

ECCEZIONI

Se il programma si blocca a causa di un errore in esecuzione viene creata un'eccezione: l'interprete si ferma e mostra un messaggio d'errore.

Le eccezioni più comuni per i programmi che hai visto finora possono essere:

la divisione di un valore per zero:

```
>>> print 55/0
ZeroDivisionError: integer division or modulo
```

la richiesta di un elemento di una lista con un indice errato:

```
>>> a = []
>>> print a[5]
IndexError: list index out of range
```

la richiesta di una chiave non esistente in un dizionario:

```
>>> b = {}
>>> print b['pippo']
KeyError: pippo
```

In ogni caso il messaggio d'errore è composto di due parti: la categoria dell'errore e le specifiche separati dai due punti. Normalmente Python stampa la traccia del programma al momento dell'errore ma in questi esempi sarà omessa per questioni di leggibilità.

Molte operazioni possono generare errori in esecuzione ma in genere desideriamo che il programma non si blocchi quando questo avviene. La soluzione è quella di gestire l'eccezione usando **le istruzioni try ed except**.

Per fare un esempio possiamo chiedere ad un operatore di inserire il nome di un file per poi provare ad aprirlo. Se il file non dovesse esistere non vogliamo che il programma si blocchi mostrando un messaggio di errore; così cerchiamo di gestire questa possibile eccezione:

```
NomeFile = raw_input('Inserisci il nome del file: ')
try:
    f = open (NomeFile, "r")
except:
    print 'Il file', NomeFile, 'non esiste'
```

L'istruzione **try** esegue le istruzioni nel suo blocco. Se non si verificano eccezioni (e cioè se le istruzioni del blocco **try** sono eseguite senza errori) l'istruzione **except** ed il blocco corrispondente vengono saltate ed il flusso del programma prosegue dalla prima istruzione presente dopo il blocco **except**. Nel caso si verifichi qualche eccezione (nel nostro caso la più probabile è che il file richiesto non esiste) viene interrotto immediatamente il flusso del blocco **try** ed eseguito il blocco **except**.

Possiamo anche incapsulare questa capacità in una funzione: **FileEsiste** prende un nome di un file e ritorna vero se il file esiste, falso se non esiste.

```
def FileEsiste(NomeFile):
    try:
        f = open(NomeFile)
        f.close()
        return 1
    except:
        return 0
```

Esercizio

Modificare il programma sulla rubrica telefonica in modo che venga effettuato il controllo sull'esistenza del file da aprire e sull'esistenza del file da scrivere; quest'ultimo controllo è importante per chiedere la conferma prima di sovrascrivere un file.

Esercizio

Modificare il programma sulla rubrica telefonica in modo che venga effettuato il controllo sull'esistenza del file da aprire e sull'esistenza del file da scrivere; quest'ultimo controllo è importante per chiedere la conferma prima di sovrascrivere un file.

Per effettuare il controllo sull'esistenza del file da aprire modifichiamo la funzione `carica_numeri` come segue:

```
def carica_numeri(numbers, filename):
    try:
        in_file = open(filename, "r")
        while True:
            in_line = in_file.readline()
            if in_line == "":
                break
            in_line = in_line[:-1]
            [name, number] = string.split(in_line, ",")
            numbers[name] = number
        in_file.close()
    except:
        print "Il file ", filename, " non esiste"
```

Per effettuare il controllo sull'esistenza del file da salvare modifichiamo la funzione `salva_numeri` come segue:

```
def salva_numeri(numbers, filename):
    try:
        f=open(filename)
        f.close()
        #l'esecuzione del programma prosegue se il filename esiste
        # se filename non esiste il programma prosegue con except
        risposta="s"
        print "Il file", filename, " esiste già."
        risposta=raw_input("Vuoi sovrascriverlo(s/n)? ")
        if risposta=="s":
            out_file = open(filename, "w")
            for x in numbers.keys():
                out_file.write(x+", "+numbers[x)+"\n")
            out_file.close()
            print "Ho salvato le voci della rubrica nel file: ", filename
        else:
            filename = raw_input("Inserisci un altro nome per il file da salvare: ")
            salva_numeri(phone_list, filename)
    except:
        out_file = open(filename, "w")
        for x in numbers.keys():
            out_file.write(x+", "+numbers[x)+"\n")
        out_file.close()
        print "Ho salvato le voci della rubrica nel file: ", filename
```