

## Exit status e output

Ogni programma o script è tenuto a restituire un **valore numerico** compreso tra 0 e 255, che comunica l'esito della sua esecuzione. Tale valore numerico prende il nome di **exit status**; questo non è da confondersi con **l'output del programma/script**: il primo un valore che determina se l'esito del programma è positivo o meno, mentre il secondo è cosa il programma effettivamente produce.

In C, l'exit status è generato dall'istruzione `return`; in `bash` è l'istruzione `exit` a generare tale valore.

Anche se i possibili valori variano da 0 a 255, ci sono alcuni valori fissati per alcune occasioni specifiche; `1` per errori generici, `126` per errori con permessi, `127` è `command not found`, ecc...

**È possibile ricavare l'exit value dell'ultimo comando eseguito** tramite la variabile d'ambiente `$?`.

```
ciao
# Output -> bash: ciao: command not found. L'exit code fissato è 127.
echo $?
# Output: 127 -> Il comando echo è andato a buon fine.
echo $?
# Output: 0 -> l'echo di prima è andato a buon fine!
```

L'applicazione più eclatante degli exit status sta nella **valutazione aritmetica delle espressioni**; effettuando operazioni tramite `$(( ))` o `(( ))` verrà modificato il valore di `$?` come segue:

- `$? = 1` se l'espressione non è valutabile aritmeticamente (o se avviene un errore).
- `$? = 0` se l'espressione è valutata logicamente come *true*; `$? = 1` se è *false*.
- `$? = 0` se l'espressione fornisce un risultato intero **diverso da 0**; `$? = 1` altrimenti.
- Viene valutato solo il risultato dell'ultima operazione eseguita, nel caso di `()` annidati.

## Liste di comandi

Una lista di comandi è un insieme di comandi da lanciare in successione. Ogni comando (o generalmente elemento) va **separato dal carattere punto e virgola** `;`.

Gli elementi della lista possono essere di vario tipo:

- Comandi semplici: `cd`, `ls`, `./script.sh`, ecc...
- Espressioni valutate aritmeticamente: `(( VAR=(2*3)+($VAR+2) ))`

- Sequenza di comandi separati da *pipe* | : `cat file.txt | grep ciao`
- Sequenza di comandi condizionali: `gcc file.c && ./file.exe`
- Raggruppamento di comandi: `( cat file1.c ; cat file2.c )`
- Espressione condizionale: `[[ -e file.txt && $1 -gt 13 ]]`

L'*exit status* di una lista di comandi è **determinato dall'ultimo comando della lista che viene eseguito**. Non è sempre detto che l'ultimo comando della lista sia quello che restituisce l'*exit status*, in quanto potrebbe avvenire che i risultati di comandi precedenti dipendano dall'esito di altri comandi precedenti, tutti quanti possono fallire.

Le sequenze di comandi possono essere **condizionali o non condizionali** in base ai metacaratteri usati in concatenazione:

- Le sequenze *non condizionali* sono sequenze di comandi separati dal metacarattere `;`. Vengono eseguiti in sequenza senza controllare l'esito dei comandi precedenti.
- Le sequenze *condizionali* sono sequenze di comandi che vengono eseguiti sulla base dell'*exit code* del/dei comandi precedenti;
  - Se separati da **doppia pipe** `||`, il comando successivo a queste ultime viene eseguito se il primo termina con un *exit code* **diverso da 0**.
  - Se separati da **doppio and** `&&`, il comando successivo a questi ultimi viene eseguito se il comando precedente termina con un *exit code* **uguale a 0**.