Files let you

- Save data while program is not running
- Send data to others
- Work with large amounts of data

In Python, files are objects

- But use a special `open` constructor

```
f = open("testing.txt", "w")
f.write("let's write some words")
f.write("in a file")
f.write("and see what")
f.write("happens")
f.close()
```

Why do you need to close files?

- Buffering

```
f = open("testing.txt", "w")
f.write("let's write some words")
f.write("in a file")
f.write("and see what")
f.write("happens")
f.close()
```

Creates the following file:

```
let's write some wordsin a fileand see whathappens
```

Be careful about:

- Exact formatting (spaces, etc.)
- Line breaks, `\n`

```
f = open("testing.txt", "w")
f.write("Let's write some words ")
f.write("in a file\n")
f.write("and see what ")
f.write("happens.")
f.close()
```

Creates the following file:

```
Let's write some words in a file
and see what happens.
```

The `write` method only accepts strings

- Use `str()` to convert

Line endings are platform-specific

```
f = open("testing.txt", "w")
f.write("Let's write some words ")
f.write("in a file\n")
f.write("and see what ")
f.write("happens.")
f.close()
```

In order to make sure the file gets closed:

- try-except blocks
- with statement

```
with open("testing.txt", "w") as f:
    f.write("Let's write some words ")
    f.write("in a file\n")
    f.write("and see what ")
    f.write("happens.")
```

```
f = open("testing.txt", "w")
```

Details of the open statement:

File name:

- Relative to current directory
- Absolute

Mode:

- “w” – writing
  - Will replace an existing file!
- “a” – appending
  - Will add data to end of the file
- “r” – reading
  - Get data from the file
- “r+” – both reading and writing

Text modes

Given this file:

```
Let's write some words in a file
and see what happens.
```

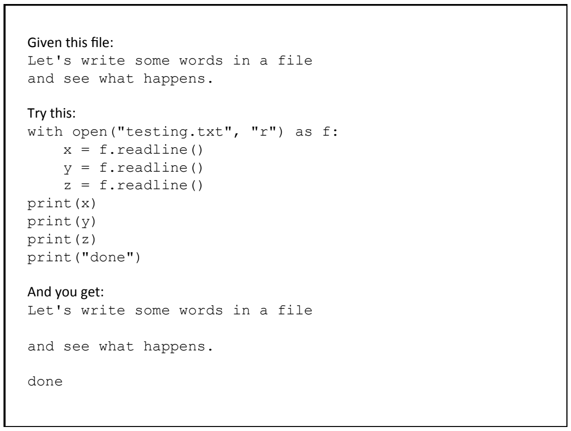
Try this:

```
with open("testing.txt", "r") as f:
    x = f.readline()
    y = f.readline()
print(x)
print(y)
print("done")
```

And you get:

```
Let's write some words in a file

and see what happens.
done
```



Given this file:

```
Let's write some words in a file
and see what happens.
```

Try this:

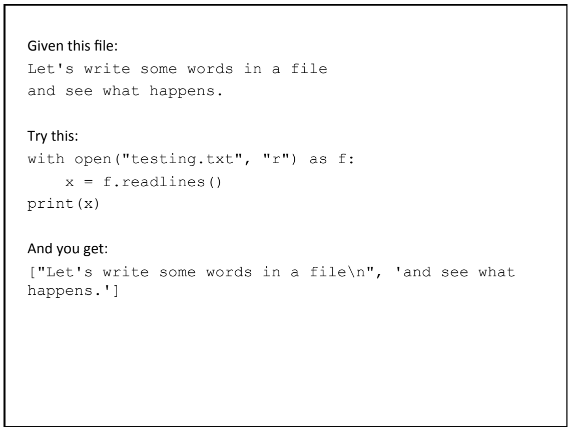
```
with open("testing.txt", "r") as f:
    x = f.readline()
    y = f.readline()
    z = f.readline()
print(x)
print(y)
print(z)
print("done")
```

And you get:

```
Let's write some words in a file

and see what happens.

done
```



Given this file:

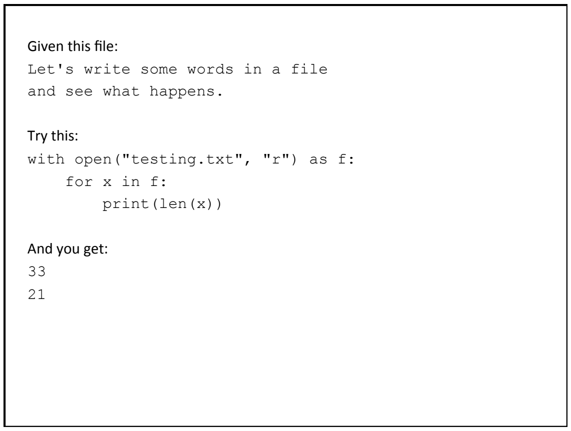
```
Let's write some words in a file
and see what happens.
```

Try this:

```
with open("testing.txt", "r") as f:
    x = f.readlines()
print(x)
```

And you get:

```
["Let's write some words in a file\n", 'and see what
happens.']
```



Given this file:

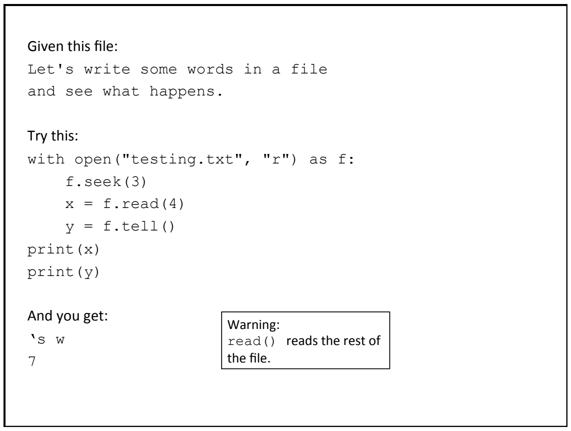
```
Let's write some words in a file
and see what happens.
```

Try this:

```
with open("testing.txt", "r") as f:
    for x in f:
        print(len(x))
```

And you get:

```
33
21
```



Given this file:

```
Let's write some words in a file
and see what happens.
```

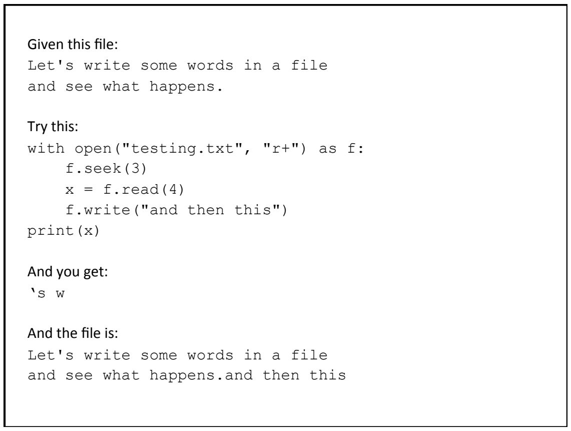
Try this:

```
with open("testing.txt", "r") as f:
    f.seek(3)
    x = f.read(4)
    y = f.tell()
print(x)
print(y)
```

And you get:

```
‘s w
7
```

Warning:
`read()` reads the rest of the file.



Given this file:

```
Let's write some words in a file
and see what happens.
```

Try this:

```
with open("testing.txt", "r+") as f:
    f.seek(3)
    x = f.read(4)
    f.write("and then this")
print(x)
```

And you get:

```
‘s w
```

And the file is:

```
Let's write some words in a file
and see what happens.and then this
```