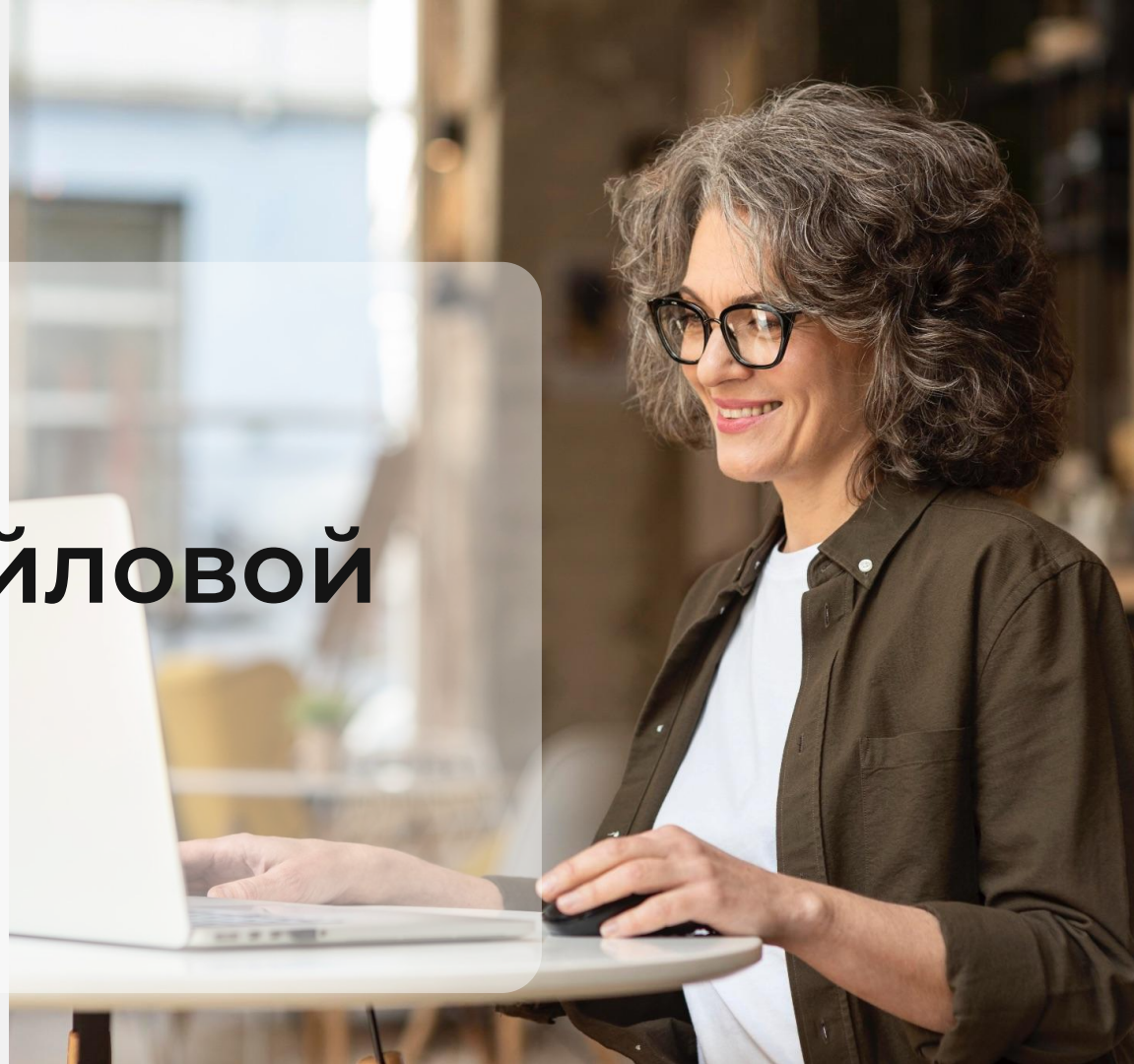
IT CAREER HUB

Python

Работа с файловой системой



Преподаватель

Портрет

Имя Фамилия

Текущая должность

Количество лет опыта

Какой у Вас опыт - ключевые кейсы

Самые яркие проекты

Дополнительная информация по вашему усмотрению

Корпоративный e-mail

Социальные сети (по желанию)

Важно

- Камера должна быть включена на протяжении всего занятия
- В течение занятия вопросы задавать в чате или когда преподаватель спрашивает, есть ли у Вас вопросы
- Вести себя уважительно и этично по отношению к остальным участникам занятия
- Организационные вопросы по обучению решаются с кураторами, а не на тематических занятиях
- Во время занятия будут интерактивные задания, будьте готовы включить камеру или демонстрацию экрана по просьбе преподавателя

Повторение

- Исключения
- Иерархия исключений
- Обработка исключений
- Возбуждение исключений
- Логирование
- Лучшие практики обработки исключений

План занятия

- Введение в файловую систему
- Модуль os. Навигация по файловой системе
- Работа с файлами и директориями
- Работа с путями
- Модуль sys. Работа с командной строкой

ICH IT CAREER HUB

ОСНОВНОЙ БЛОК

IT CAREER HUB

Введение в файловую систему

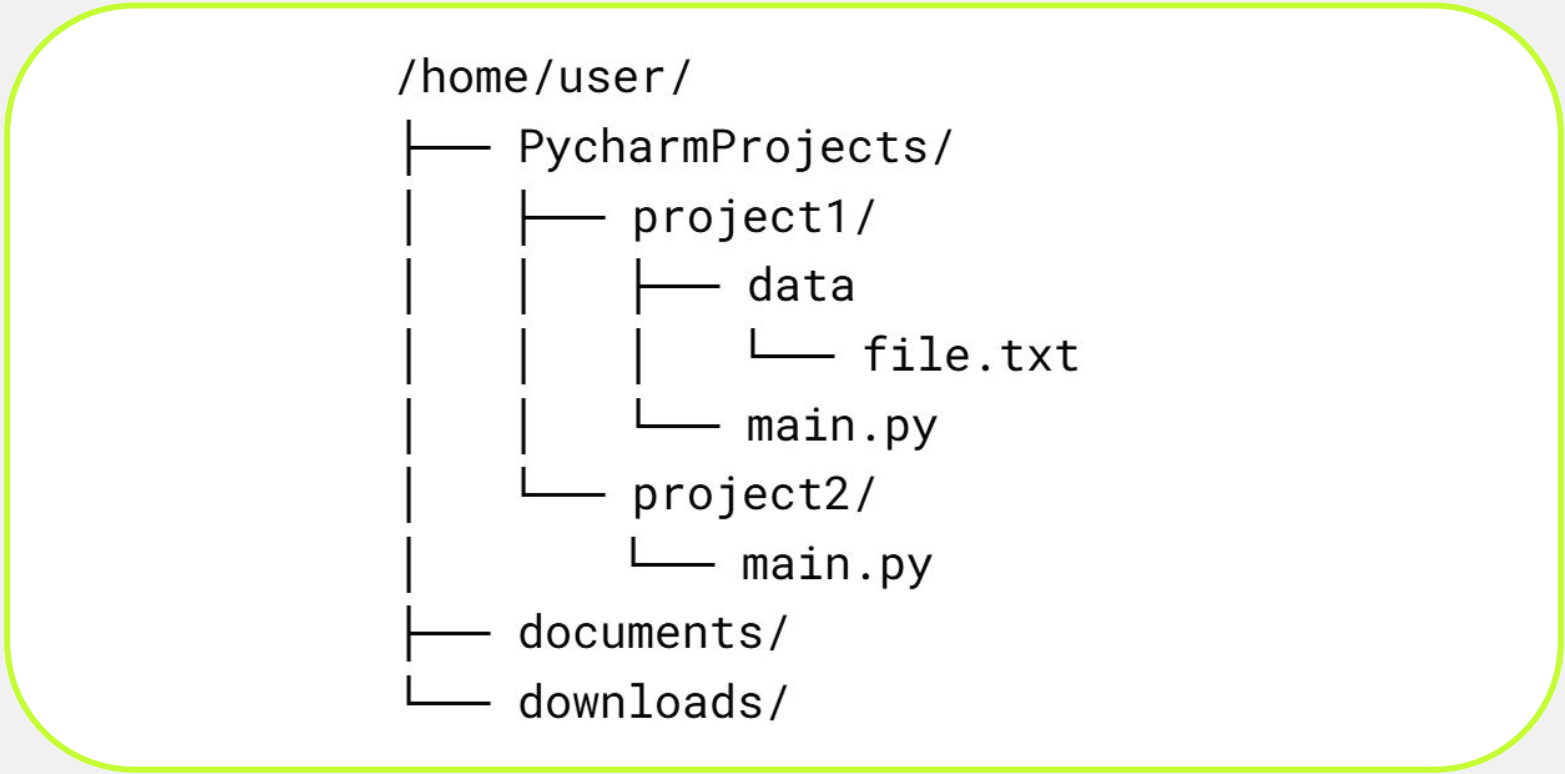
Полезно знать

ICH
IT CAREER HUB

Файловая система

Это способ хранения и организации данных на диске компьютера. Это структура из файлов и папок (директорий), которые можно читать, изменять, удалять или перемещать.

Пример структуры файловой системы



ICH IT CAREER HUB

Абсолютные и относительные пути

Полезно знать

ICH
IT CAREER HUB

Абсолютный путь

Это полный путь от корневой директории до файла или папки. Он указывает точное местоположение объекта в файловой системе.

Пример абсолютного пути

```
#Абсолютный путь в Linux/macOS
absolute_path = "/home/user/documents/file.txt"
```

```
#Абсолютный путь в Windows
absolute_path = "C:\\Users\\User\\Documents\\file.txt"
```

Полезно знать

ICH
IT CAREER HUB

Относительный путь

Это путь к файлу или папке относительно текущей рабочей директории. Не показывает путь до корневой директории

Пример относительного пути

```
#текущая директория /home/user/PycharmProjects/project1/data

./file.txt

#указывает на файл /home/user/PycharmProjects/project1/data/file.txt

../file.txt

#указывает на файл /home/user/PycharmProjects/project1/file.txt
```

Использование

Абсолютный путь

- Когда нужно работать с файлами независимо от текущей директории
- Когда путь к файлу фиксирован (например, конфигурационные файлы)

Относительный путь

- Когда работа с файлами локализована в пределах текущей директории
- Для перемещаемых скриптов

Специальные обозначения для пути

Текущая директория

Знак	Пояснение
.	Указывает на текущую директорию, где находится программа во время выполнения. Используется для построения относительных путей

Windows | MacOS | Linux

```
./file.txt
#файл file.txt в текущей директории
```

ICH
IT CAREER HUB

Специальные обозначения для пути

Родительская директория

Знак	Пояснение
`..`	Указывает на директорию на уровень выше (родительскую)

Windows | MacOS | Linux

```
../file.txt
#файл file.txt в родительской директории

../../file.txt
#файл в директории на два уровня выше
```

Разделитель в пути

Специальные обозначения для пути

Знак	Пояснение
/	В Linux и macOS используется для указания вложенных директорий

MacOS Linux

```
/home/user/file.txt
#абсолютный путь
```

Специальные обозначения для пути

Разделитель в пути

Знак	Пояснение
\\	Только в Windows

Windows

```
C:\\Users\\User\\Documents\\file.txt
#абсолютный путь
```

ICH
IT CAREER HUB

Специальные обозначения для пути

Домашняя директория

Знак	Пояснение
~	Указывает на домашнюю директорию пользователя (Linux и macOS)

MacOS Linux

```
~/documents/file.txt
#путь к файлу в директории documents домашней папки
```

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

Модуль os. Навигация по файловой системе

Модуль os

Модуль os предоставляет инструменты для работы с операционной системой, включая файловую систему

```
#Чтобы использовать модуль, нужно его импортировать
import os
```

Полезно знать

IT CAREER HUB

os.getcwd()

Это метод, который используется для определения текущей рабочей директории

Пример получения названия текущей директории

```python
import os

# Получаем текущую рабочую директорию
current_dir = os.getcwd()
print(f"Текущая директория: {current_dir}")
```

Полезно знать

IT CAREER HUB

os.chdir(path)

Это метод, который используется для перехода в другую директорию

Пример изменения текущей директории

```
import os

# Переход в директорию с использованием относительного пути
os.chdir("./subdirectory")  # Ошибка, если директории нет!
print(f"Текущая директория: {os.getcwd()}")

# Переход в родительскую директорию
os.chdir("..")
print(f"Текущая директория после перехода: {os.getcwd()}")

# Переход в директорию с использованием абсолютного пути
os.chdir("/home/user/documents")  # Замените путь на существующий
print(f"Текущая директория: {os.getcwd()}")
```

IT CAREER HUB

Работа с файлами и директориями

Проверка существования объекта

Для проверки существования файла или директории используется функция `os.path.exists(path)`

Пример проверки существования файла и директории

```python
import os

# Проверка существования файла
file_path = "example.txt"
# или
# file_path = "example_folder/example.txt"
if os.path.exists(file_path):
    print(f"Файл '{file_path}' существует.")
else:
    print(f"Файл '{file_path}' не найден.")

# Проверка существования директории
directory_path = "example_folder"
# или
# directory_path = "parent_folder/child_folder"
if os.path.exists(directory_path):
    print(f"Директория '{directory_path}' существует.")
else:
    print(f"Директория '{directory_path}' не найдена.")
```

Полезно знать

IT CAREER HUB

os.listdir(path)

Это метод, который возвращает список файлов и папок в указанной директории

Пример получения содержимого директории

```
import os

# Список содержимого текущей директории
contents = os.listdir(".")
print("Содержимое текущей директории:", contents)

# Список содержимого указанной директории
specific_dir = "parent_folder"
if os.path.exists(specific_dir):
                print(f"Содержимое    директории    '{specific_dir}':",
os.listdir(specific_dir))
```

Создание новой директории

os.mkdir(path)

- Создаёт одну директорию
- Вызывает ошибку, если директория уже существует

os.makedirs(path)

- Создаёт директорию вместе с родительскими, если их ещё нет
- Вызывает ошибку, если директория уже существует, но с использованием параметра `exist_ok=True` ошибка не возникает

Пример создания директории и вложенных директорий

```
import os

dir_path = "example_folder"
if not os.path.exists(dir_path):
    # Создание одной директории
    os.mkdir("example_folder")
    print(f"Директория '{dir_path}' создана.")
else:
    print(f"Директория '{dir_path}' существует.")

# Создание вложенных директорий
os.makedirs("parent_folder/child_folder", exist_ok=True)
print("Вложенные директории 'parent_folder/child_folder' созданы.")
```

Полезно знать

IT CAREER HUB

open(path, mode)

Это функция, которая создает файл.
Она не является частью модуля os

Пример создания файла

```
# Создание нового пустого файла
file_name = "example.txt"
open(file_name, "w").close()
print(f"Файл '{file_name}' создан или перезаписан.")
```

Полезно знать

IT CAREER HUB

os.path.isfile(path)

Это метод, который используется для проверки, является ли указанный путь файлом

Пример определения типа объекта: файл или нет

```python
import os


file_path = "example.txt"
# Проверяем, существует ли объект
if os.path.exists(file_path):
    # Проверяем, является ли файлом
    if os.path.isfile(file_path):
        print(f"'{file_path}' существует и это файл.")
    else:
        print(f"'{file_path}' существует, но это не файл.")
else:
    print(f"'{file_path}' не существует.")
```

Полезно знать

IT CAREER HUB

os.path.isdir

Это метод, который позволяет проверить, является ли путь директорией

Пример определения типа объекта: файл или директория

```python
import os

path = "example.txt"
# path = "parent_folder"

# Проверяем, является ли файлом
if os.path.isfile(path):
    print(f"'{path}' существует и это файл.")
# Проверяем, является ли директорией
elif os.path.isdir(path):
    print(f"'{path}' существует и это директория.")
```

Полезно знать

IT CAREER HUB

os.rename(src, dst)

Эта функция используется для переименования файлов или директорий

Пример переименования

```python
# Переименование файла
old_file_name = "example.txt"
new_file_name = "renamed.txt"
if os.path.exists(old_file_name):
    os.rename(old_file_name, new_file_name)
    print(f"Файл переименован в '{new_file_name}'.")
else:
    print(f"Файл '{old_file_name}' не найден.")

# Переименование директории
old_dir_name = "example_folder"
new_dir_name = "renamed_folder"
if os.path.exists(old_dir_name):
    os.rename(old_dir_name, new_dir_name)
    print(f"Директория переименована в '{new_dir_name}'.")
else:
    print(f"Директория '{old_dir_name}' не найдена.")
```

Полезно знать

ICH
IT CAREER HUB

os.remove(path), os.rmdir(path)

Для удаления файлов используется функция `os.remove(path)`, а для удаления пустых директорий – `os.rmdir(path)`

Пример удаления

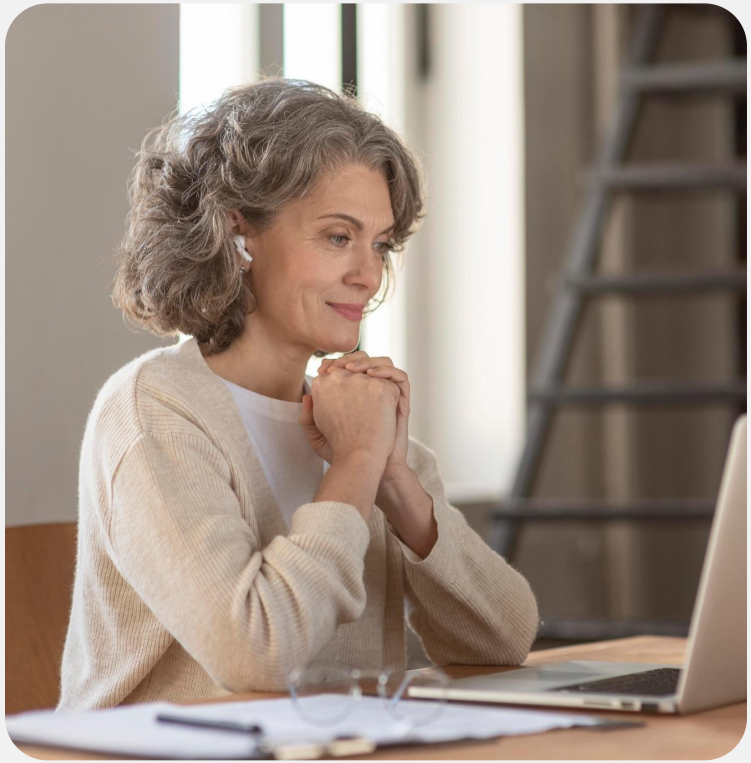
```
# Удаление файла
file_to_delete = "renamed.txt"
if os.path.exists(file_to_delete):
    os.remove(file_to_delete)
    print(f"Файл '{file_to_delete}' удалён.")
else:
    print(f"Файл '{file_to_delete}' не найден, удаление невозможно.")

# Удаление пустой директории
empty_dir_to_delete = "renamed_folder"
if os.path.exists(empty_dir_to_delete):
    os.rmdir(empty_dir_to_delete)
    print(f"Пустая директория '{empty_dir_to_delete}' удалена.")
else:
    print(f"Директория '{empty_dir_to_delete}' не найдена или не пуста.")
```

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

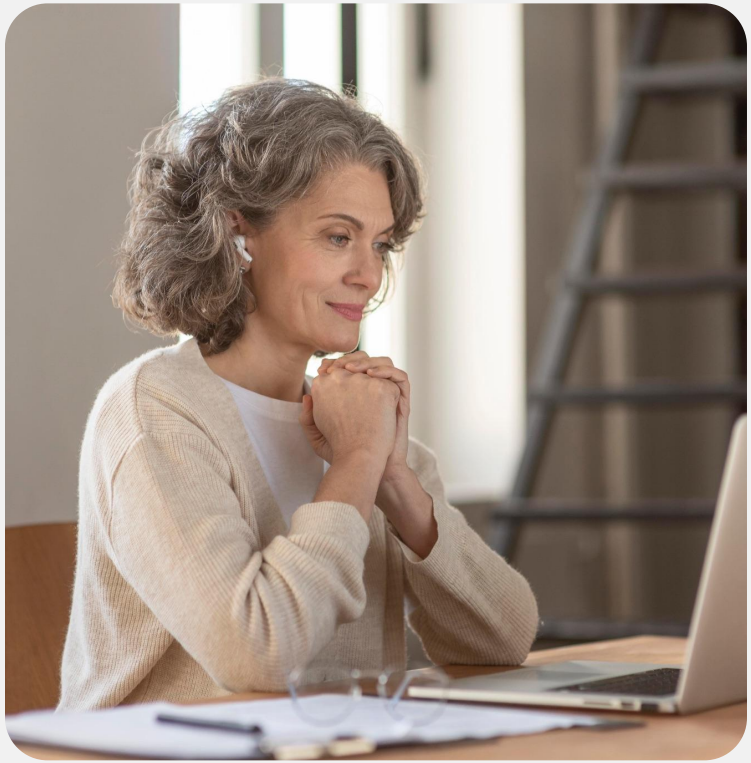
ЗАДАНИЯ



Выберите верный вариант ответа

Что делает os.getcwd()?

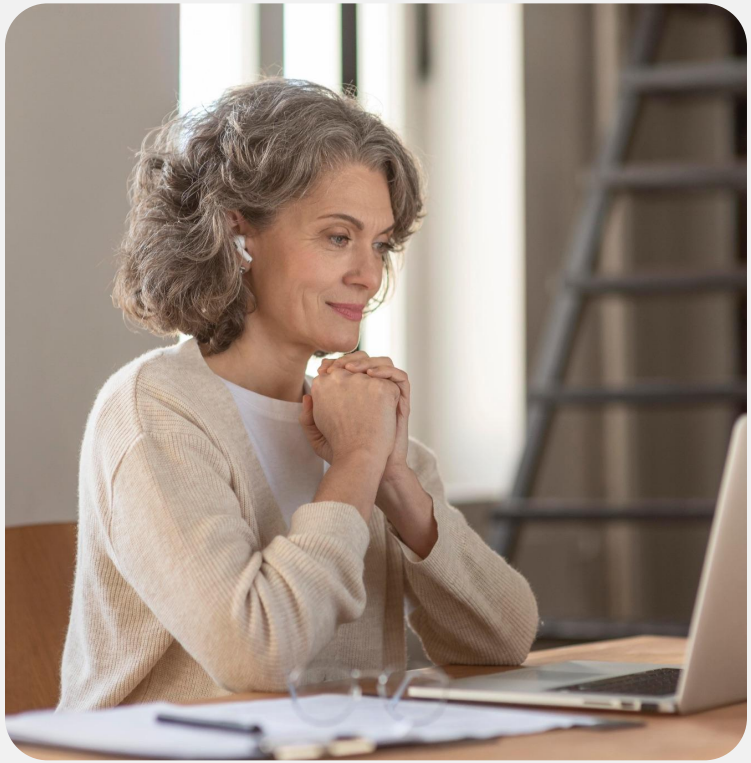
a. Устанавливает текущую рабочую директорию
b. Получает текущую рабочую директорию
c. Создаёт новую директорию
d. Удаляет текущую директорию



Выберите верный вариант ответа

Что делает os.getcwd()?

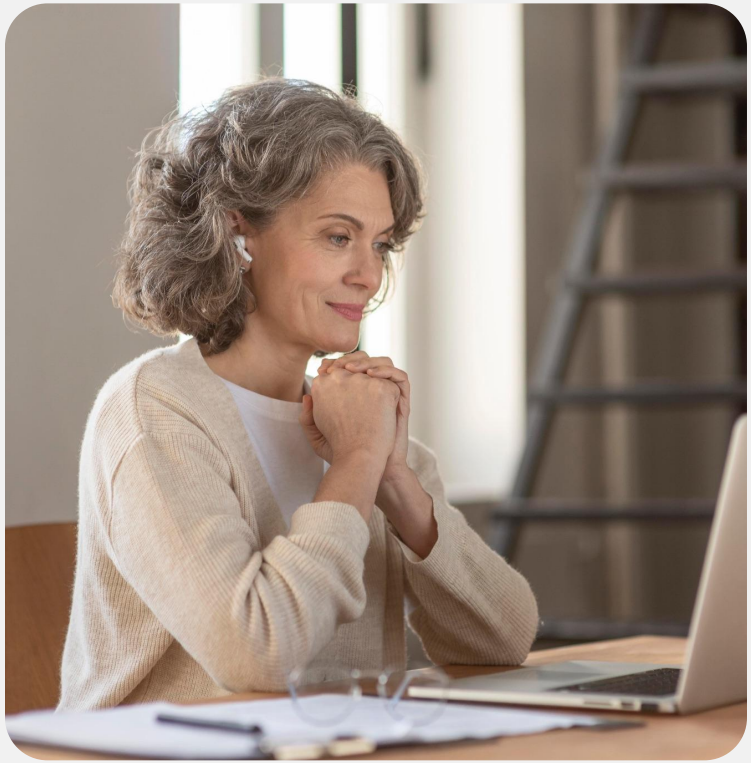
a. Задает текущую рабочую директорию
b. Получает текущую рабочую директорию
c. Создаёт новую директорию
d. Удаляет текущую директорию



Выберите верный вариант ответа

Какой путь является абсолютным?

a. `C:\\Users\\Admin\\Documents\\file.txt`
b. `./Documents/file.txt`
c. `../file.txt`
d. `Documents/file.txt`



Выберите верный вариант ответа

Какой путь является абсолютным?

a. `C:\\Users\\Admin\\Documents\\file.txt`
b. `./Documents/file.txt`
c. `../file.txt`
d. `Documents/file.txt`

IT CAREER HUB

Работа с путями

Разделение пути на компоненты

Функция	Назначение
`os.path.split(path)`	Разделяет путь на две части: директорию и имя файла
`os.path.basename(path)`	Возвращает только имя файла или папки
`os.path.dirname(path)`	Возвращает только директорию без имени файла

Пример получения компонент пути

```python
import os

path = "/subdirectory/example.txt"

# Разделение пути на директорию и файл
directory, file = os.path.split(path)
print(f"Директория: {directory}, Файл: {file}")

# Получение только имени файла
print(f"Имя файла: {os.path.basename(path)}")

# Получение только директории
print(f"Директория: {os.path.dirname(path)}")
```

Полезно знать

IT CAREER HUB

os.path.abspath(path)

Это метод, который преобразует относительный путь в абсолютный

Пример получения абсолютного пути

```python
import os


# Относительный путь
relative_path = "example.txt"

# Преобразование в абсолютный путь
absolute_path = os.path.abspath(relative_path)
print(f"Абсолютный путь: {absolute_path}")
```

Полезно знать

ICH
IT CAREER HUB

os.path.join()

Это метод, который используется для объединения нескольких компонентов пути к файлам и директориям

Пример соединения путей

```python
import os


# Объединение нескольких компонентов
current_dir = os.getcwd()
sub_dir = "docs"
file_name = "data.txt"
full_path = os.path.join(current_dir, sub_dir, file_name)
print(f"Путь: {full_path}")
```

Полезно знать

IT CAREER HUB

os.walk

Это функция, которая позволяет рекурсивно обойти директорию и её содержимое, включая файлы и поддиректории

Пример рекурсивного обхода директории

```python
import os

# Рекурсивный обход директории
for root, dirs, files in os.walk("."):
    print(f"Текущая директория: {root}")
    print(f"Поддиректории: {dirs}")
    print(f"Файлы: {files}")
    print("-" * 50)
```

Пример использования: поиск файлов с нужным расширением

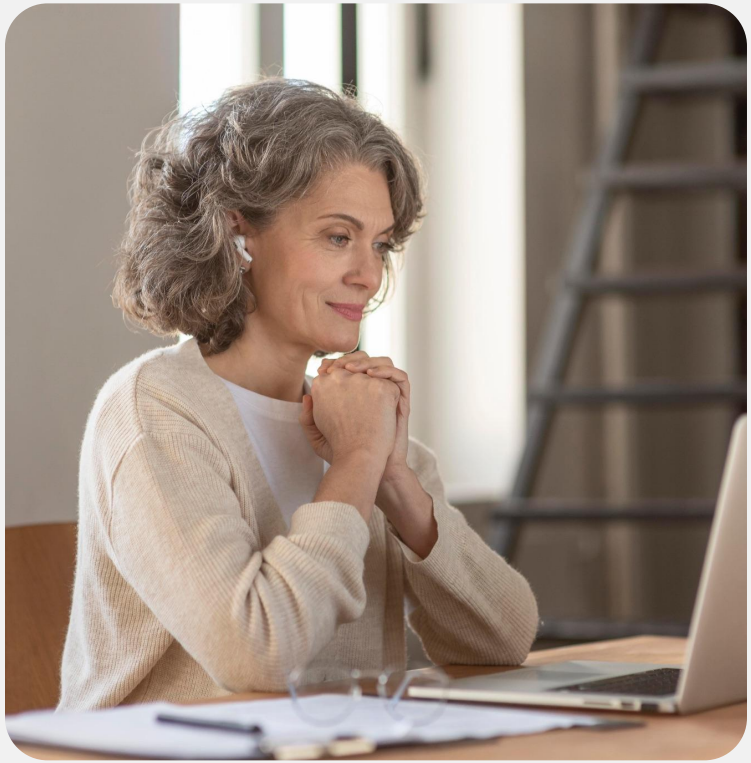
```python
import os


# Поиск всех .txt файлов
for root, dirs, files in os.walk("."):
    for file in files:
        if file.endswith(".txt"):
            print(f"Найден файл: {os.path.join(root, file)}")
```

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

ЗАДАНИЯ

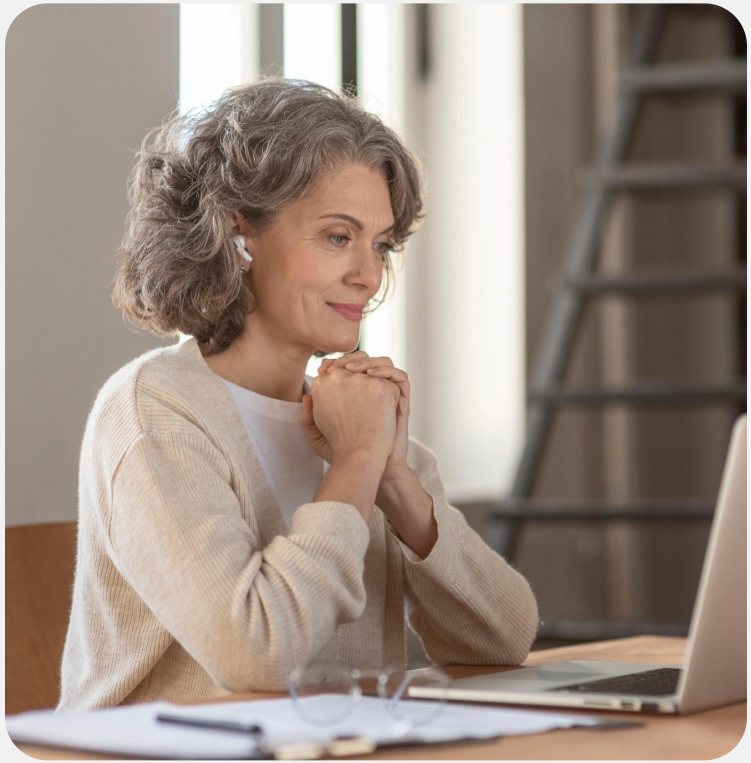


Выберите верный вариант ответа

Какой результат будет выведен при выполнении следующего кода?

```
import os

path = "/home/user/docs/file.txt"
print(os.path.dirname(path))
```

a. /home/user/docs/file.txt
b. /home/user/docs
c. File.txt
d. docs/file.txt

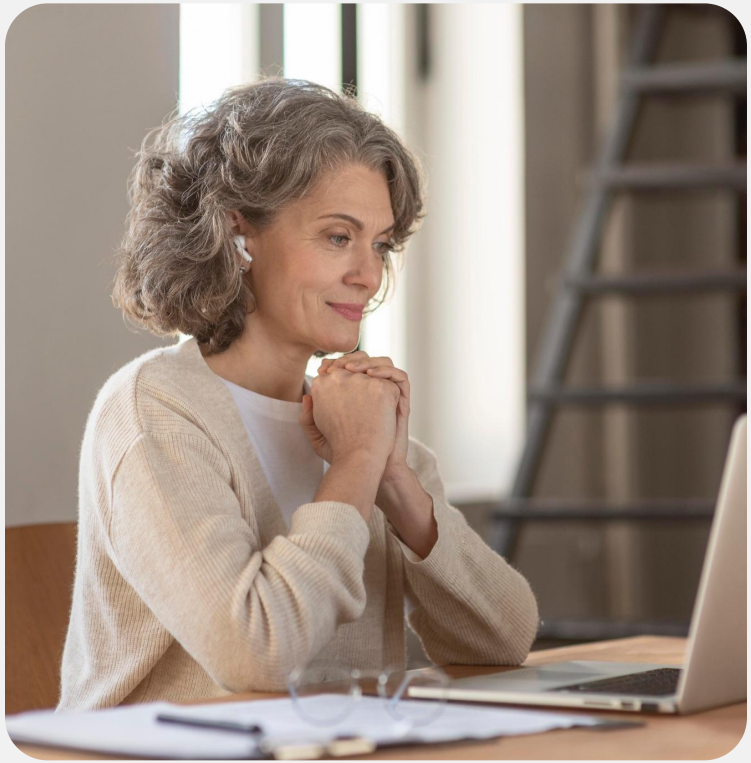


Выберите верный вариант ответа

Какой результат будет выведен при выполнении следующего кода?

```
import os

path = "/home/user/docs/file.txt"
print(os.path.dirname(path))
```

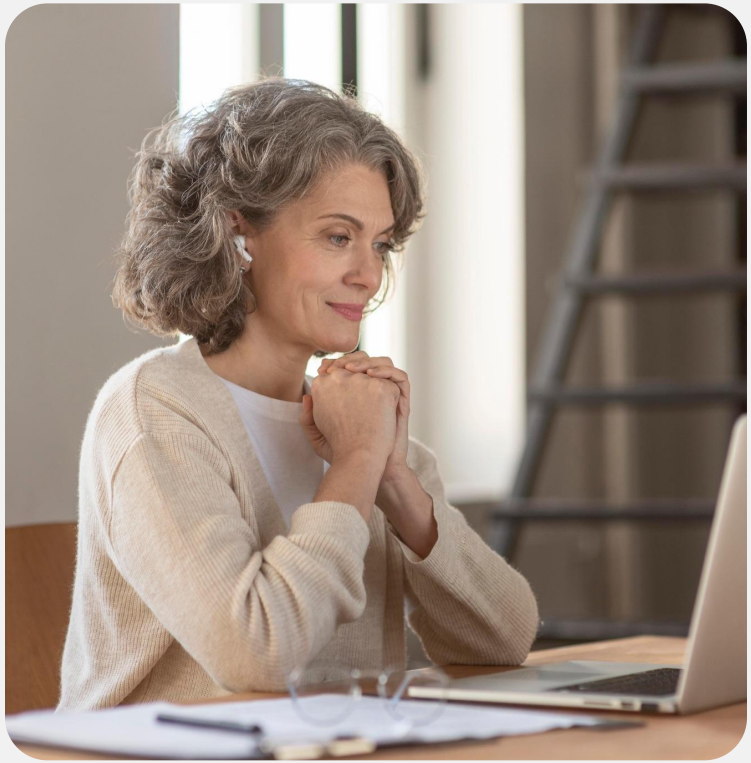
a. /home/user/docs/file.txt
b. /home/user/docs
c. File.txt
d. docs/file.txt



Выберите верный вариант ответа

Какой метод используется для рекурсивного обхода файлов и папок в директории?

a. `os.listdir()`
b. `os.walk()`
c. `os.recursion_walk()`
d. `os.scan()`



Выберите верный вариант ответа

Какой метод используется для рекурсивного обхода файлов и папок в директории?

a. `os.listdir()`
b. `os.walk()`
c. `os.recursion_walk()`
d. `os.scan()`

ICH IT CAREER HUB

Модуль sys. Аргументы командной строки

Модуль sys

Модуль `sys` предоставляет доступ к функциям и переменным, которые используются и поддерживаются интерпретатором Python. Этот модуль позволяет взаимодействовать с окружением Python, управлять аргументами командной строки, путями поиска модулей, стандартным вводом и выводом

```
#Чтобы использовать модуль, нужно его импортировать
import sys
```

Полезно знать

IT CAREER HUB

Скрипт

Это исполняемый файл с кодом

Полезно знать

IT CAREER HUB

sys.argv

Это список аргументов для скрипта

Аргументы командной строки sys.argv

- Первый элемент списка (`sys.argv[0]`) всегда имя скрипта
- Остальные элементы — собственно, аргументы

Синтаксис

```
import sys

# Доступ к аргументам командной строки
arguments = sys.argv
```

Пример работы с аргументами

```python
import sys

# Вывод всех аргументов командной строки
print("Все аргументы:", sys.argv)
# Работа с первым аргументом (если он передан)
if len(sys.argv) > 1:
    print(f"Переданный аргумент: {sys.argv[1]}")
else:
    print("Аргументы не переданы.")
```

Запуск python файла через консоль

```
# Замените script.py на название вашего файла
python script.py argument1
```

Вывод

```
(.venv) tanya@tanya-nitro:~/PycharmProjects/pythonProgramItch$ python script.py argument1
Все аргументы: ['script.py', 'argument1']
Переданный аргумент: argument1
```

Обработка нескольких аргументов

Пример

```
import sys

if len(sys.argv) > 1:
     for i, arg in enumerate(sys.argv[1:],
start=1):
        print(f"Аргумент {i}: {arg}")
else:
    print("Аргументы не переданы.")
```

Запуск через консоль

```
python script.py arg1 arg2 arg3
```

Практическое применение sys.argv

Аргументы командной строки часто используются для передачи данных в скрипт, делая программу более гибкой

Пример: удаление файлов с определённым расширением

```python
import os

# Проверяем, передан ли аргумент
if len(sys.argv) != 2:
    print("Использование: python script.py <расширение>")
    sys.exit(1)

extension = sys.argv[1]

# Удаляем файлы с указанным расширением
for file_name in os.listdir("."):
    if file_name.endswith(extension):
        os.remove(file_name)
        print(f"Удалён файл: {file_name}")
```

Запуск через консоль

```
python script.py .txt
```

Пример: вывод содержимого директории

```python
import sys
import os

# Проверяем, передан ли аргумент
if len(sys.argv) != 2:
    print("Использование: python script.py <директория>")
    sys.exit(1)

directory = sys.argv[1]

# Проверяем существование директории
if not os.path.isdir(directory):
    print(f"Директория '{directory}' не существует.")
    sys.exit(1)

# Выводим содержимое директории
print(f"Содержимое директории '{directory}':")
for item in os.listdir(directory):
    print(f"- {item}")
```

Запуск через консоль

```
python script.py
/home/user/documents
```

Пример: конфигурации логирования

```
import sys
import logging

# Проверяем количество аргументов
if len(sys.argv) != 2:
        print("Использование:  python  script.py
<режим>")
    print("<режим> - debug или release")
    sys.exit(1)

# Получаем режим работы
mode = sys.argv[1]

# Проверяем корректность режима
if mode not in ["debug", "release"]:
    print("Ошибка: режим должен быть 'debug' или
'release'.")
    sys.exit(1)
```

```
# Настройка логирования
log_level = logging.DEBUG if mode == "debug"
else logging.INFO
logging.basicConfig(
       format="%(asctime)s  -  %(levelname)s  -
%(message)s",
    level=log_level
)
# Пример событий во время выполнения программы
logging.info("Инициализация программы")
logging.debug("Чтение конфигурации")
logging.debug("Подключение к базе данных")
logging.info("Обработка данных")
logging.critical("Критическая        ошибка:
соединение с базой данных потеряно!")
logging.info("Завершение работы программы")
```

Пример: конфигурации логирования

Запуск в режиме отладки

```
python script.py debug
```

Запуск в рабочем режиме

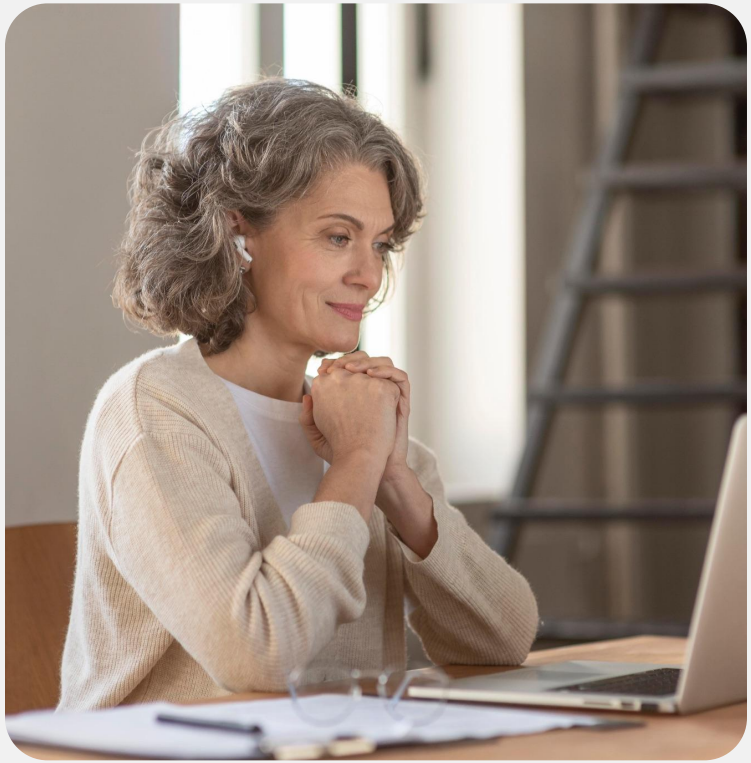
```
python script.py release
```

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

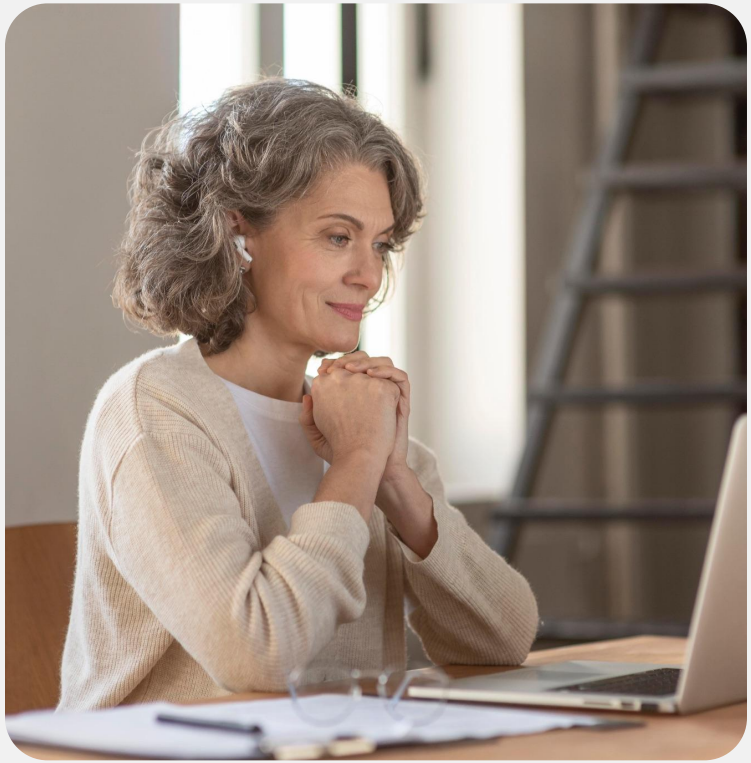
ЗАДАНИЕ



Выберите верный вариант ответа

Какие утверждения о sys.argv верны?

a. `sys.argv` — это список строк
b. `sys.argv` содержит только числовые значения
c. `sys.argv` всегда содержит ровно два элемента
d. Первый элемент `sys.argv[0]` — это имя запущенного скрипта



Выберите верный вариант ответа

Какие утверждения о sys.argv верны?

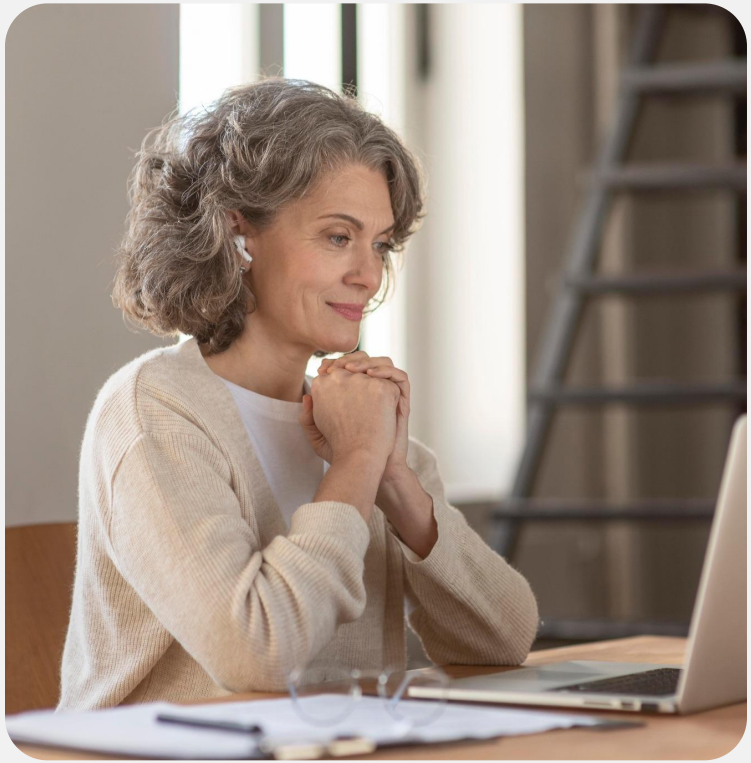
a. `sys.argv` — это список строк
b. `sys.argv` содержит только числовые значения
c. `sys.argv` всегда содержит ровно два элемента
d. Первый элемент `sys.argv[0]` — это имя запущенного скрипта

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

ПРАКТИЧЕСКАЯ РАБОТА



Определение типа объекта

Напишите программу, которая:

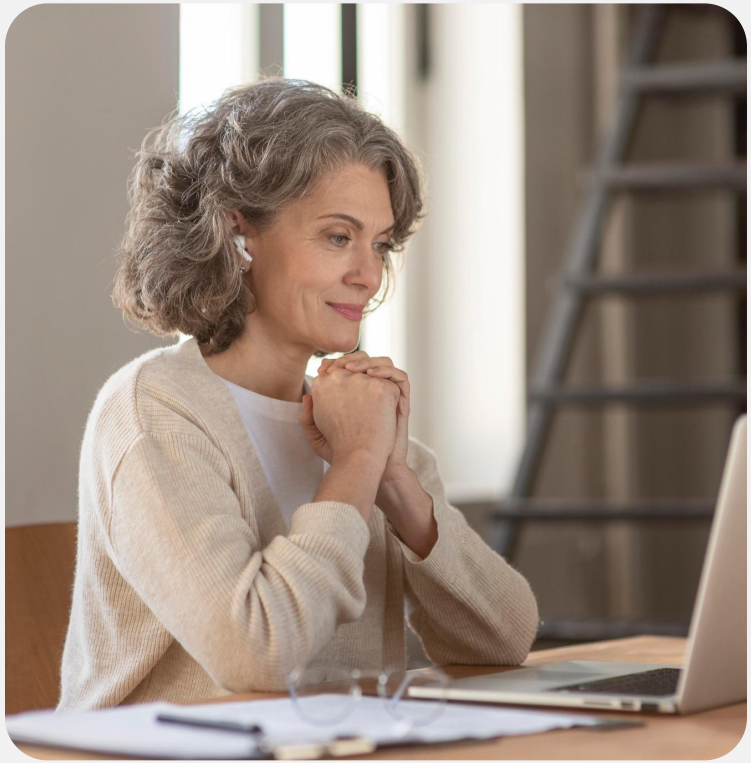
- Запрашивает у пользователя путь.
- Определяет, является ли путь файлом, папкой или он не существует.

Пример ввода

```
Введите путь: example.txt
```

Пример вывода

```
example.txt — это файл.
```



Файлы с заданным расширением

Напишите программу, которая:

- Принимает путь к директории и расширение файлов через аргументы командной строки.
- Ищет файлы с указанным расширением в указанной директории.
- Выводит список найденных файлов

Пример запуска

```
python script.py /home/user/projects .py
```

Пример вывода

```
Найденные файлы в директории '/home/user/projects':
script.py, test.py, main.py
```

Домашнее задание

Список файлов и папок

Напишите программу, которая принимает путь к директории через аргумент командной строки и выводит:

- Отдельно список папок
- Отдельно список файлов

Пример запуска:

```
python script.py /home/user/documents
```

Пример вывода:

```
Содержимое директории '/home/user/documents':
Папки:
- folder1
- folder2
Файлы:
- file1.txt
- file2.txt
- notes.docx
```

Домашнее задание

Поиск и удаление файлов с указанным расширением

Напишите программу, которая

- Принимает путь к директории и расширение файлов через аргумент командной строки.
- Рекурсивно ищет файлы с этим расширением во всех вложенных папках.
- Спрашивает у пользователя, хочет ли он удалить найденные файлы.
- Если пользователь подтверждает, удаляет их.

Пример запуска

```
python script.py /home/user/PycharmProjects/project1 .log
Пример вывода:
Найдены файлы с расширением '.log':
- logs/error.log
- logs/system.log
- logs/backup/old.log
- logs/backup/debug.log

Вы хотите удалить эти файлы? (y/n): y
Удаление завершено.
```

Заключение