

# how to print type error handling in python

Use Try-except block

To **print** an **error** in **Python**, use the **try-except** block. The **try block** is similar to the if statement, and the testing code is kept inside the try block. If the code raises an error, it is handled in an except statement. If the code does not raise any error, then this code is treated as a standard code. The code usually executes, and the except block is not executed

```
except Exception as e: print(e)
```

Ex :

```
a = 10
b = 0
try:
    c = a / b
    print(c)
except ZeroDivisionError as e:
    print("You cannot divide a number by zero")
```

==> **Output**

You cannot divide a number by zero

Or

```
import traceback

try:
    1/0
except Exception:
    traceback.print_exc()
```

Or

Instead of using the basic print() function, the more flexible [logging](#) module can be used to log the exception

```
import logging

try:
    1/0
except BaseException:
    logging.exception("An exception was thrown!")
```

Ex:

```
import logging

try:
    1/0
```

```
except BaseException:
    logging.exception("An exception was
thrown!")
```

### ==> Output

```
ERROR:root:An exception was thrown!
Traceback (most recent call last):
File ".../Desktop/test.py", line 4, in
<module>
    1/0
ZeroDivisionError: division by zero
```