**BASH SHELL**

La **Bash shell** (Bourne Again Shell) è il **software** che ci consente di eseguire il codice di linux, ed è in grado di interpretare i comandi permettendoci di gestire un intero sistema operativo.

**Bash** è una **"Unix shell"**, vale a dire un'**interfaccia a linea** (o riga) di comando che permette l'interazione con sistemi operativi derivati da Unix; è stata utilizzata su molti altri sistemi, non necessariamente derivati da Unix. È stata creata verso la fine degli anni '80 da Brian Fox.

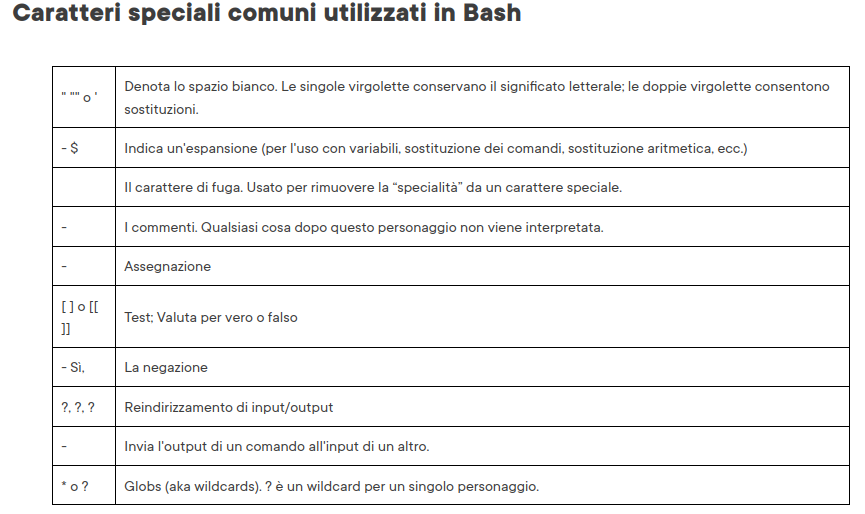
Lo scopo era quello di fornire un software libero alternativo alla **Bourne shell**, in precedenza una delle principali shell per i sistemi GNU/Linux, incorporandone tutte le caratteristiche assieme ad alcune valide aggiunte, come l'aritmetica intera e la valutazione delle espressioni regolari.

Bash può eseguire comandi con una risposta in tempo reale ed eseguire interi script conosciuti come "Bash shell script".

In definitiva, un Bash Shell script è un programma scritto nel linguaggio di programmazione Bash.

I Bash script vengono comunemente usati per molte **attività di amministrazione di sistema**, come l'analisi dei log, il backup dei dischi, e così via. Sono anche molto usati come script di installazione di programmi complessi. Insomma, la conoscenza pratica dello Shell Scripting è essenziale per tutti coloro che desiderano diventare degli amministratori di sistema esperti.

Qui di seguito c’è una tabella con la sintassi dei comandi principali utilizzati nello scripting nella Bash Shell, Bash offre un sacco di controlli e confronti integrati, arrivando abbastanza utili in molte situazioni.

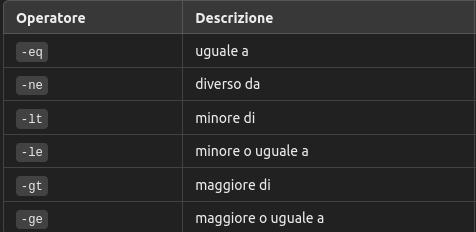


|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Quando inizi a scrivere e utilizzare le tue condizioni, ci sono alcune regole che dovresti sapere per evitare di ottenere errori difficili da rintracciare. Qui seguiamo tre importanti:

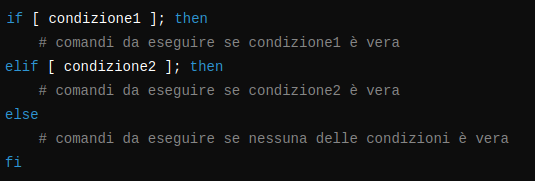
1. **Spazi:** Utilizza spazi adeguati all'interno delle strutture di controllo (come if, for, while) per separare le parole chiave e migliorare la leggibilità.
2. **Una linea per ogni parola chiave:** Le parole chiave come if, then, else, elif e fi devono sempre iniziare una nuova riga. Questo rende il codice più leggibile e evita errori di sintassi. Se hai bisogno di concatenare più istruzioni su una stessa riga, utilizza il carattere ; per separarle.
3. **Citazione delle variabili:** Quando utilizzi variabili di tipo stringa all'interno di condizioni o comandi, è sempre consigliabile racchiuderle all'interno di apici singoli (') o doppi ("). Questo previene problemi causati da spazi o caratteri speciali all'interno

Per confrontare le variabili vengono utilizzati diversi operatori, questi sono elencati nella tabella:

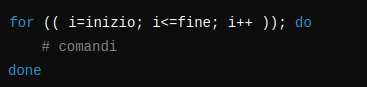


oltre al confronto delle variabili si può creare ed utilizzare anche strutture logiche come if, for, while…

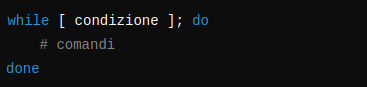
**if:**

****

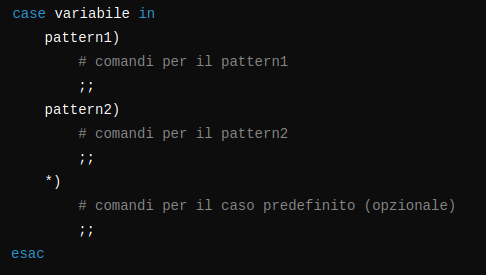
**for:**

****

**while:**



**switch:**

****