

# Linguaggio di programmazione Batch

Il linguaggio di programmazione Batch è molto semplice e non occorre installare software applicativi per sviluppare.

Questo linguaggio non permette di fare grandi cose, è possibile creare programmi con i quali l'utente può interagire testualmente.

Per creare un file batch occorre il blocco note di Windows e finito di scrivere si salva con nome il file selezionando Tutti i file (\*.\*) e scrivendo .bat alla fine del nome.

## -1 Comandi

*Nel linguaggio batch sono disponibili tutti quei comandi che si possono usare sul cmd di Windows, in più ce ne sono altri che permettono al programma di interagire meglio con chi lo usa.*

### I comandi principali del cmd.

- **systeminfo** = visualizza le caratteristiche del sistema
- **color** = modifica momentaneamente i colori dell'interfaccia
- **ipconfig /all** = visualizza la configurazione IP del sistema
- **ping** = invia pacchetti ad un server e attende risposta
- **defrag** = riorganizza i files in un disco specificato in seguito
- **mkdir** o **md** = crea una cartella nel percorso specificato in seguito
- **calc** = avvia la calcolatrice
- **explorer** = avvia Esplora File
- **notepad** = avvia Blocco Note
- **tasklist** = visualizza tutti i processi attivi
- **start** = avvia una nuova finestra cmd o un programma specificato in seguito
- **help** = visualizza tutti i comandi cmd disponibili
- **exit** = arresta il cmd o il programma
- **shutdown** = arresta il sistema

*Tutti questi comandi possono essere inseriti in un file batch: il comando start, però, ha più di una funzione. Con start è possibile eseguire:*

- **cmd** = scrivendo solamente "start"
- **programma** = scrivendo "start nomeprogramma"
- **cartella** = scrivendo "start U:\Percorso\Cartella"
- **indirizzo web** = scrivendo "start [www.indirizzoweb.it/pagina](http://www.indirizzoweb.it/pagina)"

### I nomi dei programmi.

*Per poter eseguire un programma con il comando start, occorre conoscere o avere a portata di occhiata i principali nomi dei programmi. Eccoli qua:*

- **cmd** = esegue cmd o Windows Command Prompt
- **explorer** = esegue Esplora File

- **calc** = esegue Calcolatrice
- **notepad** = esegue Blocco Note
- **winword** = esegue Word
- **mspaint** = esegue Paint
- **osk** = esegue Tastiera su Schermo
- **control** = esegue Pannello di Controllo
- **taskmgr** = esegue Gestione Attività
- **msinfo32** = esegue System Information

Conoscendo tutti questi comandi non resta che imparare la sintassi del linguaggio Batch.

## -2 Sintassi

*La sintassi del linguaggio Batch non è complicata e si impara molto velocemente. È molto "elastico" e in alcuni casi si possono anche evitare gli spazi, ma andando a capo più di una volta e usando gli spazi il codice risulterà più comprensibile per lo sviluppatore.*

### All'inizio.

All'inizio del codice di un programma va scritto come si vuole l'interfaccia. Nella prima riga va sempre @echo off per nascondere il percorso in cui si sta lavorando e dopo essere andati a capo si sceglie il nome del programma specificato in seguito a title. Per scegliere i colori della scritta e dello sfondo va usato color.

```
@echo off
title Programma Batch
color F9
```

*Color è molto semplice da usare: dopo aver scritto color va messo un codice di due caratteri che specifica di quali colori si tratta. Eccoli qua:*

- **0** = Nero
- **1** = Blu scuro
- **2** = Verde
- **3** = Verde acqua
- **4** = Bordeaux
- **5** = Viola
- **6** = Verde oliva
- **7** = Grigio chiaro
- **8** = Grigio
- **9** = Blu
- **A** = Verde limone
- **B** = Azzurro
- **C** = Rosso
- **D** = Fucsia
- **E** = Giallo

- F = Bianco

## Corpo del programma.

Ora si comincia a sviluppare la parte più importante del programma, lo sviluppo. Il corpo del programma è diviso in paragrafi che iniziano con un'etichetta. Se volessimo far leggere all'utente un testo, a questo punto dovremmo usare echo, seguito dal testo da vedere.

```
@echo off
title Ciao
color F9
```

```
:inizio
echo Ciao! Come stai?
pause>nul
```

*In questo modo il programma si chiamerà "Ciao" e l'utente vedrà la scritta "Ciao! Come stai?". Dopo aver scritto il messaggio **va messo pause>nul per evitare che il programma si chiuda** e l'utente abbia il tempo di leggere la frase. Quando il destinatario premerà un tasto, il programma si chiuderà perchè non sono presenti altri comandi.*

Ma se noi volessimo far rispondere all'utente? Nessun problema. Subito dopo il comando echo scriviamo set/p **A=** che serve per far settare dall'utente la variabile "A". Si può scrivere qualsiasi cosa in alternativa ad "A", perchè con il testo che inseriremo in quel punto denomineremo la variabile. A quel punto il programma darà la possibilità all'utente di scrivere e qualsiasi cosa scriva verrà impostata come valore della variabile %A%. Ora noi dobbiamo **prevedere quali risposte darà l'utente con if**.

```
@echo off
title Ciao
color F9
```

```
:inizio
echo Ciao! Come stai?
set/p Ciao=
if %Ciao% == bene goto :r_bene
if %Ciao% == male goto :r_male
```

*Ok, la variabile si chiama "Ciao" e l'utente può rispondere con "bene" o con "male". Attenzione però: il programma fa distinzione tra maiuscole e minuscole sia nelle risposte che nei nomi delle variabili, quindi "Ciao" è diverso da "ciao" e "CIAO". Con if abbiamo detto al computer che se l'utente risponde "bene" lui deve andare all'etichetta ":r\_bene" e se risponde male di andare a ":r\_male". Goto significa "**vai a**" ed è sempre seguito dal nome di un'etichetta.*

Ma mettiamo che l'utente scriva "bho", il programma crasherebbe. Proprio per questo prima di aprire le etichette ":r\_bene" e ":r\_male" dobbiamo inserire un paragrafo che dia il messaggio d'errore. Se l'utente inserisse una risposta non prevista, il programma passerebbe immediatamente all'etichetta successiva: quindi **va messo il messaggio d'errore prima delle etichette di risposta**.

```
@echo off
title Ciao
color F9

:inizio
echo Ciao! Come stai?
set/p Ciao=
if %Ciao% == bene goto :r_bene
if %Ciao% == male goto :r_male

:errore
cls
echo Puoi rispondere solo con "bene" o "male"
pause>nul
goto :inizio
```

*Ora il messaggio d'errore dice: Puoi rispondere solo con "bene" o "male". E l'utente avrà il tempo per leggerlo e tornare poi all'inizio. **Cls** significa "**pulisci schermo**" e serve per cancellare la cronologia precedente, per una visione più "pulita" va usato all'inizio di ogni paragrafo.*

Ora possiamo scegliere le risposte. Faremo in modo che se l'utente digita "bene" il programma risponderà con "Yuu! Anche io." e se invece risponde con "male" risponderà dicendo "Oh, mi dispiace. Riposati!"

```
@echo off
title Ciao
color F9

:inizio
echo Ciao! Come stai?
set/p Ciao=
if %Ciao% == bene goto :r_bene
if %Ciao% == male goto :r_male

:errore
cls
echo Puoi rispondere solo con "bene" o "male"
pause>nul
goto :inizio

:r_bene
cls
echo Yuu! Anche io.
pause>nul
exit

:r_male
```

```
cls
echo Oh, mi dispiace. Riposati!
pause>nul
exit
```

*In questo modo abbiamo impostato un loop nel caso l'utente sbagliasse risposta e il programma si chiuderà dopo aver risposto all'utente. Il programma si sarebbe chiuso ugualmente se non avessimo usato exit perchè dopo pause non ci sono ulteriori comandi.*

**Ricapitoliamo tutti i comandi usati fino ad ora.**

- **@echo off** = va messo all'inizio di ogni programma
- **title** = serve per denominare il programma
- **color** = serve per scegliere i colori dell'interfaccia
- **echo** = serve per far visualizzare un testo specificato in seguito
- **set/p** = serve per creare e settare una variabile il cui valore è scelto dall'utente
- **if** = significa "se" e serve a prevedere gli input dell'utente
- **pause>nul** = serve per mettere in pausa il programma
- **cls** = serve per pulire lo schermo dalla cronologia

Ci sono altre due versioni dei comandi echo e pause:

- **pause** = ha la stessa funzione di pause>nul ma visualizza "Press a key to continue..."
- **echo.** = serve a saltare una riga

Esiste anche un tipo di **variabile che, scelta dallo sviluppatore, non può essere modificata dall'utente**. Ad esempio, in un programma che spiega geometria, va usato set per impostare a "pi" il valore di 3,14. Ecco l'esempio:

```
@echo off
title Geometria
color E9

:home
cls
set PI= 3,14
echo Cosa vuoi sapere?
set/p Domanda=
if %Domanda% == pi goto :pi_greco

:errore
cls
echo posso dirti solo il valore di  $\pi$ .
pause>nul
goto :home

:pi_greco
```

```
cls
echo il valore di pi equivale a %PI%
pause>nul
goto :home
```

*Perchè non ho usato il simbolo " $\pi$ "? Va saputo che **il linguaggio Batch non supporta lettere accentate e caratteri strani non presenti nella codifica ANSI.***

E se fossi uno sviluppatore a cui piace la matematica? Voglio programmare una calcolatrice. Bhè, in questo caso non posso fare a meno del comando set/a. **Set/a serve per le variabili numeriche, quindi l'utente non può digitare lettere.**

```
@echo off
title Calcolatrice
color 0F

:input
cls
echo Inserisci i due numeri che vuoi sommare:
set/p N1=
set/p N2=
set/a Rf= %N1%+%N2%

:risultato
cls
echo La somma dei due numeri equivale a %Rf%
pause>nul
goto :input
```

*Ecco fatto. Il programma chiede due numeri all'utente e poi ne calcola la somma. Nel caso l'utente digitasse un carattere non supportato (come una lettera) il programma non crasherebbe, ma darebbe 0 come risultato.*

Le variabili sono utili in diversi modi. Si possono inserire in molti punti del programma, anche per modificare un comando. Ad esempio per creare una nuova cartella.

```
@echo off
title Folder Maker
color FE

:nomecartella
cls
echo Come vuoi chiamare la tua nuova cartella?
set/p f=
mkdir C:\Users\Public\Documents\%f%
cls
echo La cartella e' stata creata con successo!
echo Puoi trovarla nei documenti.
```

```
pause>nul
goto :nomecartella
```

*Tramite questo programma si possono creare sottocartelle in "Documenti pubblici" e il nome della cartella è scelto dall'utente.*

Ora andiamo ad approfondire un po' il comando start. Start ha molte funzioni, può aprire il Prompt dei Comandi (cmd), può eseguire un programma specificato in seguito, può aprire una cartella con il processo explorer e può aprire un sito tramite il browser predefinito.

```
@echo off
title Launcher
color F0
```

```
:seleziona
cls
echo Cosa vuoi che esegua?
echo.
set/p S=
if %S% == word goto :word
if %S% == documenti goto :documenti
if %S% == sito goto :devsito
```

```
:errore
cls
echo Scegli una delle tre opzioni.
echo.
echo word, documenti, sito.
pause>nul
goto :seleziona
```

```
:word
cls
start winword
goto :seleziona
```

```
:documenti
cls
start C:\Users\Public\Documents
goto :seleziona
```

```
:devsito
cls
echo sito dello sviluppatore.
start www.crodattilo.tk
goto :seleziona
```

*Finito. Utilizzando start in questo modo e abbinandolo ai programmi scritti sopra (cercate una lista completa su Internet) potete eseguire tutti i processi di Windows, anche quelli dei programmi installati.*

### **-3 Personalizzazione**

È possibile personalizzare e modificare l'interfaccia del programma mentre quest'ultimo è in esecuzione, usando title e color all'inizio di un paragrafo. Ad esempio modificando il colore della scritta, rossa o verde, in risposte rispettivamente errate o corrette.

```
@echo off
title Quiz

:inizioquiz
color F9
cls
echo La RAM è una memoria volatile?
set/p q=
if %q% == sì goto :rispostagiusta
if %f% == no goto :rispostaerrata

:rispostaerrata
title : (
color FC
echo Risposta errata, riprova!
pause>nul
goto :inizioquiz

:rispostagiusta
title : )
color F2
echo Risposta giusta, bravo!
pause>nul
exit
```

*In questo caso non ci siamo serviti dell'etichetta e del paragrafo di errore, perchè qualsiasi parola non prevista da if porterà al messaggio di risposta errata. Ovviamente se vogliamo che l'interfaccia si modifichi senza bug dobbiamo usare title e color all'inizio di ogni tag e non all'inizio del programma.*