

# Navigating the File System

## Navigating the File System

### Current Working Directory:

```
import os
# Get the current working directory
cwd = os.getcwd()
print("Current working directory:", cwd)
```

✓  
0s



```
import os

# Get the current working directory
cwd = os.getcwd()
print("Current working directory:", cwd)
```



Current working directory: /content

## Change Directory:

```
# Change to a different directory  
os.chdir('/path/to/directory')  
print("Changed directory to:", os.getcwd())
```

✓  
0s



```
# Change to a different directory  
os.chdir('/content/sample_data')  
print("Changed directory to:", os.getcwd())
```

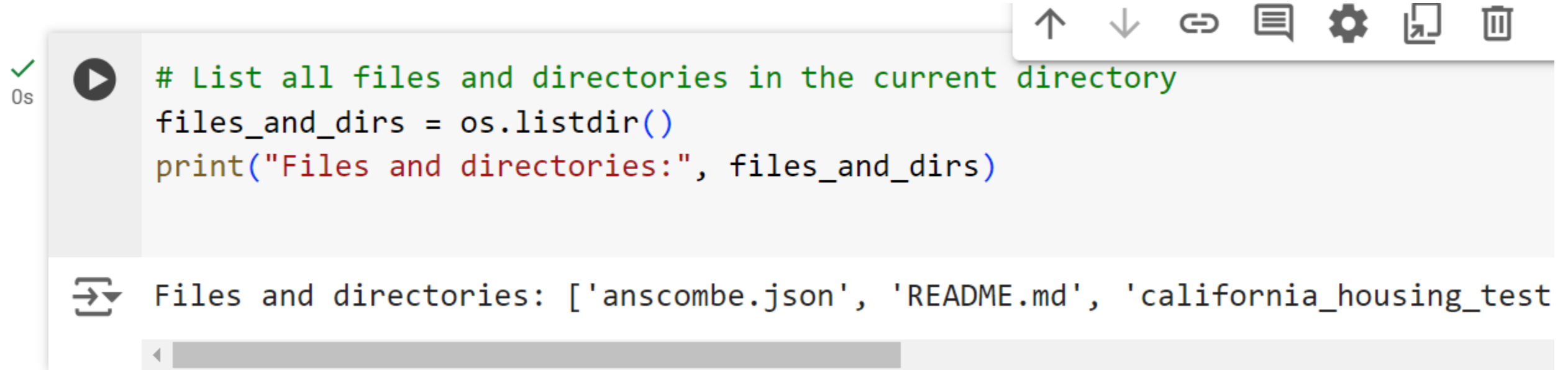


Changed directory to: /content/sample\_data



## List Files and Directories:

```
# List all files and directories in the current directory
files_and_dirs = os.listdir()
print("Files and directories:", files_and_dirs)
```



The screenshot shows a code editor interface. At the top, there is a toolbar with icons for navigation (up, down), linking, commenting, settings, and file management. Below the toolbar, a code editor displays a Python script. To the left of the code editor, there is a play button icon and a green checkmark with '0s' next to it, indicating successful execution. The output of the script is shown in a separate line at the bottom, starting with a magnifying glass icon.

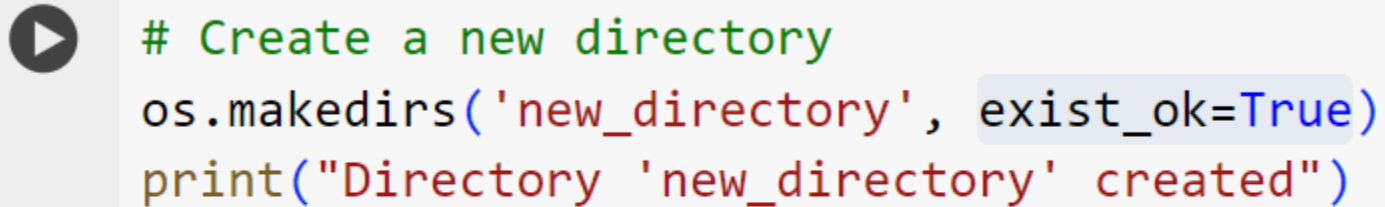
```
# List all files and directories in the current directory
files_and_dirs = os.listdir()
print("Files and directories:", files_and_dirs)
```

Files and directories: ['anscombe.json', 'README.md', 'california\_housing\_test']

# Creating Directories and Files

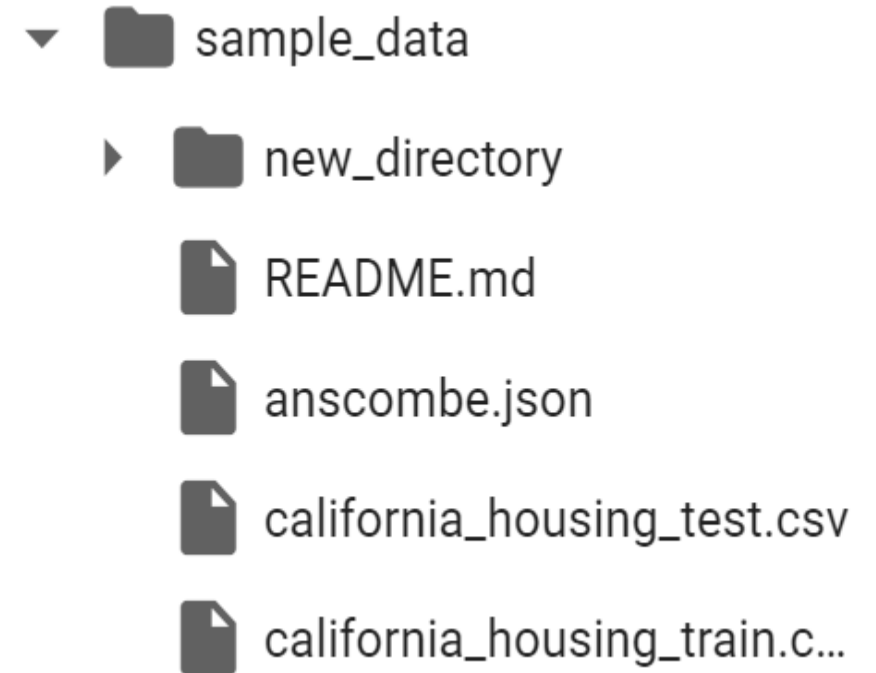
## Create a Directory:

```
# Create a new directory
os.makedirs('new_directory', exist_ok=True)
print("Directory 'new_directory' created")
```

A terminal window with a light gray background. On the left, there is a green checkmark and the text '0s'. Next to it is a black play button icon. The code is displayed in a monospaced font with syntax highlighting: '# Create a new directory' in green, 'os.makedirs('new\_directory', exist\_ok=True)' in blue, and 'print("Directory 'new\_directory' created")' in red. The output 'Directory 'new\_directory' created' is shown below the code in a lighter font.

```
# Create a new directory
os.makedirs('new_directory', exist_ok=True)
print("Directory 'new_directory' created")
```

Directory 'new\_directory' created



`exist_ok=True`: argument ensures that no error is raised if the directory already exists. Without this argument, an `OSError` would be raised if the directory already exists.

# Creating Directories and Files

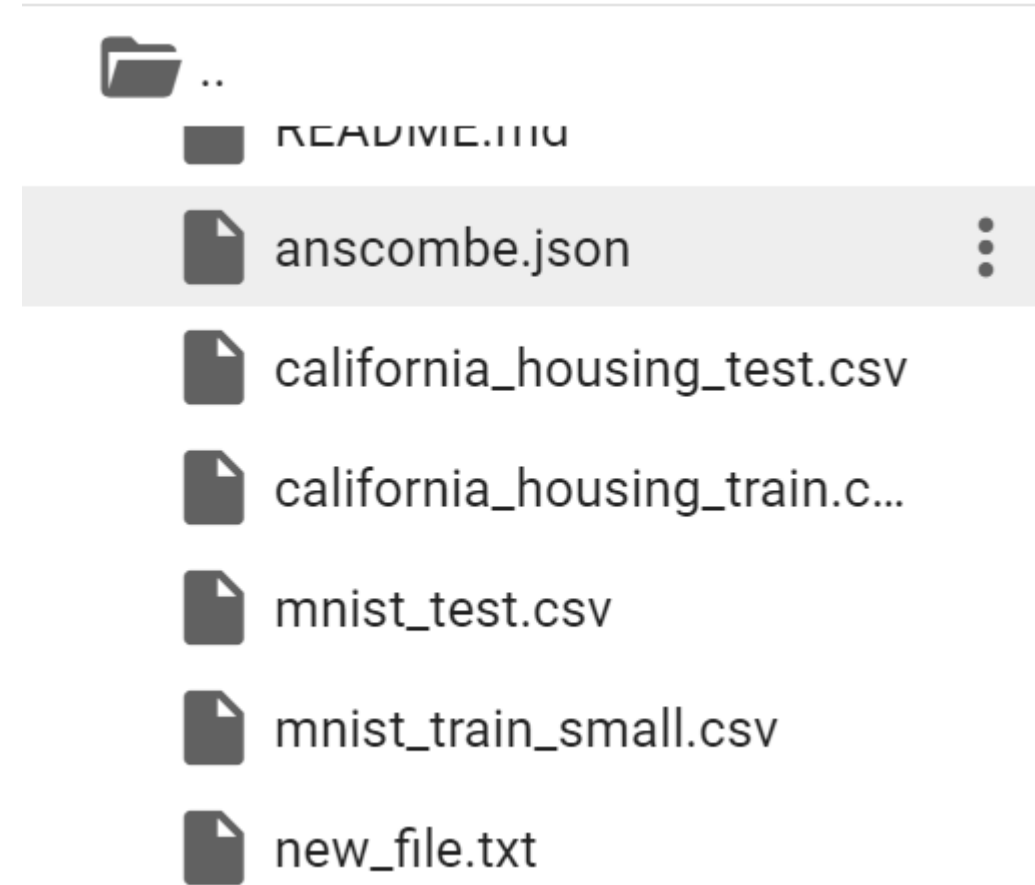
## Create a File:

```
# Create a new file
with open('new_file.txt', 'w') as file:
    file.write("This is a new file.")
print("File 'new_file.txt' created")
```

✓  
0s

```
# Create a new file
with open('new_file.txt', 'w') as file:
    file.write("This is a new file.")
print("File 'new_file.txt' created")
```

File 'new\_file.txt' created



# Creating Directories and Files

## Deleting Directories and Files

# Delete a file

```
os.remove('new_file.txt')
```

```
print("File 'new_file.txt' deleted")
```

✓  
0s



```
# Delete a file
```

```
os.remove('new_file.txt')
```

```
print("File 'new_file.txt' deleted")
```



```
File 'new_file.txt' deleted
```



..



new\_directory



README.md



anscombe.json



california\_housing\_test.csv



california\_housing\_train.c...



mnist\_test.csv



mnist\_train\_small.csv

# Creating Directories and Files

## Delete an Empty Directory:

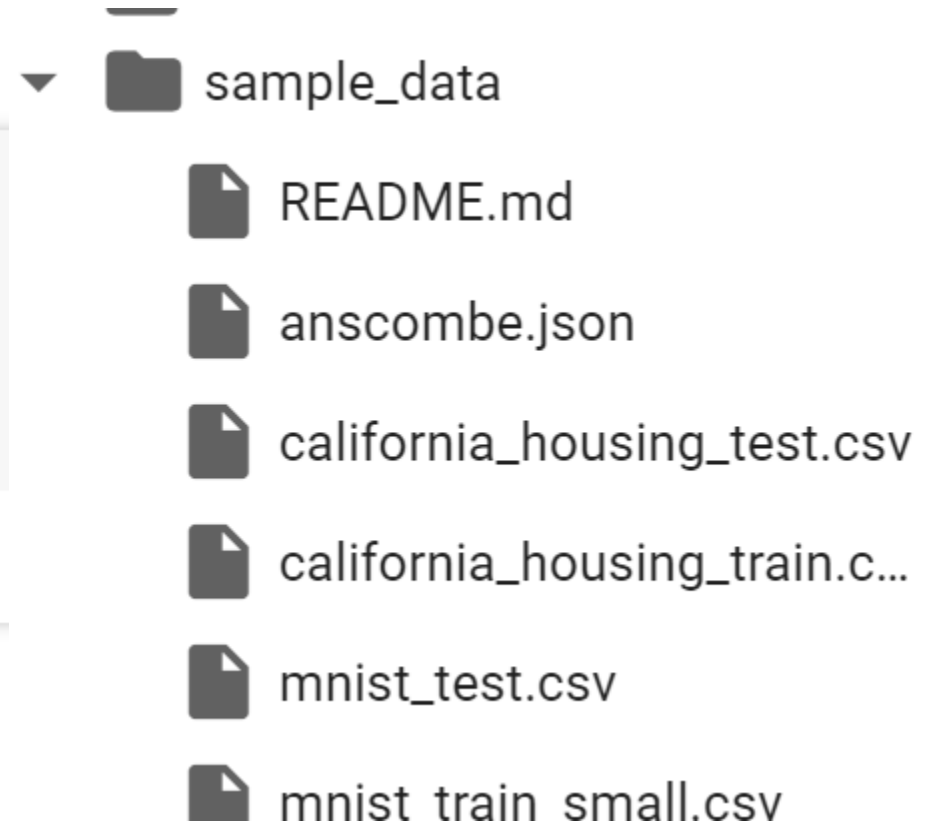
```
# Delete an empty directory
os.rmdir('new_directory')
print("Directory 'new_directory' deleted")
```



```
# Delete an empty directory
os.rmdir('new_directory')
print("Directory 'new_directory' deleted")
```



Directory 'new\_directory' deleted





# Creating Directories and Files

## Delete a Directory with Contents:

```
import shutil
# Delete a directory with all its contents
shutil.rmtree('directory_with_contents')
print("Directory 'directory_with_contents' and its contents deleted")
```

# Moving and Copying Files and Directories

## Move a File or Directory:

# Move a file

```
shutil.move('source_file.txt', 'destination_directory/')  
print("File moved to destination directory")
```

# Move a directory

```
shutil.move('source_directory', 'destination_directory/')  
print("Directory moved to destination directory")
```

## Using pathlib for an Object-Oriented Approach

- The pathlib module offers a more modern and object-oriented approach to file system operations.

### Navigating and Creating Paths:

```
from pathlib import Path
```

```
# Define a path
```

```
path = Path('new_directory')
```

```
# Create a directory
```

```
path.mkdir(parents=True, exist_ok=True)
```

```
print(f"Directory {path} created")
```

## Iterating Over Directory Contents:

```
# Iterate over all items in a directory
for item in path.iterdir():
    print(item)
```

## Delete a File or Directory:

```
# Delete a file
file_path = Path('new_file.txt')
file_path.unlink()
print(f"File {file_path} deleted")
```

```
# Delete a directory
path.rmdir()
print(f"Directory {path} deleted")
```

## Check if a File or Directory Exists:

```
path = Path('some_directory_or_file')
```

```
if path.exists():
```

```
    print(f"{path} exists")
```

```
else:
```

```
    print(f"{path} does not exist")
```

## Task:

Write a Python script that lists all the files and directories in the current working directory and prints them separately.

```
import os
# Get the current working directory
cwd = os.getcwd()

# List all files and directories
files = []
directories = []
for item in os.listdir(cwd):
    if os.path.isfile(item):
        files.append(item)
    elif os.path.isdir(item):
        directories.append(item)
print("Files:", files)
print("Directories:", directories)
```