

LU08.L01: Arbeiten mit Dateien und Verzeichnissen

```
import subprocess
import sys

def create_directory(directory_name):
    """
    Creates a directory with the specified name.
    :param directory_name:
    :return:
    """
    try:
        result = subprocess.run(
            'mkdir ./python_bash_test&&cd python_bash_test',
            shell=True, # Use a shell to interpret the command
            check=True, # Raise an error if the command fails
            text=True, # Capture output as a string
            stdout=subprocess.PIPE, # Redirect standard output
            stderr=subprocess.PIPE # Redirect standard error
        )
    except subprocess.CalledProcessError as e:
        print(f'Error executing the command: {e.cmd}')
        print(f'Error message: {e.stderr}')
        sys.exit(1)
    except Exception as ex:
        print(f'An unexpected error occurred: {ex}')
        sys.exit(1)

def create_files(directory_name, file_names):
    """
    Creates files with the specified names in the specified directory.
    :param directory_name:
    :param file_names:
    :return:
    """
    for filename in file_names:
        try:
            result = subprocess.run(
                f'touch ./{directory_name}/' + filename,
                shell=True, # Use a shell to interpret the command
                check=True, # Raise an error if the command fails
                text=True, # Capture output as a string
                stdout=subprocess.PIPE, # Redirect standard output
                stderr=subprocess.PIPE # Redirect standard error
            )
```

```
except subprocess.CalledProcessError as e:
    print(f'Error executing the command: {e.cmd}')
    print(f'Error message: {e.stderr}')
    sys.exit(1)
except Exception as ex:
    print(f'An unexpected error occurred: {ex}')
    sys.exit(1)

def list_files(directory_name):
    """
    List the files in the specified directory.
    :param directory_name:
    :return:
    """
    try:
        result = subprocess.run(
            f'ls ./{directory_name}',
            shell=True, # Use a shell to interpret the command
            check=True, # Raise an error if the command fails
            text=True, # Capture output as a string
            stdout=subprocess.PIPE, # Redirect standard output
            stderr=subprocess.PIPE # Redirect standard error
        )
        print(result.stdout)
    except subprocess.CalledProcessError as e:
        print(f'Error executing the command: {e.cmd}')
        print(f'Error message: {e.stderr}')
        sys.exit(1)
    except Exception as ex:
        print(f'An unexpected error occurred: {ex}')
        sys.exit(1)

def file_size(directory_name):
    """
    Shows the size of the files in the specified directory.
    :param directory_name:
    :return:
    """
    try:
        result = subprocess.run(
            f'du -h ./{directory_name}',
            shell=True, # Use a shell to interpret the command
            check=True, # Raise an error if the command fails
            text=True, # Capture output as a string
            stdout=subprocess.PIPE, # Redirect standard output
            stderr=subprocess.PIPE # Redirect standard error
        )
        print(result.stdout)
    except subprocess.CalledProcessError as e:
        print(f'Error executing the command: {e.cmd}')
```

```
    print(f'Error message: {e.stderr}')
    sys.exit(1)
except Exception as ex:
    print(f'An unexpected error occurred: {ex}')
    sys.exit(1)

def delete_files(directory_name, filename):
    """
    Deletes files with the specified names in the specified directory.
    :param directory_name:
    :param filename:
    :return:
    """

    try:
        result = subprocess.run(
            f'rm ./[directory_name]/' + filename,
            shell=True, # Use a shell to interpret the command
            check=True, # Raise an error if the command fails
            text=True, # Capture output as a string
            stdout=subprocess.PIPE, # Redirect standard output
            stderr=subprocess.PIPE # Redirect standard error
        )
    except subprocess.CalledProcessError as e:
        print(f'Error executing the command: {e.cmd}')
        print(f'Error message: {e.stderr}')
        sys.exit(1)
    except Exception as ex:
        print(f'An unexpected error occurred: {ex}')
        sys.exit(1)

if __name__ == '__main__':
    directory = 'python_bash_test'
    files = ['file1.txt', 'file2.txt', 'file3.txt']

    create_directory(directory)
    create_files(directory, files)
    list_files(directory)
    file_size(directory)
    delete_files(directory, 'file3.txt')
```

M122-LU08



Marcel Suter

From:
<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:
<https://wiki.bzz.ch/modul/m122/learningunits/lu08/loesungen/dateien>

Last update: **2024/12/16 09:31**

