

Operazioni su file di testo: grep

Si considera il file "miofile" contenente un elenco di parole:

```
$ cat miofile
cr
ca
car
cor
caar
```

Le seguenti "wildcards" possono essere usate nelle *espressioni regolari* per la ricerca di parole nel file con `grep -E`

- * (+) : il carattere precedente compare zero (una) o più volte nel pattern

Esempio:

```
$ grep -E ca*r miofile
cr
car
caar
```

Operazioni su file di testo: grep

- . (?) : il carattere precedente compare (al più) una volta nel pattern

Esempio:

```
$ grep -E ca?r miofile
cr
car
```

- {n,m} : il carattere precedente almeno n e al più m volte nel pattern

Esempio:

```
$ grep -E 'ca{1,2}r' miofile
car
caar
```

- [] : l'espressione tra parentesi è l'unione delle espressioni contenute

Esempio:

```
$ grep -E 'ca[a-z]r' miofile
car
cor
```

Operazioni su file di testo: grep

- [str] : un qualunque carattere in *str*

Esempio:

```
$ grep 'c[ao]r' miofile
car
cor
```

- [a-z] : un qualunque carattere tra a e z

Esempio:

```
$ grep 'c[m-p]r' miofile
cor
```

- [^str] : un qualunque carattere **non** in *str*

Esempio:

```
$ grep 'c[^o]r' miofile
cr
ca
car
```

Operazioni su file di testo: grep

- . : un qualunque carattere

Esempio:

```
$ grep 'c.r' miofile
car
cor
```

- {n,m} : il carattere precedente almeno n e al più m volte

Esempio:

```
$ grep -E 'ca{1,2}r' miofile
car
caar
```

- ? : il carattere precedente compare zero o una volta

Esempio:

```
$ grep -E 'ca?r' miofile
cr
car
```

Operazioni su file di testo: grep

- () : l'espressione tra parentesi viene trattata come un carattere

Esempio:

```
$ grep -E 'c(aa)*r' miofile
cr
caar
```

- | : compare una delle due espressioni a destra e a sinistra del simbolo (o entrambe)

Esempio:

```
$ grep -E 'ca|or' miofile
ca
car
cor
caar
```

Esempio:

```
$ grep -E 'c(a|o)r' miofile
car
cor
```

Operazioni su file di testo: grep

- ^ : inizio riga
- \$: fine riga
- \< : inizio parola
- \> : fine parola

- Condideriamo ora il seguente file di esempio

```
quale
le stesse
questo file
tra le linee
```

Operazioni su file di testo: grep

- Righe che terminano con **le**
- Righe con parole terminanti in **le**
- Righe che contengono esattamente la parola **le**
- Righe che iniziano con la parola **le**

```
$ grep -E 'le$' miofile
quale
questo file
```

```
$ grep -E 'le\>' miofile
quale
le stesse
questo file
tra le linee
```

```
$ grep -E '\<le\>' miofile
le stesse
Tra le linee
```

```
$ grep -E '^le\>' miofile
le stesse
```

Operazioni su file di testo: grep

- Se si cerca una sequenza di caratteri in cui compare uno degli appena elencati "wildcards" (es. cerco la stringa "pippo.txt"), tale carattere deve essere preceduto da \ (es. cerco "pippo\\.txt") altrimenti sarà interpretato come "wildcard" e non come semplice carattere.
- La stessa cosa vale per il \ stesso, essendo anch'esso un carattere speciale.

metacarattere	tipo	significato
^	B	inizio della linea
\$	B	fine della linea
\<	B	inizio di una parola
\>	B	fine di una parola
.	B	un singolo carattere (qualsiasi)
[str]	B	un qualunque carattere in <i>str</i>
[^str]	B	un qualunque carattere non in <i>str</i>
[a-z]	B	un qualunque carattere tra a e z
\	B	inibisce l'interpretazione del metacarattere che segue
*	B	zero o più ripetizioni dell'elemento precedente
+	E	una o più ripetizioni dell'elemento precedente
?	E	zero od una ripetizione dell'elemento precedente
{j,k}	E	un numero di ripetizioni compreso tra j e k dell'elemento precedente
s t	E	l'elemento s oppure l'elemento t
(exp)	E	raggruppamento di <i>exp</i> come singolo elemento

dove B (basic) indica che la sequenza di caratteri è utilizzabile sia in `grep` che in `egrep`, mentre E (extended) indica che la sequenza di caratteri è utilizzabile solo in `egrep` (o in `grep` usando l'opzione `-E`).