La Shell

(man bash)

Quando scriviamo un comando nella shell, questa prima di eseguirlo fa una serie di operazioni:

1) ***Brace Expansion***
2) ***Tilde Expansion***
3) ***Variable Expansion***
4) ***Arithmetic Expansion***
5) ***Comman Substitution***
6) ***Word Splitting***
7) ***Pathname Expansion***

Brace Expansion
La shell interpreta una certa sintassi e la sostituisce con un risultato.

Ad es. le parentesi graffe distribuiscono quello che c'è nelle parentesi su quello che c'è fuori:
nimloth@nimloth-mac:~$ echo a{1,2,3}
a1 a2 a3
nimloth@nimloth-mac:~$ echo {1,2,3}a
1a 2a 3a
nimloth@nimloth-mac:~$ echo a{1,2,3}b
a1b a2b a3b

Se inserisco uno spazio la shell non lo riconosce come il carettere spazio e quindi non effettua la distribuzione:
nimloth@nimloth-mac:~$ echo a{1,2, 3}b
a{1,2, 3}b
Per far sì che riconosca lo spazio vuoto ci vuole il carattere \
nimloth@nimloth-mac:~$ echo a{1,2,\ 3}b
a1b a2b a 3b

Tilde Expansion
Con il carattere tilde la shell indica la home di un utente.

Se richiamo la tilde da sola ottengo la home dell'utente che ha effettuato il comando:
nimloth@nimloth-mac:~$ echo ~
/home/nimloth

Se specifico dopo la tilde il nome di un utente allora ottengo il percorso della home dell'utente specificato. (La shell risale a questa informazione leggendo nel file /etc/passwd, che infatti non è protetto in lettura).
nimloth@nimloth-mac:~$ echo ~root
/root

Se scrivo una stringa sbagliata e quindi c'è un errore non ottengo niente:
nimloth@nimloth-mac:~$ echo ~rook
~rook

Questo esempio fa capire che la *Tilde Expansion* avviene dopo la *Brace Expansion*:
nimloth@nimloth-mac:~$ echo ~roo{t,k}
/root ~rook

Variable Expansion
Se dichiariamo delle variabili e poi le utilizziamo la shell traduce il nome della variabile sostituendogli il suo valore.

nimloth@nimloth-mac:~$ pippo1=ciao
nimloth@nimloth-mac:~$ pippo2=bello
nimloth@nimloth-mac:~$ echo $pippo1
ciao
nimloth@nimloth-mac:~$ echo $pippo2
bello
nimloth@nimloth-mac:~$ echo $pippo{1,2}
ciao bello

Possono essere inizializzati come variabili anche dei comandi:
nimloth@nimloth-mac:~$ comando=ls
nimloth@nimloth-mac:~$ $comando
amsn_received Documenti Immagini Projects Scrivania Video Examples Informatica Grafica

Dopo la variabile che usiamo come comando possiamo mettere i parametri del comando scelto!
nimloth@nimloth-mac:~$ $command -l
totale 292
drwx------ 2 nimloth nimloth 4096 2008-02-25 22:39 amsn_received
drwxr-xr-x 2 nimloth nimloth 4096 2008-02-27 19:02 Documenti
drwxr-xr-x 2 nimloth nimloth 4096 2008-02-27 19:02 Immagini
drwxr-xr-x 2 nimloth nimloth 4096 2008-03-04 12:37 Informatica Grafica

Arithmetic Expansion

Il discorso fatto per le variabili vale anche se i valori sono numerici. Si possono inoltre fare operazioni aritmetiche con una particolare notazione di parentesi.

nimloth@nimloth-mac:~$ pippo=5
nimloth@nimloth-mac:~$ echo $(($pippo+2))
7
nimloth@nimloth-mac:~$ echo $((5+2))
7
nimloth@nimloth-mac:~$ echo $pippo
5
nimloth@nimloth-mac:~$ pippo=$(($pippo+1))
nimloth@nimloth-mac:~$ echo $pippo
6

Quando la shell non conosce il valore di una variabile (perchè questa non è stata inizializzata) allora la considera di valore pari a zero.
nimloth@nimloth-mac:~$ echo $(($pluto+2))
2

Command Substitution

La shell può creare dei comandi a partire dall'output di altri programmi.

Il comando hostname ritorna il nome della macchina:
nimloth@nimloth-mac:~$ hostname
nimloth-mac

Touch crea un file vuoto:
(in realtà touch aggiorna la data di un file, ma quando questo file non esiste per aggiornargli la data deve anche crearlo, quindi si usa spesso per creare file)
nimloth@nimloth-mac:~$ touch prova
nimloth@nimloth-mac:~$ ls -l
-rw-r--r-- 1 nimloth nimloth 0 2008-03-04 16:21 prova

Con backquote diamo come parametro a touch l'output di hostname
nimloth@nimloth-mac:~$ echo touch `hostname`
touch nimloth-mac
nimloth@nimloth-mac:~$ touch `hostname`
nimloth@nimloth-mac:~$ ls -l
totale 292
-rw-r--r-- 1 nimloth nimloth 0 2008-03-04 16:23 nimloth-mac
-rw-r--r-- 1 nimloth nimloth 0 2008-03-04 16:21 prova

Word Splitting

La shell deve prendere ciò che è scritto sulla linea di comando e dopo averlo elaborato secondo i passi precedenti deve separare le varie voci utilizzando lo spazio come separatore di campo.

nimloth@nimloth-mac:~$ echo touch `hostname`

Dopo le traduzioni e sostituzioni la shell sa che deve eseguire *echo touch nimloth-mac* quindi divide questa stringa considerando che la prima voce è il comando da eseguire e le altre voci son parametri.

Gli spazi non sono considerati parametri:
nimloth@nimloth-mac:~$ echo a b c
a b c

Pathname Expansion

In quest'ultima operazione la shell risolve la traduzione di caratteri speciali come * e *?*

nimloth@nimloth-mac:~$ ls -l
totale 296
-rw-r--r-- 1 nimloth nimloth 30603 2008-02-26 00:04 alita.JPG
drwx------ 2 nimloth nimloth 4096 2008-03-04 17:51 amsn_received
-rw-r--r-- 1 nimloth nimloth 0 2008-03-04 18:08 a.txt
-rw-r--r-- 1 nimloth nimloth 0 2008-03-04 18:08 b.txt
-rw-r--r-- 1 nimloth nimloth 0 2008-03-04 18:08 c.txt
drwxr-xr-x 2 nimloth nimloth 4096 2008-03-04 18:13 copia
-rw-r--r-- 1 nimloth nimloth 0 2008-03-04 18:08 d.txt
-rw-r--r-- 1 nimloth nimloth 0 2008-03-04 18:09 e.txt
lrwxrwxrwx 1 nimloth nimloth 26 2008-02-25 13:18 Examples -> /usr/share/example-content
-rw-r--r-- 1 nimloth nimloth 0 2008-03-04 18:09 f.txt
drwxr-xr-x 4 nimloth nimloth 4096 2008-03-03 15:25 workspace

Con l'espressione *.txt seleziono tutti i file che hanno estensione txt:
*nimloth@nimloth-mac:~$ ls -l *.txt*
-rw-r--r-- 1 nimloth nimloth 0 2008-03-04 18:08 a.txt
-rw-r--r-- 1 nimloth nimloth 0 2008-03-04 18:08 b.txt
-rw-r--r-- 1 nimloth nimloth 0 2008-03-04 18:08 c.txt
-rw-r--r-- 1 nimloth nimloth 0 2008-03-04 18:08 d.txt
-rw-r--r-- 1 nimloth nimloth 0 2008-03-04 18:09 e.txt
-rw-r--r-- 1 nimloth nimloth 0 2008-03-04 18:09 f.txt

Spostiamo tutti i file txt nella cartella di nome Copia :
*nimloth@nimloth-mac:~$ mv *.txt copia*
*nimloth@nimloth-mac:~$ ls -l *.txt*
*ls: *.txt: Nessun file o directory*
nimloth@nimloth-mac:~$ cd copia/
nimloth@nimloth-mac:~/copia$ ls
a.txt b.txt c.txt d.txt e.txt f.txt

Abbiamo visto che i caratteri speciali vengono interpretati dalla shell secondo determinate regole e quindi non sono riconosciuti come semplici caratteri.

*nimloth@nimloth-mac:~$ echo **
20089225 alita.JPG amsn_received Desktop Documenti Examples Immagini Modelli Musica plasm-5 plasm6 Progetto Projects prova Pubblici Scrivania Unnamed0.psm Video wep workspace

Se vogliamo utilizzarli come caratteri puri dobbiamo farli precedere dal carattere \ oppure racchiuderli tra singoli apici. La differenza è che il carattere \ funziona con un solo carattere, mentre tra singoli apici possiamo mettere stringhe

*nimloth@nimloth-mac:~$ echo **
**
nimloth@nimloth-mac:~$ echo ' ?'*
*** ?**

Tra apici gli spazi vengono riconosciuti come caratteri*!*
nimloth@nimloth-mac:~$ echo 'a b'
a b
nimloth@nimloth-mac:~$ echo 'a b'
a b

Dentro doppi apici la shell può effettuare anche la sostituizione delle variabili:
nimloth@nimloth-mac:~$ echo 'a b $pippo'
a b $pippo
nimloth@nimloth-mac:~$ echo "a b $pippo"
a b 6

I caratteri speciali più utilizzati sono:
nimloth@nimloth-mac:~$ echo ' ? & ; \ $'*
*** ? & ; \ $**