Python Files Methods:

Here are some of the common methods used in Python for working with files:

1. `open()`: This method is used to open a file and returns a file object. The first parameter is the name of the file to open, and the second parameter is the mode. For example:

```
file = open("example.txt", "r")
```

2. `read()`: This method is used to read the contents of a file. It takes an optional parameter that specifies the number of bytes to read. For example:

```
content = file.read() # Read the entire file
content = file.read(10) # Read the first 10 bytes of the file
```

3. `write()`: This method is used to write data to a file. It takes a string as input and writes it to the file. For example:

```
file.write("Hello, World!")
```

4. `close()`: This method is used to close an open file. It's important to close files when you're done with them to free up resources. For example:

```
file.close()
```

5. `readline()`: This method is used to read a single line from a file. For example:

```
line = file.readline()
```

6. `seek()`: This method is used to move the file pointer to a specific location in the file. For example:

```
file.seek(0) # Move the file pointer to the beginning of the file
```

```
```

7. `tell()`: This method is used to get the current file position. For example:

```
position = file.tell() # Get the current file position
```

8. `truncate()`: This method is used to truncate a file to a specified size. For example:

```
file.truncate(10) # Truncate the file to 10 bytes
```

9. `flush()`: This method is used to flush the internal buffer of a file object to disk. For example:

```
file.flush() # Flush the buffer to disk
```

These are just a few of the methods available for working with files in Python. By combining these methods with Python's powerful string manipulation and control structures, you can create complex programs that read and write data to files with ease.