

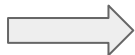
# Programming Lab

## Lezione 5

# Le eccezioni

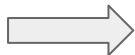
In Python gli errori si chiamano “eccezioni”:

```
1  
2 a = 'ciao'  
3 float(a)  
4
```



```
> python lezione5.py  
Traceback (most recent call last):  
  File "lezione5.py", line 3, in <module>  
    float(b)  
NameError: name 'b' is not defined
```

```
1  
2 a = 'ciao'  
3 float(b)  
4
```



```
> python lezione5.py  
Traceback (most recent call last):  
  File "lezione5.py", line 3, in <module>  
    float(a)  
ValueError: could not convert string to float: 'ciao'
```

# Cosa sono le eccezioni

Le eccezioni sono oggetti. Come tutto in Python.

→ *Estendono la classe base “Exception”*

Esempi:

- Exception

  - ArithmeticError

    - FloatingPointError

    - ZeroDivisionError

  - AttributeError

  - SyntaxError

  - NameError

  - TypeError

  - ValueError

# Il costrutto try-except

Permette di gestire le eccezioni (errori).

```
a = 'ciao'

try:
    float(a)
except:
    print('Non ho potuto convertire il parametro "a" a valore numerico!')
    print('Uso il valore di default "0" per il parametro "a"')
    a = 0

print('Qui il codice va avanti ad essere eseguito')
```

```
> python lezione5.py
Non ho potuto convertire il parametro "a" a valore numerico!
Uso il valore di default "0" per il parametro "a"
Qui il codice va avanti ad essere eseguito
```

# Il costrutto try-except

Posso anche gestire solo eccezioni specifiche:

```
try:
    float(a)
except ValueError:
    print('Ho avuto un errore di VALORE e non ho potuto convertire il parametro "a" a valore numerico!')
    print('Uso il valore di default "0" per il parametro "a"')
    a = 0
except TypeError:
    print('Ho avuto un errore di TIPO e non ho potuto convertire il parametro "a" a valore numerico!')
    print('Uso il valore di default "0" per il parametro "a"')
    a = 0
except:
    print('Ho avuto un errore SCONOSCIUTO mentre cercavo di convertire il parametro "a" a valore numerico, esco!')
    import sys
    sys.exit()
```

# Il costrutto try-except

...e posso avere l'eccezione dentro l' except:

```
try:
    float(a)
except Exception as e:
    print('Ho avuto un errore mentre cercavo di convertire il
    parametro "a" a valore numerico: "{}".format(e))
```

```
> python lezione5.py
```

```
Ho avuto un errore mentre cercavo di convertire il parametro "a" a valore
numerico: "could not convert string to float: 'ciao'"
```

# Il costrutto try-except

...e posso avere l'eccezione dentro l' except:

Per usare l'eccezione dentro l'except, devo sempre specificare che eccezione voglio gestire, se le voglio gestire tutte allora uso la classe base Exception

```
try:
    float(a)
except Exception as e:
    print('Ho avuto un errore mentre cercavo di convertire il
    parametro "a" a valore numerico: "{}".format(e))
```

```
> python lezione5.py
```

```
Ho avuto un errore mentre cercavo di convertire il parametro "a" a valore
numerico: "could not convert string to float: 'ciao'"
```

# Esercizio

Modificate l'oggetto **CSVFile** della lezione precedente in modo che dia un messaggio d'errore se si cerca di aprire un file non esistente.

Poi, aggiungete questi due campi al file “shampoo\_sales.csv”:

```
01-01-2015,  
01-02-2015,ciao
```

e gestite gli errori che verranno generati in modo che le linee vengano saltate ma che venga stampato a schermo l'errore

*Alla fine ricordatevi di committare tutto*