

Errors

Reference:

- <https://docs.python.org/3/library/exceptions.html#builtin-exceptions>
- <https://docs.python.org/3/tutorial/errors.html>

Raising Errors

You can use the `raise` keyword to stop program execution if a certain condition is met.

```
options = ["rock", "paper", "scissors"]

choice = input("Please choose either 'rock', 'paper', or 'scissors': ")

if choice in options:
    print("YOU CHOSE", choice)
else:
    raise ValueError("OOPS – Please type 'rock', or 'paper', or 'scissors'
(without using using quotation marks).")
```

Defining and Raising Custom Errors

```
class MyCustomError(Exception):
    pass

raise MyCustomError("My custom message")
```

Handling Errors

You can use a `try... except` block to handle errors, whether encountered naturally or triggered intentionally:

```
try:
    raise RuntimeError("Hello")
    print("EVERYTHING IS GOING FINE")
except RuntimeError:
    print("OOPS – MY ERROR")
```

```
#> OOPS – MY ERROR
```

```
try:
    100 / 0
    print("EVERYTHING IS GOING FINE")
except DivisionByZeroError:
    print("OOPS - MY ERROR")
```

```
#> OOPS - MY ERROR
```