

# Gestione degli errori

Informatica@DSS 2024/2025

Massimo Lauria <massimo.lauria@uniroma1.it>  
<https://massimolauria.net/informatica2024/>

# 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:
    istruzione1
    istruzione2
    istruzione3
except NomeErrore:
    gestione1
    gestione2

seguito1
seguito2
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
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

```
def scontato(prezzo,sconto):
    if not ( 0 <= sconto <= 100 ):
        raise ValueError("Lo sconto non è tra zero e cento.")

    return prezzo*(1.0 - sconto/100.0 )

try:
    print( scontato(500,10) )
except ValueError:
    print("1. Prova a mettere dei dati validi")

try:
    print( scontato(500,-10) )
except ValueError:
    print("2. Prova a mettere dei dati validi")
```

450.0

2. Prova a mettere dei dati validi

## Esempio 2

```
def leggi_un_intero(): 1
    while True: 2
        x = input("Inserisci un intero: ") 3
        try: 4
            value = int(x) 5
            return value 6
        except ValueError: 7
            print("Input non valido, riprova.") 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) 2
except (ValueError, TypeError): # parentesi necessarie 3
    print("Argomento non valido") 4
```

o con codici di gestione specifici

```
try: 1
    value = min(lista) 2
except ValueError: 3
    print("Lista vuota") 4
except TypeError: 5
    print("Lista che contiene valori di tipo diverso") 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") 5
except TypeError: 6
    print("Gestisco TypeError") 7
```

```
Traceback (most recent call last):
```

```
...
...
```

```
IndexError: list assignment index out of range
```

# Try/Except nidificati

