

Gestione degli errori

Informatica@DSS 2025/2026

Massimo Lauria <massimo.lauria@uniroma1.it>
<https://massimolauria.net/informatica2025/>

Un errore non è la fine del mondo

Un programma termina in caso di errori, dovuti a

- ▶ operazioni non valide
- ▶ errori sollevati esplicitamente

È tuttavia possibile per il programma

- ▶ *catturare* alcuni di questi errori
- ▶ decidere come gestirli
- ▶ proseguire con il programma

Sintassi

```
try:                                1
    istruzione1                      2
    istruzione2                      3
    istruzione3                      4
except NomeErrore:                  5
    gestione1                        6
    gestione2                        7
                                    8
seguito1                           9
seguito2                           10
```

1. il codice tra try e except viene eseguito
2. se c'è un errore di tipo **non** NomeErrore il programma si ferma (e quindi 3 e 4 non avvengono)
3. se c'è un errore di tipo NomeErrore viene eseguito il codice di gestione
4. viene eseguito il seguito

Esempio 1

```
1 def scontato(prezzo,sconto):  
2     if not ( 0 <= sconto <= 100 ):  
3         raise ValueError("Lo sconto non è tra zero e cento.")  
4  
5     return prezzo*(1.0 - sconto/100.0)  
6  
7 try:  
8     print( scontato(500,10) )  
9 except ValueError:  
10    print("1. Prova a mettere dei dati validi")  
11  
12 try:  
13    print( scontato(500,-10) )  
14 except ValueError:  
15    print("2. Prova a mettere dei dati validi")
```

```
450.0  
2. Prova a mettere dei dati validi
```

Esempio 2

```
def leggi_un_intero():
    while True:
        x = input("Inserisci un intero: ")
        try:
            value = int(x)
            return value
        except ValueError:
            print("Input non valido, riprova.")
```

1
2
3
4
5
6
7
8
9

Questa funzione chiede di inserire un valore intero, e riprova se l'input inserito non è valido.

Gestione di diversi tipi di errori

Potere gestire più errori con lo stesso codice di gestione

```
try: 1
    value = min(lista)
except (ValueError,TypeError): # parentesi necessarie 2
    print("Argomento non valido") 3
4
```

o con codici di gestione specifici

```
try: 1
    value = min(lista)
except ValueError: 2
    print("Lista vuota")
except TypeError: 3
    print("Lista che contiene valori di tipo diverso") 4
5
6
```

Proviamo lista=[] e lista=[1,"ciao"]

Errori non catturati

Se un tipo di errore non viene catturato, allora causa la terminazione del programma.

```
try: 1
    L = [10,4,7] 2
    L[5] = 100 3
except ValueError: 4
    print("Gestisco ValueError")
except TypeError: 5
    print("Gestisco TypeError") 6
    7
```

```
Traceback (most recent call last):
...
...
IndexError: list assignment index out of range
```

Try/Except nidificati

