

## Opening Text Files

To open a file, you need to open the file in read, write or append mode.

```
file = open(filename, mode)
```

Substitute `mode` for the mode below

Mode	Description
<b>r</b>	This mode will be open for reading only
<b>w</b>	This mode will be open for writing only. If file containing that name does not exist, it will create a new one
<b>a</b>	This mode will append to the previous output of that file
<b>r+</b>	This mode will be open for both reading and writing
<b>x</b>	Creates a new file

## Reading from Text Files

To read from a file use

```
file.method()
```

Substitute `.method()` for the method you want to use on the file

Method	Description
<b>.read()</b>	This method reads the entire file
<b>.readline()</b>	This method reads one line of the file
<b>.readlines()</b>	This method returns a list of lines from the file
<b>.close()</b>	Closes file

## Writing to Text Files

To write to a file use

```
file.method()
```

Substitute `.method()` for the method you want to use on the file. Substitute *data* for the data you want to write to the file

Method	Description
<b>.write(data)</b>	This method writes <i>data</i> to a file overwriting existing data
<b>.writelines(data)</b>	This method <i>data</i> as a list of strings to the file overwriting existing data
<b>.append(data)</b>	This method appends <i>data</i> to the file instead of overwriting.
<b>.close()</b>	Closes file

## Opening Binary Files

To open a binary file, you need to open the file in read, write or append mode.

```
file = open(filename, mode)
```

Substitute `mode` for the mode below

Mode	Description
<b>rb</b>	This mode will be open for reading only
<b>wb</b>	This mode will be open for writing only. If file containing that name does not exist, it will create a new one
<b>ab</b>	This mode will append to the previous output of that file
<b>rb+</b>	This mode will be open for both reading and writing
<b>xb</b>	Create a new binary file

## Reading from Binary Files

To read from a file use

```
file.method()
```

Substitute `.method()` for the method you want to use on the file

Method	Description
<code>.read()</code>	This method reads the entire file
<code>.readline()</code>	This method reads one line of the file
<code>.readlines()</code>	This method returns a list of lines from the file
<code>.close()</code>	Closes file

## Writing to Binary Files

To write to a file use

```
file.method()
```

Substitute `.method()` for the method you want to use on the file. Substitute `bytearray` for the data you want to write to the file.

**Note that you need to write bytes not integers or text strings otherwise you'll get an error when you run the program. To do this add 'b' before a string, such as:**

```
b"String"
```

If you're using integers use a method called `.to_bytes()` to convert it

```
num=4  
binarynumbertowrite = num.to_bytes(length=1, byteorder='little')
```

Length is the number of bytes

Now you'll be able to use the file methods in binary mode.

Method	Description
<code>.write(<i>bytearray</i>)</code>	This method writes <i>bytearray</i> to a file
<code>.writelines(<i>bytearra</i>)</code>	This method <i>bytearray</i> as a list of strings to the file overwriting existing data
<code>.append(<i>bytearra</i>)</code>	This method appends <i>bytearray</i> to the file instead of overwriting the file.
<code>.close()</code>	Closes file

## Using Pickle to Write Binary Files

To write your data in binary format, first we need to convert it to a sequence of bytes. We can do this with the pickle module using a process called pickling. Pickling is the process where data objects such as integers, strings, lists, and dictionaries are converted into a byte stream.

Import the following module at the top of your program.

```
import pickle
```

Open your file as normal

```
file = open(filename, mode)
```

To load a file use the `pickle.load()` method

```
pickle.load (file-to-read-from)
```

For example

```
datafromfile = pickle.load(file)
```

To write to a file, use the `pickle.dump()` method

```
pickle.dump (data-to-be-written, file-to-write-to)
```

For example

```
pickle.dump ("Data to be written", file)
```