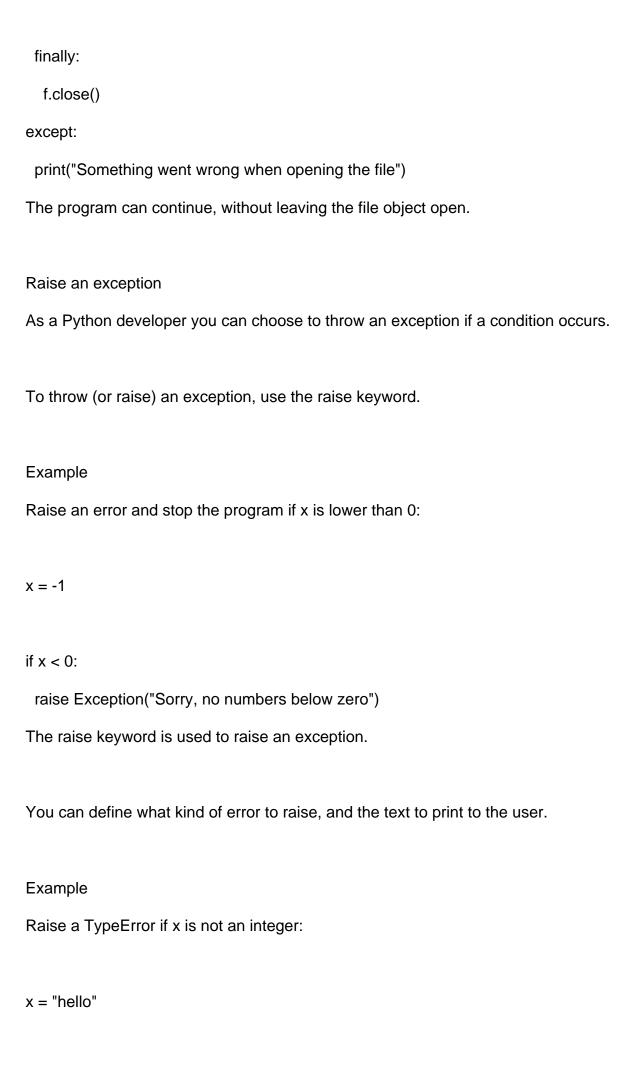


Example
This statement will raise an error, because x is not defined:
print(x)
Many Exceptions
You can define as many exception blocks as you want, e.g. if you want to execute a special block of
code for a special kind of error:
Example
Print one message if the try block raises a NameError and another for other errors:
try:
print(x)
except NameError:
print("Variable x is not defined")
except:
print("Something else went wrong")
ADVERTISEMENT
Else
You can use the else keyword to define a block of code to be executed if no errors were raised:
Example
In this example, the try block does not generate any error:
try:

```
print("Hello")
except:
 print("Something went wrong")
else:
 print("Nothing went wrong")
Finally
The finally block, if specified, will be executed regardless if the try block raises an error or not.
Example
try:
 print(x)
except:
 print("Something went wrong")
finally:
 print("The 'try except' is finished")
This can be useful to close objects and clean up resources:
Example
Try to open and write to a file that is not writable:
try:
 f = open("demofile.txt")
 try:
  f.write("Lorum Ipsum")
 except:
  print("Something went wrong when writing to the file")
```



if not type(x) is int:

raise TypeError("Only integers are allowed")