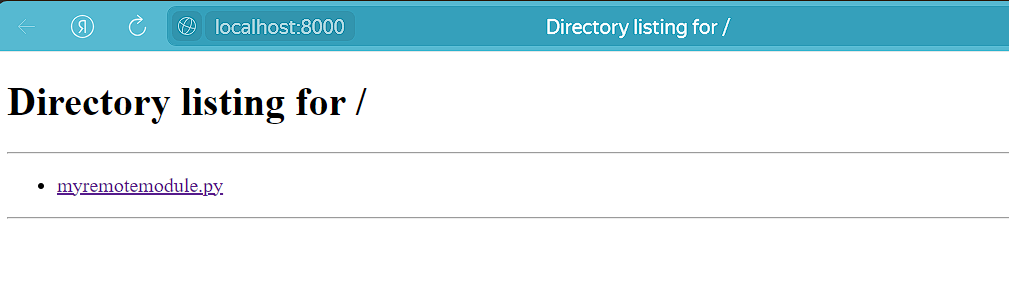
sys.path.append("http://localhost:8000")

import myremotemodule

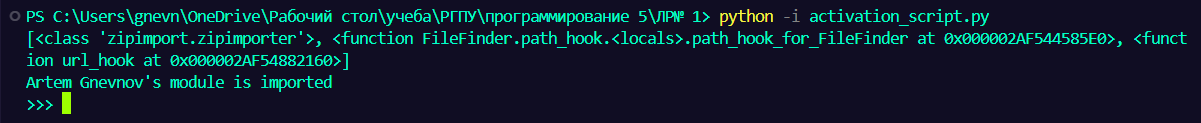
myremotemodule.myfoo()

2.Запускаем сервер командой «python -m http.server»:





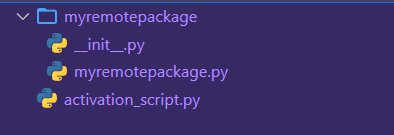
3.Для импорта выполняем команду «python -i activation\_script.py»:



python -i activation\_script.py

**Задание \*\*\***

1. Меняем нашу рабочую директорию, чтобы она выглядела так:



Обновляем код, чтобы импортило пакет, а не модуль:

Файл «\_\_init\_\_.py»:

from .myremotepackage import myfoo

Файл «myremotepackage.py»:

def myfoo():

    author = "Artem Gnevnov"

    print(f"{author}'s package is imported")

Файл «activation\_script.py»:

import re

import sys

from urllib.request import urlopen

from importlib.abc import PathEntryFinder

from importlib.util import spec\_from\_loader

def url\_hook(some\_str):

    if not some\_str.startswith(("http", "https")):

        raise ImportError

    with urlopen(some\_str) as page:

        data = page.read().decode("utf-8")

    filenames = re.findall(r"[a-zA-Z\_][a-zA-Z0-9\_]\*(/|\.py)", data)

    modnames = {name.replace('/', '')[:-3] for name in filenames if name.endswith('.py')}

    return URLFinder(some\_str, modnames)

sys.path\_hooks.append(url\_hook)

print(sys.path\_hooks)

class URLLoader:

    def create\_module(self, target):

        return None

    def exec\_module(self, module):

        with urlopen(module.\_\_spec\_\_.origin) as page:

            source = page.read()

        code = compile(source, module.\_\_spec\_\_.origin, mode="exec")

        exec(code, module.\_\_dict\_\_)

class URLFinder(PathEntryFinder):

    def \_\_init\_\_(self, url, available):

*self*.url = url

*self*.available = available

    def find\_spec(self, name, target=None):

        if name in *self*.available:

            origin = f"{*self*.url}/{name.replace('.', '/')}/\_\_init\_\_.py"

            loader = URLLoader()

            return spec\_from\_loader(name, loader, origin=origin)

        else:

            return None

sys.path.append("http://localhost:8000")

import myremotepackage

myremotepackage.myfoo()

2.Запускаем сервер командой «python -m http.server»:



3. Импортим командой «python -i activation\_script.py»:

