

In-Depth Python File Handling for Django Developers

1. Opening Files (`open()`)

```
open() is used to access a file. It returns a file object for reading or writing.
```

Syntax:

```
file = open('filename.txt', 'mode')
```

Modes:

```
'r' - Read, 'w' - Write, 'a' - Append, 'x' - Create, 'b' - Binary, 't' - Text
```

2. Reading Files

```
with open('data.txt', 'r') as f:
```

```
    content = f.read()
```

```
with open('data.txt', 'r') as f:
```

```
    line = f.readline()
```

```
with open('data.txt', 'r') as f:
```

```
    lines = f.readlines()
```

```
with open('data.txt', 'r') as f:
```

```
    for line in f:
```

```
        print(line.strip())
```

3. Writing and Appending

```
with open('log.txt', 'w') as f:
```

```
    f.write("Overwriting the file.\n")
```

```
with open('log.txt', 'a') as f:
```

```
    f.write("Appending new log.\n")
```

4. Binary File Handling

```
with open('photo.jpg', 'rb') as f:
```

```
    data = f.read()
```

In-Depth Python File Handling for Django Developers

```
with open('copy.jpg', 'wb') as f:  
    f.write(data)
```

5. Working with File Paths

```
import os  
  
filepath = os.path.join('data', 'info.txt')  
  
from pathlib import Path  
file = Path('data/info.txt')  
if file.exists():  
    print(file.read_text())
```

6. Checking and Deleting Files

```
import os  
  
if os.path.exists("sample.txt"):  
    os.remove("sample.txt")  
  
from pathlib import Path  
file = Path("sample.txt")  
if file.exists():  
    file.unlink()
```

7. CSV File Handling

```
import csv  
  
with open('people.csv', 'w', newline='') as f:  
    writer = csv.writer(f)  
    writer.writerow(['Name', 'Age'])  
    writer.writerow(['Ram', 25])  
  
with open('people.csv', 'r') as f:  
    reader = csv.reader(f)  
    for row in reader:  
        print(row)
```

8. JSON File Handling

In-Depth Python File Handling for Django Developers

```
import json
data = {'name': 'Sita', 'age': 30}

with open('data.json', 'w') as f:
    json.dump(data, f)

with open('data.json', 'r') as f:
    result = json.load(f)
    print(result)
```

9. Using 'with' for Safe File Handling

```
with open('file.txt', 'r') as f:
    data = f.read()
```

10. Handling Large Files

```
with open('large.txt', 'r') as f:
    while chunk := f.read(1024):
        print(chunk)
```

11. Directory Operations

```
import os
files = os.listdir('myfolder')
os.mkdir('new_folder')
os.rmdir('new_folder')
```