

Complementi di Programmazione

Parsing Argomenti

CdL Informatica - Università degli studi di Modena e Reggio Emilia
AA 2023/2024

Filippo Muzzini

Parsing Argomenti

E' comune dover passare degli argomenti in input ad un programma.

```
./a.out primo_arg secondo_arg
```

```
python3 ./script.py primo_arg secondo_arg
```

Parsing Argomenti

In Python è possibile ricavare gli argomenti utilizzando il modulo **sys**

Una volta importato gli argomenti si trovano in

sys.argv

sottoforma di lista.

Parsing Argomenti

usare `sys.argv` implica parsare a mano tutti gli argomenti.

- Tenere conto della posizione
- Tenere conto del tipo di dato in ingresso
- Controllare il tipo di dato
- Oppure parsare eventuali = per l'assegnazione di un parametro

Parsing Argomenti

Il modulo **argparse** gestisce automaticamente questi aspetti.

Vi è una fase in cui si definisce cosa ci sia spetta come argomenti.

Da questa definizione il modulo

- genera un help
- parse gli argomenti nel modo corretto

Parsing Argomenti

Fase di definizione

```
import argparse
```

```
parser = argparse.ArgumentParser(description='Process some integers.')
```

Si crea un oggetto `ArgumentParser` con una descrizione generica del programma

Parsing Argomenti

Fase di definizione

```
parser.add_argument(dest='integers', metavar='N', type=int,  
nargs='*', help='an integer for the accumulator')
```

Con `add_argument` si aggiungono gli argomenti che ci si aspetta in input.

Parsing Argomenti

Fase di parsing

```
arguments = parser.parse_args()
```

Con `parse_args` si analizzano i parametri passati al programma da linea di comando.

Ritorna un oggetto Namespace con le variabili popolate

Parsing Argomenti

add_argument()

Versione base

```
parser.add_argument('name')
```

il primo parametro finirà in name

Serve per gestire i parametri posizionali

Tali parametri vengono considerati obbligatori

Parsing Argomenti

`add_argument()`

`parser.add_argument('-f', '--foo')`

Specifica il nome dell'argomento nella versione corta e lunga

Tali parametri sono considerati opzionali

Parsing Argomenti

`add_argument()`

parametri opzionali:

- `dest` -> specifica la variabile di destinazione (di default usa il nome del parametro)
- `type` -> tipo di dato (default stringa) che ci si aspetta in quel parametro
 - `argparse.FileType('r')` per i file
- `default` -> valore di default per i parametri opzionali
- `nargs` -> numero di valori da considerare per quell'argomento
 - `nargs=2` -> `-foo a b` -> `foo=['a','b']`
 - `*` per tutti i valori successivi (fino al nuovo argomento definito)

Parsing Argomenti

`add_argument()`

parametri opzionali:

- `action` -> azione da compiere sui valori dell'argomento
 - `store` -> default. salva il valore
 - `store_true/false` -> salva true o false a seconda se il parametro è stato specificato (o viceversa). Utilizzato per controllare argomenti flag.
 - `save_const` -> salva il valore definito in 'const'. Usato per come il precedente ma con un valore diverso da True se l'argomento è presente
 - `append` -> crea una lista se il parametro è specificato più volte
 - `count` -> conta il numero di volte che il parametro è specificato (es. `-vvv` -> `v=3`)

Parsing Argomenti

`add_argument()`

parametri opzionali:

- `help` -> specifica una stringa da mostrare nell'help per quell'argomento
- `metavar` -> valore di esempio da mostrare nell'help per quell'argomento