# find\_ok\_url

<https://chatbot.theb.ai/#/chat/1684869237763>

Ты программист разработчик. Напиши код на Python.

Программа должна проверять ссылки на страницы сайта и находить рабочие ссылки.

Схема ссылки следующая: «https://lk.neural-university.ru/learning-program-v2/320370»

Где постоянная часть это: «https://lk.neural-university.ru/learning-program-v2/», а переменная часть: «320370».

Программа должна в цикле увеличить на единицу значение переменной части, сформировать ссылку и проверить ее, например: следующая переменная часть, увеличенная на единицу это «320371», новая ссылка это «https://lk.neural-university.ru/learning-program-v2/320371», при выполнение GET запроса по этой новой ссылке получен ответ «OK», значит ссылка рабочая. Выводим на экран рабочую ссылку.

Программа каждую рабочую ссылку выводит на экран.

В конце работы программа выводит список всех найденных рабочих ссылок.

В начале работы программа спрашивает у пользователя значение переменной части для начала и для окончания.

При достижении значения для окончания, программа останавливается и выводит путь к сохраненному файлу со списках всех найденный рабочих ссылок.

Измени код программы. Вместо проверки «response.status\_code == 200» нужно проверять в теле страницы отсутствие текста «error 404»

Ты программист разработчик. Напиши код на Python.

Программа должна проверять ссылки на страницы сайта и находить рабочие ссылки.

Схема ссылки следующая: «https://lk.neural-university.ru/learning-program-v2/320370»

Где постоянная часть это: «https://lk.neural-university.ru/learning-program-v2/», а переменная часть: «320370».

Для аторизации на этом сайте программа должна использовать login\_name = “change\_name” и password «12345».

Программа должна в цикле увеличить на единицу значение переменной части, сформировать ссылку и проверить ее, например: следующая переменная часть, увеличенная на единицу это «320371», новая ссылка это «https://lk.neural-university.ru/learning-program-v2/320371», при выполнение GET запроса по этой новой ссылке получена страница в теле которой нет текста «error 404», это значит ссылка рабочая. Выводим на экран рабочую ссылку.

Программа каждую рабочую ссылку выводит на экран.

В конце работы программа выводит список всех найденных рабочих ссылок.

В начале работы программа спрашивает у пользователя значение переменной части для начала и для окончания.

При достижении значения для окончания, программа останавливается и выводит путь к сохраненному файлу со списках всех найденный рабочих ссылок.

# Olds

Напиши код на Python, код должен выполнять следующий сценарий:

Запросить у пользователя файловый каталог,

В цикле обойти все файлы в указанном файловом каталоге и подкаталогах.

Для каждого файла добавить его содержимое в выходной файл указав к комментарии имя текущего файла.

После окончания цикла вывести содержимое выходного файла.

import os

dir\_path = input("Введите путь до файлового каталога: ")

output\_file\_path = "output.txt"

with open(output\_file\_path, "w") as output\_file:

for subdir, dirs, files in os.walk(dir\_path):

for file in files:

file\_path = os.path.join(subdir, file)

with open(file\_path, "r") as input\_file:

file\_contents = input\_file.read()

output\_file.write(f"=== {file\_path} ===\n")

output\_file.write(file\_contents + "\n")

with open(output\_file\_path, "r") as output\_file:

print(output\_file.read())

Пользователю предлагается ввести путь до файлового каталога. Затем используется функция os.walk для обхода всех файлов в указанном каталоге и подкаталогах. Для каждого файла открывается файловый поток на чтение, считывается содержимое файла, и имя файла добавляется в выходной файл в качестве комментария. После завершения цикла файлов выводится содержимое выходного файла на экран.

Напиши код на Python, код должен выполнять следующий сценарий:

Запросить у пользователя файловый каталог,

В цикле обойти все файлы в указанном файловом каталоге и подкаталогах. Для каждого файла:

Открой текущий файл, запомни его имя и путь, определи кодировку файла.

Сконвертируй в koi8 кодировку содержимое текущего файла и добавить это содержимое в выходной файл указав к комментарии имя и путь текущего файла.

После окончания цикла вывести содержимое выходного файла.

import os

import codecs

def **convert\_file**(input\_file, output\_file):

*# определяем кодировку*

with open(input\_file, 'rb') as f:

raw\_data = f.read()

encoding = 'utf-8'

try:

decoded\_data = raw\_data.decode('utf-8')

except UnicodeDecodeError:

encoding = 'windows-1251'

decoded\_data = raw\_data.decode(encoding)

except:

print(f"Unable to decode {input\_file}")

return False

*# конвертируем в koi8*

koi8\_data = codecs.encode(decoded\_data, 'koi8-r')

*# добавляем в выходной файл с комментарием*

with open(output\_file, 'ab') as f:

f.write(b"# " + input\_file.encode() + b"\n")

f.write(koi8\_data)

return True

def **process\_dir**(input\_dir, output\_file):

for root, dirs, files in os.walk(input\_dir):

for file\_name in files:

input\_file\_path = os.path.join(root, file\_name)

if convert\_file(input\_file\_path, output\_file):

print(f"Processed {input\_file\_path}")

if \_\_name\_\_ == '\_\_main\_\_':

input\_dir = input("Enter input directory: ")

output\_file = "output.koi"

*# создаем пустой выходной файл*

open(output\_file, 'wb').close()

process\_dir(input\_dir, output\_file)

*# выводим содержимое выходного файла*

with open(output\_file, 'rb') as f:

print(f.read().decode('koi8-r'))

<https://ai-talks.streamlit.app/>

Напиши код на Python, код должен выполнять следующий сценарий:

Запросить у пользователя файловый каталог и имя выходного файла

В цикле обойти все файлы в указанном файловом каталоге и подкаталогах. Для каждого файла:

Если текущий файл имеет расширение из списка: .json, .proto, .sql, .xml – тогда пропусти этот файл.

Для иных файлов:

Открой текущий файл, запомни его имя и путь, определи кодировку файла.

Сконвертируй в koi8 кодировку содержимое текущего файла и добавить это содержимое в выходной файл указав к комментарии имя и путь текущего файла.

После окончания цикла закрой выходной файл и сообщи полный путь к нему.

Покрой сгенерированный код комментариями.

. import os

import codecs

# Пользователь указывает файловый каталог и имя выходного файла

directory = input("Введите файловый каталог: ")

output\_file\_name = input("Введите имя выходного файла: ")

# Список расширений файлов, которые нужно пропустить

skip\_extensions = [".json", ".proto", ".sql", ".xml"]

# Открываем выходной файл на запись

with open(output\_file\_name, "w") as output\_file:

# Обходим все файлы в указанном каталоге и подкаталогах

for root, dirs, files in os.walk(directory):

for file in files:

# Получаем полный путь к текущему файлу

file\_path = os.path.join(root, file)

# Если текущий файл имеет расширение из списка skip\_extensions, тогда пропускаем его

if os.path.splitext(file\_path)[1] in skip\_extensions:

continue

# Открываем текущий файл

with codecs.open(file\_path, "r", encoding="utf-8", errors="ignore") as input\_file:

# Получаем содержимое текущего файла

content = input\_file.read()

# Получаем имя текущего файла

file\_name = os.path.basename(file\_path)

# Конвертируем содержимое текущего файла в koi8 кодировку

converted\_content = content.encode("koi8-r", errors="ignore").decode("koi8-r", errors="ignore")

# Добавляем содержимое в выходной файл

output\_file.write(f"# Имя файла: {file\_name}, Путь: {file\_path}\n")

output\_file.write(converted\_content)

output\_file.write("\n\n")

# Выводим полный путь к выходному файлу

print(f"Выходной файл создан по пути: {os.path.abspath(output\_file\_name)}")