

Expansioni di Bash

Il linguaggio **Bash** fornisce numerosi tipi di espansione, che permettono di trasformare o manipolare dati come stringhe, variabili e comandi. Ecco un elenco delle principali espansioni presenti:

1. Espansione delle variabili

- Consente di sostituire il nome di una variabile con il suo valore.

```
echo $VAR
```

- **Default Values:**

- `${VAR:-default}` : Usa il valore di `VAR` o un valore predefinito.
 - `${VAR:=default}` : Imposta `VAR` al valore predefinito se non è già definita.
 - `${VAR:+replacement}` : Usa `replacement` solo se `VAR` è definita.
 - `${VAR:?error_message}` : Mostra un errore se `VAR` non è definita.
-

2. Espansione dei comandi

- Sostituisce il comando con il suo output.

```
echo $(date)  
echo `ls`
```

3. Espansione dell'aritmetica

- Valuta espressioni matematiche.

```
echo $((2 + 3))  
echo $((VAR * 5))
```

4. Espansione del pathname (Globbing)

- Espande pattern in nomi di file o directory.

```
echo *.txt      # Tutti i file con estensione .txt
echo dir/*      # Tutti i file nella directory dir
```

5. Espansione delle parentesi graffe

- Genera un elenco di stringhe.

```
echo {A,B,C}    # Risultato: A B C
echo file{1..5} # Risultato: file1 file2 file3 file4 file5
```

6. Espansione della tilde

- Sostituisce `~` con la directory home dell'utente corrente o di uno specificato.

```
echo ~          # Directory home corrente
echo ~user      # Directory home di "user"
```

7. Espansione del parametro posizione

- Usa `$1`, `$2`, ecc., per accedere agli argomenti passati a uno script.

```
echo $1         # Primo argomento dello script
echo $@         # Tutti gli argomenti
```

15. Espansione delle variabili speciali

- Variabili incorporate:
 - `$?` : Stato di uscita dell'ultimo comando.
 - `$$` : ID del processo corrente.
 - `$!` : PID dell'ultimo processo in background.
 - `$@` e `$*` : Argomenti passati allo script.

Wildcards di base

1. * (Asterisco)

- Significato: Rappresenta zero o più caratteri.
- Esempio:
 - *.txt: Tutti i file con estensione .txt.
 - file*: File che iniziano con file.

2. ? (Punto interrogativo)

- Significato: Rappresenta un singolo carattere.
- Esempio:
 - file?.txt: File come file1.txt o fileA.txt (ma non file12.txt).

3. [] (Quadre)

- Significato: Rappresenta un singolo carattere che corrisponde a uno dei caratteri specificati all'interno delle parentesi.
- Esempio:
 - file[123].txt: File come file1.txt, file2.txt o file3.txt.
 - file[a-c].txt: File come filea.txt, fileb.txt, o filec.txt.

4. [^] o [!] (Negazione)

- Significato: Rappresenta un singolo carattere che **non** corrisponde ai caratteri specificati.
- Esempio:
 - file[^0-9].txt: File che non terminano con un numero (filea.txt, fileb.txt).

Altre cose

Mettendo \${!VAR} vuol dire prendi i VAR come se contenesse il nome della variabile ossia \${nomeVaribileDentroVar}