Reading Files in Python: Overview and Techniques

Python provides several ways to read file content efficiently. Here are the most common methods and their usage explained step by step.

1. Opening Files

Use the `open()` function to access files. The mode `"r"` (read) is used to open files for reading.

Syntax:

```
with open("filename.txt", "r") as file:
    # Perform file operations
```

This ensures the file is properly closed after operations, even if an error occurs.

2. Methods for Reading Files

a. `read()` Method

- Reads the **entire file** content as a single string.
- Optionally, pass an integer to read a specific number of characters.

Example:

```
with open("sample.txt", "r") as file:
    # Read the entire file
    content = file.read()
    print(content)

    # Read the first 20 characters
    partial_content = file.read(20)
    print(partial_content)
```

b. `readline()` Method

- Reads **one line at a time** from the file.
- You can pass an integer to read a specific number of characters from that line.

Example:

```
with open("sample.txt", "r") as file:
    # Read the first line
    first_line = file.readline()
    print(first_line)

    # Read the first 10 characters of the next line
    partial_line = file.readline(10)
    print(partial_line)
```

c. `readlines()` Method

- Reads all lines in the file and returns them as a **list of strings**, where each string is a line.
- Useful for iterating over file lines.

Example:

```
with open("sample.txt", "r") as file:
    # Read all lines into a list
    lines = file.readlines()
    print(lines)

    # Iterate over each line
    for line in lines:
        print(line.strip())  # Use .strip() to remove trailing newline characters
```

3. Absolute vs. Relative Paths

When specifying file paths, you can use:

- **Absolute Paths**: Full path starting from the root directory. Example:
  `"/home/user/documents/sample.txt"`
- **Relative Paths**: Path relative to the current working directory. Example: `"sample.txt"`

4. Iterating Directly Over a File

Files in Python are iterable, so you can loop through them directly line by line.

Example:

```
with open("sample.txt", "r") as file:
    for line in file:
```

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

5. Combining Techniques

For more complex operations, you can combine these methods.

Example: Filtering Lines

```
with open("sample.txt", "r") as file:
    lines = file.readlines()
    for line in lines:
        if "specific word" in line:
            print(line.strip())
```

Summary of Methods:

Method	Use Case	Output Type
`read()`	Read entire file or specific chars	String
`readline()`	Read one line or specific chars in it	String
`readlines()`	Read all lines into a list	List of Strings

By mastering these methods, you can handle files flexibly and efficiently in Python.