



# I DUE BASIC DEL PC 128

**Continuiamo il nostro viaggio all'interno di questo linguaggio di alto livello che tante soddisfazioni vi darà, grazie alla sua notevole versatilità e facilità d'uso.**

## 2ª Parte

**N**ella scorsa puntata, siamo giunti con la nostra esposizione fino alla gestione del comando PRINT, tramite la punteggiatura. Più esattamente, ci mancava ancora la virgola, e perciò è proprio da essa che iniziamo, per poi far subito un breve riepilogo.

### La virgola

Sintatticamente, la virgola segue le stesse regole della punteggiatura già vista, ha cioè effetto sul comando PRINT, quando viene posta di seguito a ciò che si vuol visualizzare:

```
PRINT a,b,c,
PRINT "Ciao", "a domani"
PRINT A$, B$
PRINT 1,2,3
```

come si può vedere abbiamo utilizzato sia stringhe, che variabili, che variabili stringa (le quali vanno logicamente definite in precedenza); termini che vi saranno più chiari fra non molto. Ciò che si ottiene, invece lo potete vedere direttamente sullo schermo: infatti tutto ciò che viene stampato dopo la virgola, viene incolonnato con spazi di tredici caratteri, partendo dalla prima colonna. Questa opzione, è normalmente utilizzata per visualizzare delle tabelle ordinate in colonne, senza dover ricorrere ad altri tipi di tabulazione più complessi che vedremo più avanti.

### Riepilogo della punteggiatura

: separa due istruzioni nella stessa riga di istruzioni

; se posto un numero o una stringa di caratteri da stampare, il punto e virgola forza una visualizzazione sulla stessa riga subito dopo quella precedente.

? abbreviazione del comando PRINT.

, se posta tra due numeri o due stringhe di caratteri, la virgola provoca la visualizzazione nella riga ogni quattordici colonne a partire dalla prima.

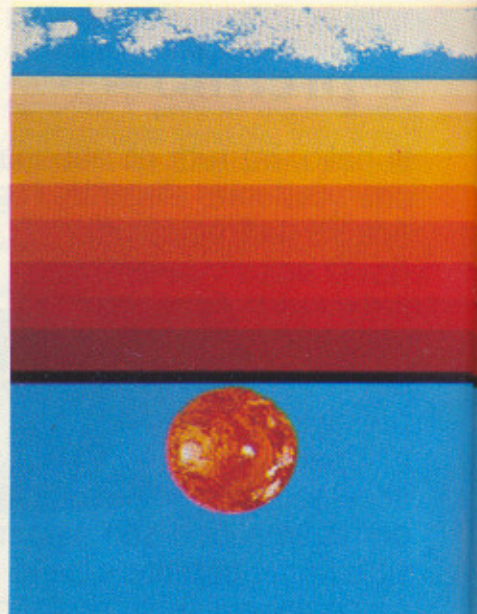
. viene usato come virgola nei numeri decimali.

### Dimensionamento del formato di scrittura

Sono disponibili quattro dimensioni diverse per dimensionare i caratteri a nostro piacimento. Essi sono selezionati tramite un apposito comando; ATTRB, di cui diamo i diversi parametri nella Tab. A.

Se non viene specificato diversamente, il Basic considera per default il valore ATTRBO,0. In caso d'attivazione del comando tramite valori diversi, per ritornare al valore di default o ad altri, questi dovranno essere indicati successivamente.

È importante tener presente, che in caso di richiesta di stampa



con caratteri in dimensione doppia, l'istruzione LOCATE, è riferita al punto inferiore sinistro del carattere. Ciò è di fondamentale interesse, al fine di non tagliare la parte superiore delle scritte, a causa di una errata utilizzazione del comando suddetto.

```
10 ATTRB,1: LOCATE 10,10:
PRINT "CIAO" 11 LOCATE10,11:
PRINT "CIAO"
```

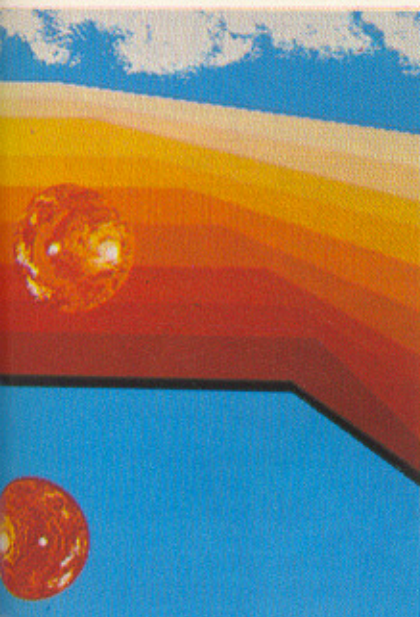
Osservare attentamente ciò che accade e trarne insegnamento.

### L'istruzione COLOR

È spesso utile, al fine di mettere maggiormente in rilievo alcune parole, che queste abbiano colore diverso dal resto del testo. Tutto ciò è facilmente realizzabile tramite l'istruzione COLOR.

Per esempio con l'istruzione COLOR, 4, si otterrà una visualizzazione di una scritta rossa (1) in campo blu (4). La sintesi di questo comando è del tutto simile a quella di SCREEN, dato che la prima cifra indica il colore dei caratteri, mentre la seconda indica il colore del fondo. C'è una differenza fondamentale però: mentre SCREEN agisce su





tutto lo schermo, l'istruzione LOCATE agisce solo sulla visualizzazione direttamente interessata.

## Alcune utili osservazioni

Nel caso in cui si introduca un comando del tipo:  
LOCATE 35, 10: PRINT "UTILISSIMA"

ciò pone il computer davanti ad un problema che però il PC 128 risolverà da solo. Richiedendo infatti la stampa della stringa "UTILISSIMA" (10 caratteri), è impossibile non dividere la frase in due tronconi, poco estetici. A questo punto il computer decide autonomamente di non scrivere la stringa nel punto da noi indicato, ma all'inizio della riga seguente. Ciò ci fa capire che il PC 128, ogni qual volta noi lo poniamo di fronte ad un'istruzione PRINT, valuta prima se è possibile stamparla senza interruzioni e in caso contrario, come abbiamo potuto vedere, ci scavalca e la stampa dove giudica più opportuno. Questo logicamente può creare dei disagi (a qualcuno anche d'identità), che però saranno facilmente risolvibili tramite l'istruzione PRINT USING "&", che verrà spiegata più avanti, o tramite un bravo Psicologo.

	ALTEZZA	LARGHEZZA
ATTRB0,0	normale	normale
ATTRB1,0	normale	doppia
ATTRB0,1	doppia	normale
ATTRB1,1	doppia	doppia

## Puntualizzazioni sul comando LOCATE

L'istruzione LOCATE prevede un terzo argomento, che decide della presenza o meno del cursore sullo schermo.

Soprattutto in campo grafico, spesso è necessaria l'assenza del cursore, il quale non avrebbe nessuna ragione d'essere se non quella di rovinare il disegno stesso. A tal fine, tramite l'aggiunta del terzo argomento a zero, si può eliminare tranquillamente l'intruso dalle nostre preziose schermate.

LOCATEX,Y,0

Con ciò si disattiva il cursore.

LOCATEX,Y,1

Con ciò si riattiva il cursore sul video.

## Le variabili

Veniamo ora ad un argomento importantissimo: le variabili.

Proviamo a scrivere:

```
PRINT CIAO
O
OK
```

Come potete vedere il computer non è molto educato e non risponde correttamente. Ciò però non è del tutto vero, è solamente un po' ingorante e non è abituato a rispondere se prima non ha ricevuto istruzioni. Infatti proviamo a scrivere:

```
10 ANNO = 1986
20 PRINT ANNO
RUN
```

```
1986
OK
```

Come potete vedere, questa volta, il vostro PC 128 ha studiato la lezione, e alla vostra richiesta di ricordarvi l'anno in corso, vi ha risposto: 1986. Tralasciando la fantascienza, ciò che è avvenuto è molto più semplice, infatti tramite l'istruzione ANNO = 1986, noi abbiamo rivolta al computer ciò che normalmente viene chiamata una "istruzione di assegnazione". Tramite questa, si ordina al PC 128, di depositare, in qualche parte della sua memoria, il numero 1986 e di legarlo alla variabile ANNO, in modo che al nostro comando PRINT ANNO, il computer ne stampi il valore. Questo avviene perché, alla nostra richiesta, il computer andrà a cercare nelle sue celle di memoria il valore corrispondente. A questo punto ogni qualvolta ci servirà il valore 1986, basterà chiamarlo tramite la variabile ANNO. A dire il vero, nell'esempio fatto, ANNO, non si comporta esattamente come una variabile, infatti non "varia" affatto. Vediamo ora un'altro esempio:

```
10 A = 25.7 - 12.8
20 C = A * B
30 PRINT A,B,C
RUN
25.7                12.8
                    328.96
OK
```

Come si può vedere, grazie al separatore ",", i rispettivi valori di A, B e C vengono incollati come abbiamo visto prima. Le variabili A, B, e C, sono seguite dopo il punto da dei decimali, perciò esse vengono chiamate "variabili reali", per distinguerle da: A%, B% e C%, che vengono chiamate "variabili intere". Infatti provate a scrivere:

```
10 A% = 25.7 : B% = 12.8
20 C% = A% * B%
30 PRINT A%, B%, C%
```





```
RUN
26
338
OK
```

Qualcosa c'è perso per strada, non c'è più il punto e tanto meno i decimali. A%, B% e C% infatti sono delle variabili intere, e come tali vengono immagazzinate senza eventuali parti decimali. È importante sapere, che prima di una qualsiasi istruzione di assegnazione, il contenuto delle variabili è inizializzato a 0. Ecco quindi perché, alla nostra richiesta di stampare "ANNO", il PC 128 ci ha risposto "0".

### Trasferimento

Fra le varie operazioni svolgibili con le variabili, c'è anche quella di trasferimento, la quale consente di passare il valore di una variabile ad un'altra:

```
10 A = 10: B = 0
20 PRINT A;B
30 B = A: PRINT A;B
RUN
10 0
10 10
OK
```

L'istruzione B=A trasferisce il valore di A in B, cancellando però il valore contenuto in B. Infatti:

```
10 A = 10: B = 20
20 PRINT A;B
30 B = A: PRINT A;B
RUN
10 20
10 10
OK
```

bisogna quindi porre molta attenzione onde evitare di perdere dei valori per strada.

### Permuta

Disponendo di una variabile X e di una variabile Y, per permuta si intende quell'operazione atta al trasferimento di Y e in X e di X in Y.

Per fare ciò non si potrà certo adottare la tecnica sopra esposta, infatti ponendo X=Y si otterrà sia il



trasferimento del valore di Y in X, ma anche la cancellazione del valore originario di X. A questo punto ci troviamo in un vicolo cieco dal quale si potrà uscire solo con l'aiuto di un'idea brillante o di un aiuto esterno. Questo aiuto esterno ci verrà dato dalla variabile Z in questo modo:

```
10 X = 30: Y = 40: PRINT X;Y
20 Z = Y: Y = X: X = Z: PRINT X;Y
RUN
20 40
40 30
OK
```

Quanto abbiamo appena visto è ciò che si sarebbe dovuto fare con qualsiasi altro BASIC, e non è detto che vi ricordiate sempre questo procedimento, ma nel vostro PC 128 c'è un'istruzione che vi darà una mano in situazioni analoghe. Provate a scrivere:

```
10 X = 30: Y = 40: PRINT X;Y
20 SWAP X,Y: PRINT X;Y
RUN
30 40
40 30
OK
```

visto com'è facile! Nota: per richiamare una variabile (in questo caso ANNO), il PC 128 utilizza un indiriz-

zo di memoria, il quale altro non è che un numero. La memoria RAM viene suddivisa in caselle, ognuna delle quali ha un suo proprio riferimento numerico.

Una casella di memoria può contenere indifferentemente un numero, un carattere, il contenuto di un'area video, una parte delle istruzioni PRINT, ecc. Nel nostro caso il numero 1986 è immagazzinato dal computer in una precisa area di memoria RAM; per poter accedere al suo indirizzo esiste una semplice istruzione BASIC: VARPTR.

```
PIRINT VARPTR (ANNO)
—24881
OK
```

Il numero così ottenuto, dipende logicamente da ciò che si è fatto in precedenza, prima cioè di questa istruzione, dà l'indirizzo della prima cella di memoria contenente il valore della variabile reale ANNO. A questo punto è opportuno dire che normalmente il contenuto di una variabile generica, viene di solito disposto in celle di memoria consecutive e viene sottoposto ad una complessa codifica matematico-informatica. Ricordiamo che per evitare interferenze indesiderate da parte dell'utente, ed anche per alleviarlo da un compito gravoso, il BASIC lascia il compito dell'organizzazione della memoria al micro computer stesso.

Precisiamo che i nomi da dare alle variabili devono seguire una semplice regola: non devono cioè iniziare con nessuna delle parole chiave del BASIC (PRINT, COLOR,...). L'elenco di tali parole è contenuto nel MANUALE D'USO DEL PC 128.

VALORE = PREZZO \* SCONTO

Questa linea d'istruzione genererà un bel "Syntax Error" (Errore di sintassi), perché la variabile VALORE inizia con VAL, la quale è una delle parole chiave del BASIC.

### Variabili stringhe ed altre

Il simbolo che caratterizza le va-



riabili stringa è \$, detto anche dollaro. Questo consente all'interprete BASIC di capire che si stanno trattando dei caratteri e non dei numeri. Anche con questo tipo di variabili è possibile utilizzare l'assegnazione. L'esempio che riportiamo consente di memorizzare la stringa "ANNO" nella variabile X\$.

```
X$ = "ANNO":PRINT X$:PRINT X$
ANNO
ANNO
OK
```

Gran parte di quanto detto in precedenza sulle variabili è valido anche per le stringhe. Ciò che le differenzia è solo il dollaro e la presenza delle virgolette. Bisogna però stare attenti a non confondere una variabile per un'altra, perché ciò indurrebbe il computer in errore.

Le istruzioni del tipo X = "ANNO" e X\$ = 139, determinano automaticamente un messaggio d'errore. Per lo stesso motivo non è possibile scambiare tra loro i valori di variabili di tipo diverso, come: A\$ = B, oppure B = A\$. È possibile invece effettuare operazioni fra variabili stringa, allo stesso modo in cui è possibile effettuare operazioni fra variabili numeriche.

```
X$ = "CA":Y$ = "SA":PRINT X$:Y$
CASA
OK
```

```
SWAPX$,Y$:PRINT X$:Y$
SACA
OK
```

Vedremo in una prossima lezione come sia possibile lavorare sulle stringhe tramite dei comandi appositi, che rendono il trattamento dei testi veramente facile e rapido; non per nulla, il BASIC è forse il linguaggio più adatto al trattamento dei testi, se non il più efficace senz'altro il più semplice.

## Di nuovo sulle variabili intere

Questo tipo di variabili, di cui abbiamo visto più sopra le caratteri-

stiche principali, possono essere utilizzate in un campo che va da -32768 a 32767. Oltre a quanto già detto, una variabile intera si può anche inizializzare in un altro modo: se si fa precedere l'assegnazione della dichiarazione DEFINT, ciò ci dispenserà dall'uso del segno "%", ed inoltre, questo metodo, permette la realizzazione di arrotondamenti automatici.

```
10 DEFINT A: A = 5.23
20 PRINT A
RUN
5
OK
```

Come si può notare il valore di A è stato arrotondato al valore inferiore, cioè a 5, senza ulteriori comandi.

## Variabili reali a singola precisione

Quando in una variabile non viene definito il tipo, essa sarà considerata una variabile reale a precisione singola. Variabili di questo tipo, prendono in considerazione un massimo di sette cifre significative. Il loro nome può essere seguito da un simbolo "!"; facoltativo.

## Variabili reali a doppia precisione

Questo tipo di variabili vengono utilizzate come le precedenti, a differenza di quelle, tengono in considerazione fino a quindici cifre significative. Il nome di queste variabili, dev'essere seguito dal simbolo "#" oppure il loro tipo dev'essere definito in precedenza, tramite l'istruzione DEFDBL. Questa variabile, può essere utilizzata per aumentare la precisione di determinati calcoli, quali la divisione. L'esempio sottostante ne mostra l'uso:

```
DEFDBL Z
B = 10/7:Z = 10 # / 7 #:PRINT
B:PRINT Z
1.42857
1.428571428571829
OK
```

Si può notare che nel caso di calcolo a doppia precisione, per effettuare queste operazioni, le costanti 10 e 7 devono essere seguite dal segno "#".

Anche per questa volta abbiamo terminato, consigliamo tutti gli interessati di provare con dei piccoli programmi, magari di propria ideazione, quanto si è esposto finora, così da prendere dimestichezza, sia con il computer che con il BASIC. Un altro consiglio che ci sentiamo di dare, è di copiare i listati riportati su questa rivista nell'apposita sezione, così da avere un rapporto più diretto con la programmazione. È molto importante infatti abituarsi a vedere i comandi che formano questo linguaggio il più possibile, in tal modo li ricorderete con più facilità. Oltre a ciò i listati presentati sono dei giochi bellissimi e divertenti. ■

