

Python 2.4 and Earlier

433

```
f = open("myfile", "rb")
try:
    byte = f.read(1)
    while byte != "":
        # Do stuff with byte.
        byte = f.read(1)
finally:
    f.close()
```

Python 2.5-2.7

```
with open("myfile", "rb") as f:
    byte = f.read(1)
    while byte != "":
        # Do stuff with byte.
        byte = f.read(1)
```

Note that the with statement is not available in versions of Python below 2.5. To use it in v 2.5 you'll need to import it:

```
from __future__ import with_statement
```

In 2.6 this is not needed.

Python 3

In Python 3, it's a bit different. We will no longer get raw characters from the stream in byte mode but byte objects, thus we need to alter the condition:

```
with open("myfile", "rb") as f:
    byte = f.read(1)
    while byte != b"":
        # Do stuff with byte.
        byte = f.read(1)
```

Or as benhoyt says, skip the not equal and take advantage of the fact that `b""` evaluates to false. This makes the code compatible between 2.6 and 3.x without any changes. It would also save you from changing the condition if you go from byte mode to text or the reverse.

```
with open("myfile", "rb") as f:
    byte = f.read(1)
    while byte:
        # Do stuff with byte.
        byte = f.read(1)
```

python 3.8

From now on thanks to `:=` operator the above code can be written in a shorter way.

```
with open("myfile", "rb") as f:
    while (byte := f.read(1)):
        # Do stuff with byte.
```