

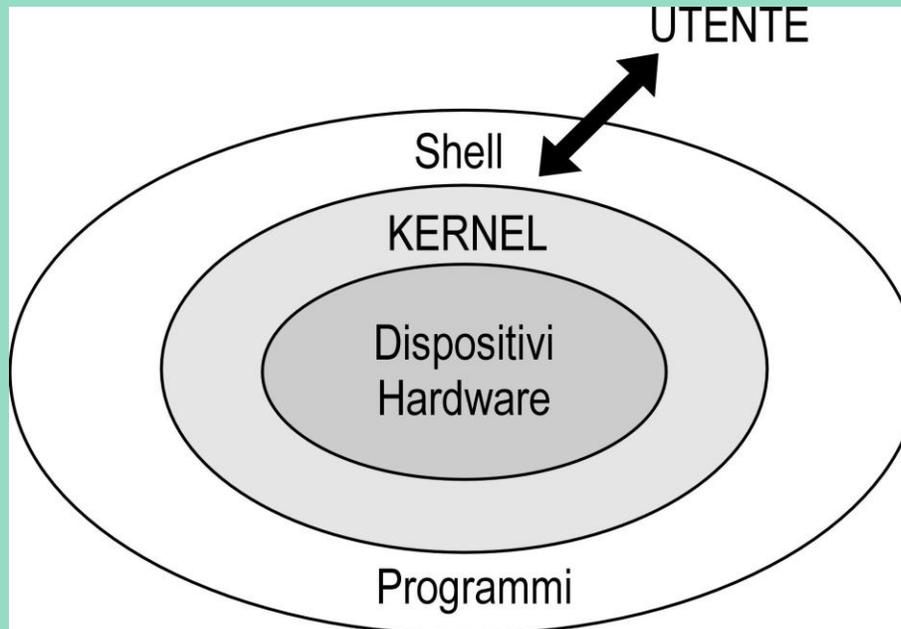
MS-DOS

shell
batch file
wildcard chars

Con il termine shell si intende l'interprete dei comandi del sistema operativo.

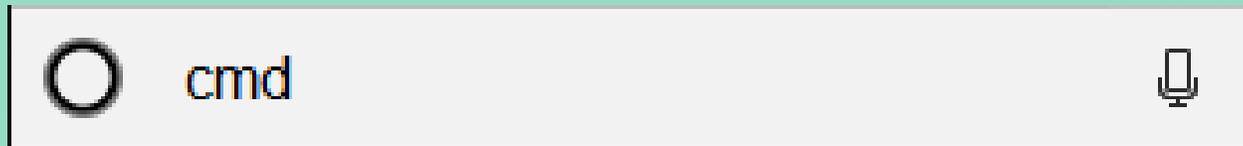
La shell è spesso testuale, ovvero i comandi vengono emessi digitando dei caratteri sulla tastiera (piuttosto che utilizzare il mouse in un ambiente grafico).

Il termine shell fa riferimento al guscio; spesso infatti il sistema operativo viene rappresentato con una serie di gusci annidati.



Per attivare la shell del DOS in Windows10 bisogna lanciare “command prompt” dal menu: Start → Sistema Windows → command prompt

Oppure digitare “cmd” nella finestrella di Cortana:



Nota: facendo click col pulsante destro sul comando e cliccando su “Altro” si può selezionare la modalità “Esegui come amministratore” si ottiene la shell col massimo dei diritti (su file e comandi)

La sintassi generale di un comando DOS è:

COMANDO parametri opzioni ridirezione

- Non tutti i comandi hanno **parametri**; ad esempio il comando di cancellazione dello schermo CLS è senza parametri
- Non tutti i comandi hanno delle **opzioni**; queste possono anche essere omesse; le opzioni sono precedute da '-' (dash) o da '/' (slash)
- Se inseriamo un comando che non esiste (p.e.: 'jcopy') l'interprete emetterà un messaggio d'errore, tipo: "jcopy" non è riconosciuto come comando interno o esterno, come programma eseguibile o file batch
- la **ridirezione** sarà affrontata più avanti

Un comando DOS viene eseguito, cioè è inviato al sistema operativo, solo dopo aver battuto il tasto INVIO (altrimenti detto ENTER)

Il DOS non fa distinzione tra lettere maiuscole e minuscole; si dice che non è **case-sensitive** dove “case” richiama uppercase (maiuscolo) e lowercase (minuscolo)

Dopo ogni comando DOS, prima di parametri e opzioni si deve lasciare almeno uno spazio vuoto (**blank**)

I comandi DOS vengono suddivisi in due grandi categorie:

Comandi interni: si trovano nel file *Command.com* e vengono caricati in memoria all'avvio del sistema

Comandi esterni: si tratta di file che risiedono sul disco in una directory particolare, e hanno estensione *.exe*, *.com*, *.bat*

Un file **batch** è un file di testo con estensione **.BAT** che contiene una sequenza di comandi DOS, uno per riga; viene usato per produrre degli “script” (sequenze di comandi) ripetibili.

Se si utilizza un file batch una sola volta si spreca tempo.

Nei file batch ogni riga/comando viene eseguita in modo sequenziale.

Un esempio di file batch in ambiente DOS è il file **AUTOEXEC.BAT** che viene eseguito automaticamente all'avvio di MS-DOS e serve per caricare driver e inizializzare variabili di ambiente utili ai vari dispositivi presenti nel sistema (ad esempio tastiera, mouse, scheda audio, stampante e directory temporanee).

Per poter scrivere un file batch si può utilizzare qualsiasi editor di testo come il blocco note (“notepad”) avviabile dalla lista dei programmi (da “Start”) oppure direttamente con il comando **notepad** dal prompt, ovvero dalla linea dei comandi (CLI)

I comandi utilizzati nei file batch sono quelli standard di MS-DOS; inoltre si può utilizzare:

ECHO *messaggio da visualizzare*

che consente di visualizzare “messaggio” (o qualunque altra stringa) sullo schermo.

Esempio:

ECHO *Salve. Questo programma serve a ...*

ECHO OFF : dalla riga in cui è inserito in poi, mostra solo gli output dei comandi (quindi non fa vedere la riga con il comando)

@ECHO OFF : in questo modo non mostra neppure ECHO OFF; in generale la chiocciola @ evita la visualizzazione del singolo comando

ECHO ON : ripristina la condizione standard in cui tutti i comandi vengono visualizzati

ECHO . : visualizza una riga vuota; ‘.’ deve essere attaccato a ECHO

REM commento

che consente di inserire commenti ai comandi inseriti nel file batch. I commenti, compresi nella programmazione, non vengono mai visualizzati e servono al programmatore per documentare il proprio operato.

:segnaposto (chiamato spesso “etichetta”)

Marca una linea dello script; segnaposto è un nome a caso, scelto dal programmatore. Come prassi di programmazione, a ogni identificatore (segnaposto) bisogna dare un nome significativo. Si usa insieme a:

GOTO segnaposto

PAUSE sospende l'esecuzione del file batch e visualizza: “Premere un tasto per continuare...”; molto utilizzato

COMANDO SET

SET: consente di impostare e inizializzare una variabile d'ambiente (environment variable)

```
C:\> SET NOME=Guido
```

← impostazione della variabile

```
C:\> ECHO Il nome e': %NOME%
```

← utilizzo della variabile

In MSDOS il nome delle variabili non è case-sensitive

Possiamo emettere il comando **SET** con l'opzione **/P** che consente l'input da parte dell'utente; per esempio:

```
SET /P NOME=Qual è il tuo nome?
```

PARAMETRI DALLA LINEA DI COMANDO

Sono a disposizione delle variabili predefinite, indicate con **%1**, **%2** ... **%9** che contengono i parametri passati dalla linea di comando; si possono passare un max di 9 parametri.

Per esempio, digitando:

```
C:> myBatch 7 ABC 65
```

si invoca il file batch con i parametri 7, ABC e 65

Si avrà che: nella variabile **%1** sarà contenuto il numero 7, nella variabile **%2** la stringa ABC, nella variabile **%3** il numero 65;

queste variabili possono essere utilizzate all'interno del file batch. Inoltre:

%0 contiene il nome del file batch (**myBatch**)

%* contiene la stringa con tutti i parametri invocati (**7 ABC 65**)

COMANDO IF

IF consente di emettere un comando solo quando si verifica una certa condizione:

Sintassi: **IF condizione comando**

condizione può essere:

[NOT] stringa1 == stringa2 : confronta le due stringhe; possono anche essere parametri (tipo: %1, %2 ecc.); tipico utilizzo: (non c'è il 1° parametro)

[NOT] EXIST filename : verifica l'esistenza di un file

[NOT] ERRORLEVEL n° : verifica l'esistenza di un certo tipo di errore

il NOT è facoltativo, per questo è indicato con le parentesi []

Esempi:

- IF "%1" == "" (per verificare se il parametro %1 è stato emesso)
- IF NOT EXIST %1 ECHO File inesistente
- IF %2 == "all" ECHO Attenzione: il comando agirà su tutti i file!!!

COMANDO FOR

Il comando **FOR** consente di ripetere un comando, anche con parametri. Bisognerebbe chiamarlo **FOR...IN...DO...** perché compaiono sempre insieme.

Sintassi:

FOR %%V **IN** (*lista*) **DO** *comando*

(*lista*) è un insieme qualsiasi di elementi separati da spazio, virgola o punto e virgola; *comando* rappresenta un comando che verrà eseguito tante volte quanti sono gli elementi di *lista*, ogni volta considerando come %%V un valore presente in *lista*, in modo sequenziale.

Normalmente %%V compare come parametro di *comando*

Esempio: cancellare i file con estensione: chi fis sci mat tlc

```
FOR %%A IN (chi fis sci mat tlc) DO DEL *.*%%A
```

COMANDO FIND

FIND consente la ricerca di una stringa all'interno di un file e restituisce le righe in cui è stata trovata

SINTASSI:

FIND "Stringa da cercare" [Percorso e nome del file]

Esempio:

FIND "guido" file.txt

FIND "guido" ..\Documenti\file.txt

La ricerca della stringa è case-sensitive, per cui le lettere maiuscole e le lettere minuscole vengono considerate diverse tra loro

E' possibile inserire ulteriori opzioni di ricerca con:

/V per mostrare tutte le righe che non contengono la stringa cercata;

/C visualizza il numero totale di righe in cui è stata trovata la stringa;

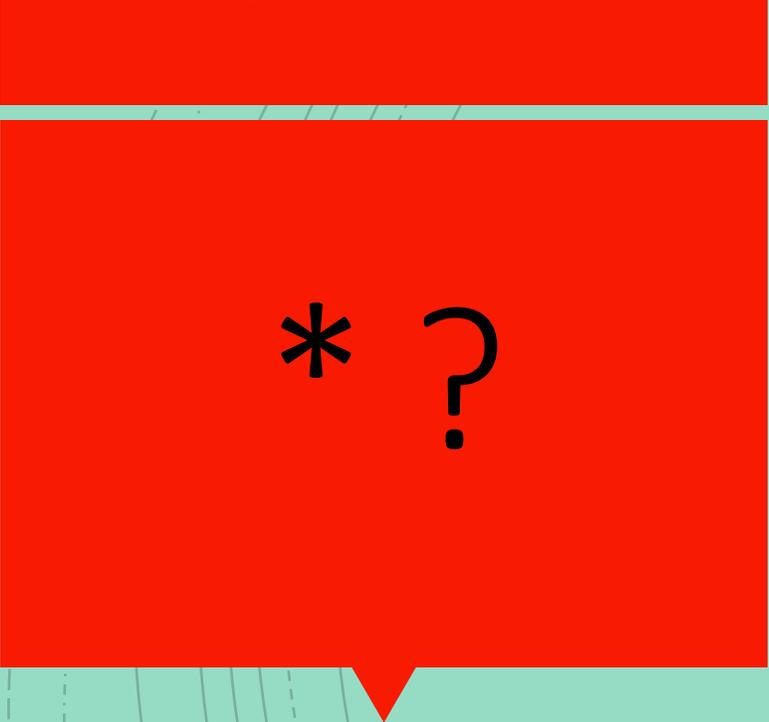
/N visualizza il numero delle righe in cui non è stata trovata la stringa;

/I serve per ignorare il case-sensitive dei caratteri;

CARATTERI WILDCARD

* ?

Ricorda: MS-DOS prevede per il nome del file
un formato 8.3



* ?

- **carattere jolly o wildcard character** è un **carattere** che, all'interno di una stringa (es. di ricerca), non rappresenta se stesso bensì sequenze di uno o più **caratteri**
- Se ad esempio dobbiamo ricercare un file del quale conosciamo il nome (myDoc) ma non ricordiamo l'**estensione** possiamo digitare:
- `C:\> dir myDoc.*`

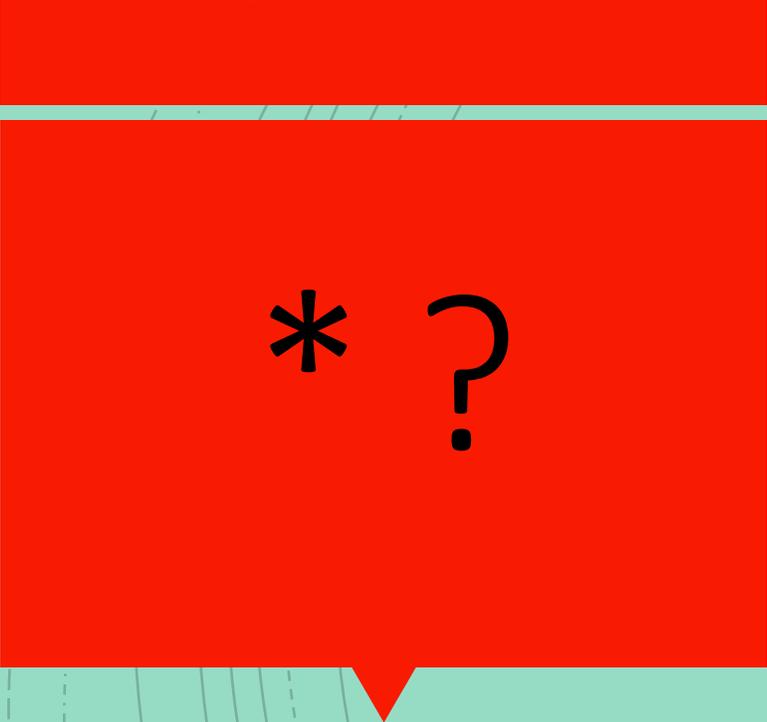
In questo modo la ricerca darà per risultato tutti i file il cui nome inizia con myDoc; per esempio:

- myDoc.txt
- myDoc.doc
- myDoc.png
- myDoc

Nel caso conoscessimo solo una parte del nome del file (es. "my") la sintassi sarebbe:

- C:\> dir my*

L'output del comando comprenderà tutti i file il cui nome&estensione inizia con la stringa "my"



* ?

L'asterisco può essere posizionato anche davanti alla stringa di ricerca:

- *.txt cerca tutti i file con estensione txt

Digitando al prompt di sistema la seguente stringa:

- C:\mydir> dir *Doc*

cerca tutti i file il cui nome contiene la stringa "Doc"

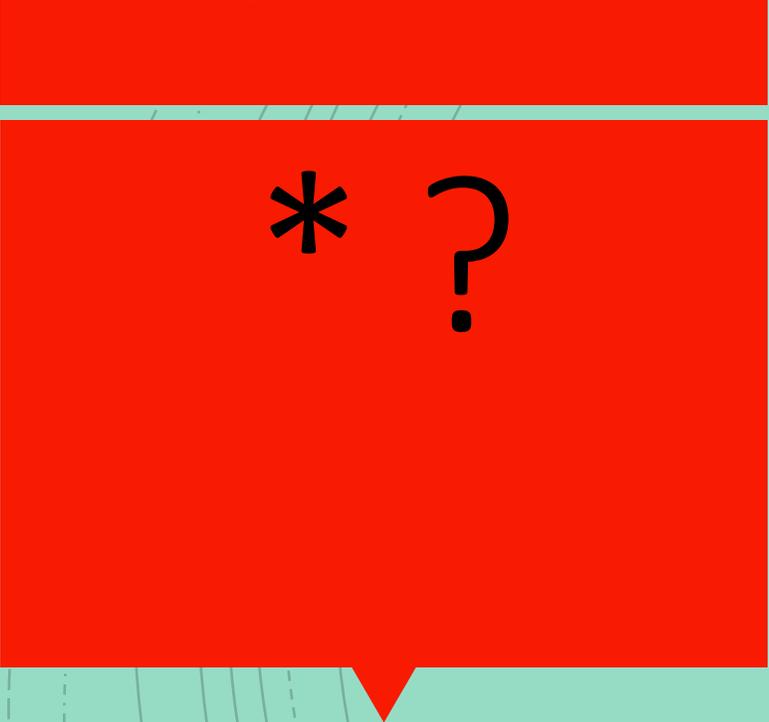
quindi ad esempio:

myDoc2.doc,

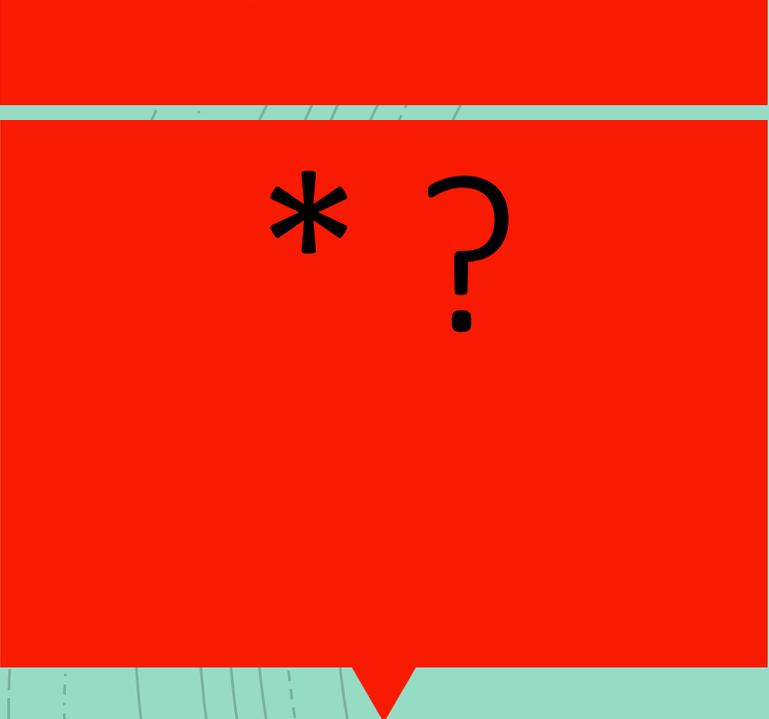
Doc_1.jpg

myDoc.png

ovviamente se presenti nella directory di ricerca

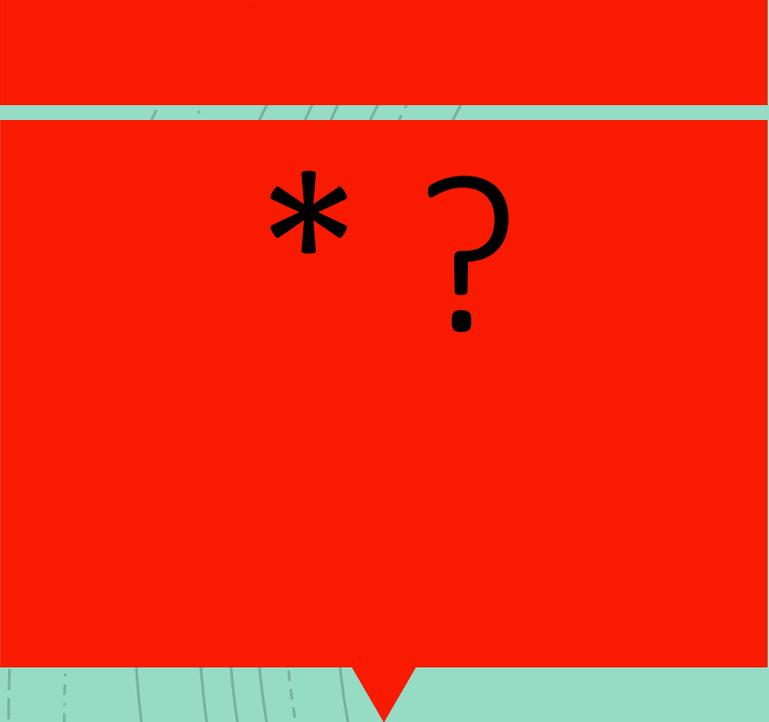


* ?



* ?

- In sintesi: l'asterisco (star in inglese) sostituisce una **qualunque stringa** nel posto dove è collocato, anche una stringa vuota; si può inserire anche più di una volta (es.: *.*) all'interno della stringa da definire.



* ?

- Il punto interrogativo può essere utilizzato allo stesso modo dell'asterisco, ma sostituisce un **solo carattere** della stringa di ricerca.

Ad esempio :

```
C:\> dir myDoc?.doc
```

potrebbe dare come risultato:

```
myDoc1.doc
```

```
myDocA.doc
```

ecc.