TI-Nspire™ Guida dell'editor del programma

Per sapere di più sulle tecnologie TI, consultare la guida online all'indirizzo education.ti.com/eguide..

Important Information

Tranne che espressamente specificato nella Licenza che accompagna un programma, Texas Instruments non rilascia alcuna garanzia, esplicita o implicita, comprese ma non limitate a garanzie implicite di commerciabilità e idoneità a un determinato scopo, riguardanti qualsiasi programma o materiale di materiali disponibili solo su base "as-is". Texas Instruments non sarà in nessun caso responsabile di danni specifici, collaterali, incidentali o consequenziali in connessione con o derivanti dall'acquisto o dall'utilizzo di questi materiali e dalla responsabilità esclusiva e esclusiva di Texas Instruments, indipendentemente dalla forma di non supera l'importo indicato nella licenza per il programma. Inoltre, Texas Instruments non sarà responsabile per qualsiasi pretesa di alcun tipo contro l'utilizzo di questi materiali da parte di qualsiasi altra parte.

Il software TI-Nspire™ utilizza Lua come ambiente di scripting. Per informazioni sul copyright e sulle licenze, vedere http://www.lua.org/license.html.

Il software TI-Nspire™ utilizza Chipmunk Fisica versione 5.3.4 come ambiente di simulazione. Per informazioni sulle licenze, vedere http://chipmunk-physics.net/release/Chipmunk-5.x/Chipmunk-5.3.4-Docs/.

Microsoft® e Windows® sono marchi registrati di Microsoft Corporation negli Stati Uniti e / o in altri paesi.

Mac OS®, iPad® e OS X® sono marchi registrati di Apple Inc.

Unicode® è un marchio registrato di Unicode, Inc. negli Stati Uniti e in altri paesi.

I prodotti reali possono differire leggermente dalle immagini pubblicate.

Contenuto

Guida introduttiva all'Editor di programmi

È possibile creare funzioni o programmi definiti dall'utente digitando istruzioni nella riga di introduzione di Calcolatrice oppure utilizzando l'Editor di programmi L'Editor di programmi offre diversi vantaggi ed è descritto in questo capitolo. Per ulteriori informazioni, vedere il capitolo *Calcolatrice*.

- L'editor dispone di modelli di programmazione e finestre di dialogo che aiutano a definire funzioni e programmi utilizzando la sintassi corretta.
- L'editor consente di inserire istruzioni multiple di programmazione senza dover specificare una speciale sequenza di tasti per aggiungere ogni riga.
- È possibile creare facilmente oggetti libreria pubblica e privata (variabili, funzioni e programmi). Per ulteriori informazioni, consultare *Librerie*.

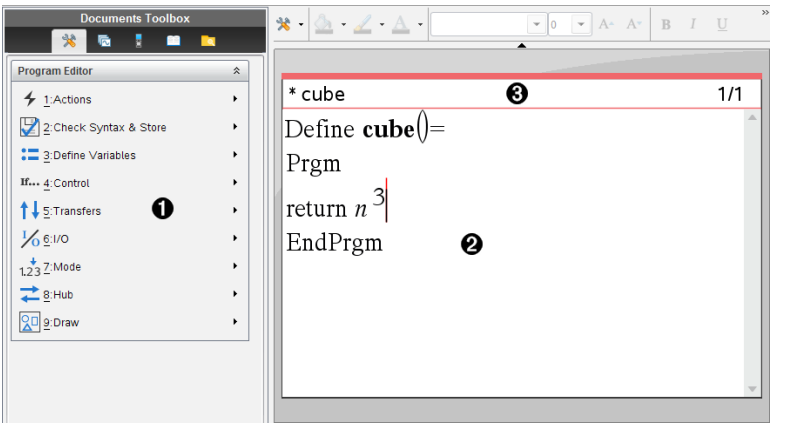
Avvio dell'Editor di programmi

▶ Per aggiungere una pagina nuova dell'Editor di programmi all'attività corrente:

Dalla barra degli strumenti, fare clic su **Inserisci > Editor di programmi > Nuovo**.

Palmare: premere doc e selezionare **Inserisci > Editor di programmi > Nuovo**.

Nota: l'editor è accessibile anche dal menu **Funzioni e& programmi** di una pagina Calcolatrice.



❶ Menu dell'Editor di programmi– Questo menu è sempre disponibile nell'area di lavoro dell'Editor di programmi quando si utilizza la modalità di visualizzazione Normale.

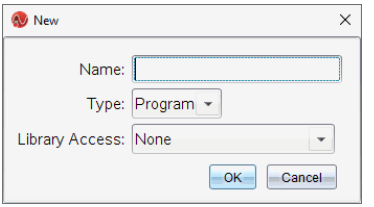
❷ Area di lavoro dell'Editor di programmi

❸ La linea di stato mostra le informazioni del numero della riga e il nome della funzione o del programma che si sta modificando. Un asterisco (*) indica che l'informazione non è più aggiornata poiché è cambiata dall'ultima volta che se ne è controllata la sintassi e che è stata salvata.

Definizione di un programma o di una funzione

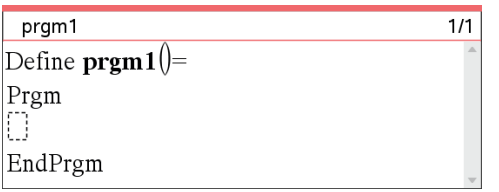
Avvio di una nuova sessione di Editor di programmi

1. Assicurarsi di trovarsi nel documento o nel problema in cui si desidera creare il programma o la funzione.
2. Fare clic sul pulsante **Inserisci** sulla barra degli strumenti dell'applicazione, quindi selezionare **Program Editor > Nuovo**. Sul palmare, premere doc e selezionare **Inserisci > Program Editor > Nuovo**.



3. Digitare un nome per la funzione o il programma che si sta definendo.
4. Selezionare il **Tipo** (**Programma** o **Funzione**).
5. Impostare l'**Accesso libreria**:
   - Per utilizzare la funzione o il programma solo dal documento o dal problema corrente, selezionare **Nessuno**.
   - Per rendere la funzione o il programma accessibile da qualsiasi documento ma non visibile nel Catalogo, selezionare **LibPriv**.
   - Se si desidera che la funzione o il programma sia accessibile da qualsiasi documento e che sia anche visibile nel Catalogo, selezionare **LibPub (mostra nel catalogo)**. Per ulteriori informazioni, consultare *Librerie*.
6. Fare clic su **OK**.

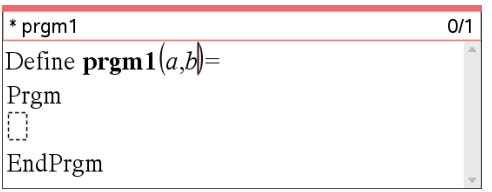
   Viene aperta una nuova istanza di Program Editor con un modello basato sulle selezioni effettuate.



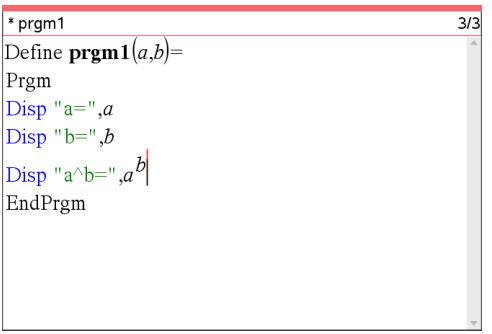
Introduzione di righe in una funzione o in un programma

L'Editor di programmi non esegue i comandi o non calcola le espressioni in fase di introduzione. Vengono eseguite solo quando si calcola la funzione o si esegue il programma.

1. Se la funzione o il programma richiede all'utente di specificare degli argomenti, digitare i nomi dei parametri tra le parentesi che seguono il nome. Separare i parametri con una virgola.



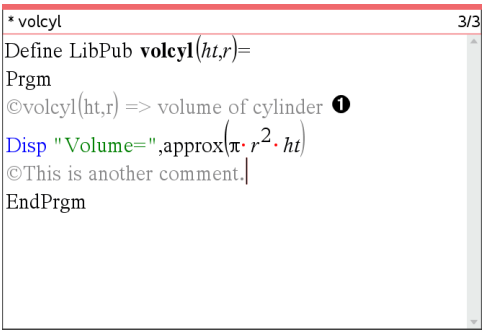
2. Tra le linee Func e EndFunc (o Prgm e EndPrgm), digitare le linee di istruzioni che compongono la funzione o il programma.



- È possibile digitare i nomi delle funzioni o dei comandi oppure inserirli dal Catalogo.
- Una linea può essere più lunga della larghezza dello schermo; in questo caso, è possibile scorrere per visualizzare l'intera istruzione.
- Dopo aver digitato ciascuna linea, premere **Invio**. Così facendo si inserisce una nuova linea vuota ed è possibile continuare a inserire un'altra linea.
- Utilizzare i tasti freccia ◄, ►, ▲ e ▼ per scorrere lungo la funzione o il programma e inserire o modificare i comandi.

Inserimento di commenti

I commenti possono essere utili per chi visiona o modifica il programma. Non appaiono quando il programma è in esecuzione e non hanno alcun effetto sul funzionamento del programma. Il simbolo © viene visualizzato all'inizio della linea con il commento.



❶ Commento che illustra la sintassi richiesta. Poiché questo oggetto libreria è pubblico e questo commento è la prima linea di un blocco Func o Prgm, il commento viene visualizzato nel Catalogo come guida. Per ulteriori informazioni, consultare *Librerie*.

Per inserire un commento:

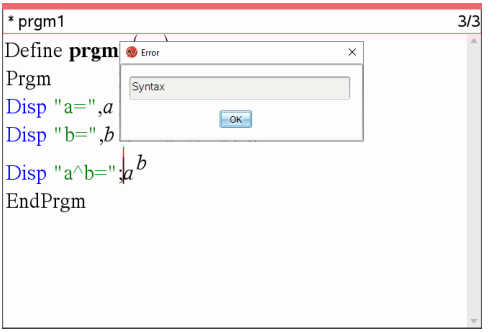
1. Posizionare il cursore alla fine della linea in cui si desidera inserire un commento.
2. Dal menu **Azioni**, fare clic su **Inserisci commento** oppure premere **Ctrl+T**.
3. Digitare il testo del commento dopo il simbolo ©.

Controllo della sintassi

Program Editor consente di verificare la correttezza della sintassi della funzione o del programma.

- Dal menu **Controlla sintassi e salva**, fare clic su **Controlla sintassi**.

  Se vengono rilevati eventuali errori di sintassi, l'applicazione di controllo visualizza un messaggio di errore e tenta di portare il cursore in prossimità del primo errore in modo che sia possibile correggerlo.



Salvataggio della funzione o del programma

È necessario salvare la funzione o il programma per renderlo accessibile. L'Editor di programmi controlla automaticamente la sintassi prima di procedere al salvataggio.

Un asterisco (*) visualizzato nell'angolo superiore sinistro di Program Editor indica che non è stato eseguito il salvataggio della funzione o del programma.

- Dal menu **Controlla sintassi e salva**, fare clic su **Controlla sintassi e salva**.

  Se vengono rilevati eventuali errori di sintassi, l'applicazione di controllo visualizza un messaggio di errore e tenta di portare il cursore in prossimità del primo errore.

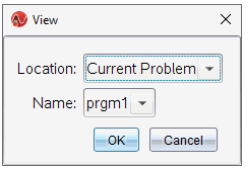
  Se non vengono rilevati errori di sintassi, viene visualizzato il messaggio "Salvataggio effettuato correttamente" sulla riga di stato nella parte superiore di Program Editor.

Nota: Se la funzione o il programma è definito come oggetto libreria, occorre inoltre salvare il documento nella cartella della libreria designata e aggiornare le librerie per rendere l'oggetto accessibile ad altri documenti. Per ulteriori informazioni, consultare *Librerie*.

Visualizzazione di un programma o di una funzione

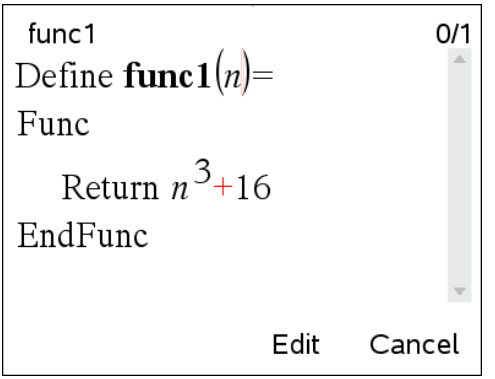
1. Nel menu **Azioni (Actions)** selezionare **Visualizza (View)**.

   Viene visualizzata la finestra di dialogo Visualizza.



2. Se la funzione o il programma è un oggetto libreria, selezionare la relativa libreria dall'elenco **Ubicazione** (Location).
3. Selezionare il nome della funzione o del programma dall'elenco **Nome**.

   La funzione o il programma viene mostrato in un visualizzatore.



4. Utilizzare i tasti freccia per vedere la funzione o il programma.
5. Per modificare il programma, fare clic su **Modifica (Edit)**.

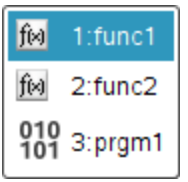
Nota: Modifica (Edit) può essere selezionato solo per funzioni e programmi definiti nell'attività corrente. Per modificare un oggetto libreria, occorre aprirne prima il documento libreria.

Apertura di una funzione o di un programma per la modifica

È possibile aprire una funzione o un programma solo dall'attività corrente.

Nota: non è possibile modificare un programma o una funzione bloccati. Per sbloccare l'oggetto, accedere alla pagina di Calcolatrice e utilizzare il comando **unLock**.

1. Visualizza l'elenco delle funzioni e dei programmi disponibili.
   - Nel menu **Azioni (Actions)** selezionare **Apri (Open)**.

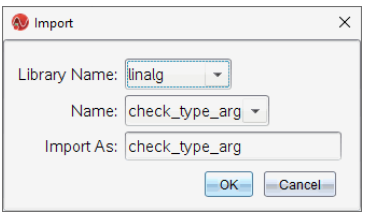


2. Selezionare l'elemento da aprire.

Importazione di un programma da una libreria

È possibile importare una funzione o un programma definito come oggetto libreria in un Editor di programmi nell'attività corrente. La copia importata non è bloccata, anche se l'originale è bloccato.

1. Nel menu **Azioni (Actions)**, selezionare **Importa (Import)**.



2. Selezionare il **Nome libreria (Library Name)**.
3. Selezionare il **Nome (Name)** dell'oggetto.
4. Se si desidera che l'oggetto importato abbia un nome diverso, digitare il nome in **Importa come (Import As)**.

Creazione di una copia di una funzione o di un programma

Quando si crea una nuova funzione o un nuovo programma, potrebbe essere più facile iniziare da una copia della funzione o del programma corrente. La copia che si crea non è bloccata, anche se l'originale è bloccato.

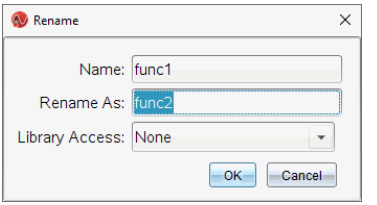
1. Nel menu **Azioni (Actions)** selezionare **Crea copia (Create Copy)**.

2. Digitare un nuovo nome oppure fare clic su **OK** per accettare il nome proposto.
3. Se si desidera modificare il livello di accesso, selezionare **Accesso libreria**, quindi selezionare un nuovo livello.

Ridenominazione di un programma o di una funzione

È possibile rinominare e (facoltativamente) modificare il livello di accesso della funzione o del programma corrente.

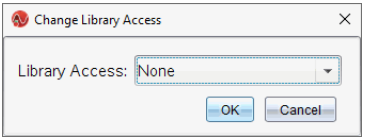
1. Nel menu **Azioni (Actions)**, selezionare **Rinomina (Rename)**.



2. Digitare un nuovo nome oppure fare clic su **OK** per accettare il nome proposto.
3. Se si desidera modificare il livello di accesso, selezionare **Accesso libreria**, quindi selezionare un nuovo livello.

Modifica del livello di accesso libreria

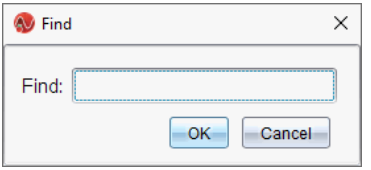
1. Nel menu **Azioni (Actions)** selezionare **Cambia accesso libreria (Change Library Access)**.



2. Selezionare l'**Accesso libreria**:
   - Se si desidera utilizzare la funzione o il programma solo dal dall'attività corrente di Calcolatrice, selezionare **Nessuno (None)**.
   - Se si desidera che la funzione o il programma sia accessibile da qualsiasi documento, ma non sia visibile nel Catalogo, selezionare **LibPriv**.
   - Se si desidera che la funzione o il programma sia accessibile da qualsiasi documento e sia anche visibile nel Catalogo, selezionare **LibPub**.

Ricerca di testo

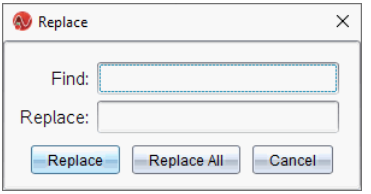
1. Nel menu **Azioni (Actions)**, selezionare **Trova (Find)**.



2. Digitare il testo da ricercare e fare clic su **OK**.
   - Il testo, se trovato, viene evidenziato nel programma.
   - Se non viene trovato, appare un messaggio di notifica.

Ricerca e sostituzione di testo

1. Nel menu **Azioni (Actions)** selezionare **Trova e sostituisci (Find and Replace)**.



2. Digitare il testo da ricercare.
3. Digitare il testo sostitutivo.
4. Fare clic su **Sostituisci** per sostituire la prima occorrenza dopo la posizione del cursore, oppure fare clic su **Sostituisci tutto** per sostituire tutte le occorrenze.

   Nota: Se il testo viene trovato in un modello matematico, compare un messaggio che vi avverte che il testo sostitutivo sta per rimpiazzare l'intero modello — non soltanto il testo trovato.

Chiusura della funzione o del programma corrente

- Nel menu **Azioni Actions (Actions)** selezionare **Chiudi (Close)**.

  Se nella funzione o nel programma ci sono modifiche non ancora salvate, viene chiesto di controllare la sintassi e di salvare prima di chiudere.

Esecuzione di programmi e calcolo di funzioni

Dopo aver definito e memorizzato un programma o una funzione, è possibile utilizzarli da un'applicazione. Tutte le applicazioni possono calcolare le funzioni, ma solo le applicazioni Calcolatrice e Notes possono eseguire i programmi.

Le istruzioni del programma vengono eseguite in ordine sequenziale (anche se alcuni comandi alterano il flusso del programma). L'output, se esiste, viene visualizzato nell'area di lavoro dell'applicazione.

- L'esecuzione di un programma continua fino al raggiungimento dell'ultima istruzione o di un comando **Stop**.

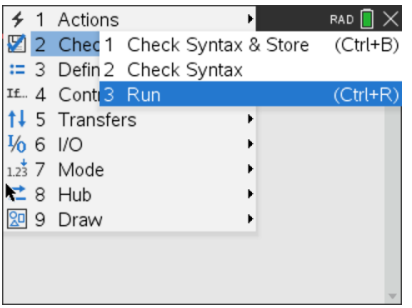
- L'esecuzione di una funzione continua fino al raggiungimento di un comando **Return**.

Esecuzione di un programma o di una funzione dall'Editor di programmi

1. Assicurarsi di aver definito un programma o una funzione e che l'Editor di programmi corrisponda al riquadro (computer) o alla pagina (palmare) attiva.

2. Sulla barra degli strumenti, fare clic sul pulsante **Strumenti documento** quindi selezionare **Controlla sintassi e salva > Esegui**.
   -oppure-
   Premere **Ctrl+R**.

   Palmare: Premere menu 2 3, oppure premere ctrl R.



   In questo modo verrà effettuato quanto segue:

   - verrà controllata la sintassi e verrà memorizzato il programma o la funzione;
   - verrà incollato il nome del programma o della funzione sulla prima linea disponibile dell'applicazione Calcolatrice immediatamente dopo l'Editor di programmi. Se non è presente una Calcolatrice in tale posizione, ne verrà inserita una nuova.

3. Se la funzione o il programma richiede di specificare uno o più argomenti, digitare i valori o i nomi di variabile all'interno delle parentesi.

4. Premere enter.

Nota: è possibile inoltre eseguire un programma o una funzione nelle applicazioni Calcolatrice o Notes digitando il nome del programma con parentesi ed eventuali argomenti richiesti, quindi premendo enter.

Utilizzo di abbreviazioni e nomi estesi

Ogniqualvolta si incorre nel problema di definire un oggetto, è possibile accedervi immettendone l'abbreviazione (il nome fornito nel comando **Define** dell'oggetto). Ciò vale per tutti gli oggetti definiti, compresi gli oggetti privati, pubblici e non libreria.

È possibile accedere a un oggetto libreria da qualsiasi documento digitando il nome esteso dell'oggetto. Un nome esteso è costituito dal nome del documento della libreria dell'oggetto seguito da una barra retroversa "\" seguita dal nome dell'oggetto. Ad esempio, il nome dell'oggetto definito come **func1** nel documento della libreria **lib1** è **lib1\func1**. Per digitare il carattere "\" sul palmare, premere shift ÷.

Nota: se non si riesce a ricordare il nome esatto o l'ordine degli argomenti richiesti per un oggetto libreria privata, è possibile aprire il documento libreria oppure utilizzare l'Editor di programmi per visualizzare l'oggetto. È inoltre possibile utilizzare **getVarInfo** per visualizzare un elenco di oggetti in una libreria.

Utilizzo di una funzione o di un programma libreria pubblica

1. Accertarsi di avere definito l'oggetto nella prima attività del documento, di averlo memorizzato, di aver salvato il documento libreria nella cartella MieLibrerie e di aver aggiornato le librerie.
2. Aprire l'applicazione TI-Nspire™ in cui si desidera utilizzare la funzione o il programma.

   Nota: tutte le applicazioni possono calcolare funzioni, tuttavia solo le applicazioni Calcolatrice e Notes possono eseguire programmi.

3. Aprire il Catalogo e utilizzare la scheda librerie per trovare e inserire l'oggetto.
   -oppure-
   Digitare il nome dell'oggetto. Nel caso di una funzione o di un programma, terminare sempre il nome con parentesi.

   ```
   libs2\func1()
   ```

4. Se la funzione o il programma richiede di specificare uno o più argomenti, digitare i valori o i nomi di variabile all'interno delle parentesi.

   ```
   libs2\func1(34,power)
   ```

5. Premere enter.

Utilizzo di una funzione o di un programma libreria privata

Per utilizzare un oggetto libreria privata, è necessario conoscerne il nome esteso. Ad esempio il nome esteso dell'oggetto definito come **func1** nel documento libreria **lib1** è **lib1\func1**.

Nota: se non si riesce a ricordare il nome esatto o l'ordine degli argomenti richiesti per un oggetto libreria privata, è possibile aprire il documento libreria oppure utilizzare l'Editor di programmi per visualizzare l'oggetto.

1. Accertarsi di avere definito l'oggetto nella prima attività del documento, di averlo memorizzato, di aver salvato il documento libreria nella cartella MieLibrerie e di aver aggiornato le librerie.
2. Aprire l'applicazione TI-Nspire™ in cui si desidera utilizzare la funzione o il programma.

   Nota: tutte le applicazioni possono calcolare funzioni, tuttavia solo le applicazioni Calcolatrice e Notes possono eseguire programmi.
3. Digitare il nome dell'oggetto. Nel caso di una funzione o di un programma, terminare sempre il nome con parentesi.

   ```
   libs2\func1()
   ```
4. Se l'oggetto richiede di specificare uno o più argomenti, digitare i valori o i nomi di variabile all'interno delle parentesi.

   ```
   libs2\func1(34,power)
   ```
5. Premere enter.

Interruzione di un programma o di una funzione in esecuzione

Durante l'esecuzione di una funzione o di un programma, il puntatore visualizzato è ⧖ "occupato".

- Per arrestare la funzione o il programma:
  - Windows®: Tenere premuto il tasto **F12** e premere **Invio** più volte.
  - Mac®: Tenere premuto il tasto **F5** e premere **Invio** più volte.
  - Palmare: Tenere premuto il tasto ⌂ on e premere enter più volte.

  Viene visualizzato un messaggio. Per modificare il programma o la funzione nell'Editor di programmi, selezionare **Vai a**. Il cursore viene visualizzato in corrispondenza del comando in cui avviene l'interruzione.

Inserimento di valori in un programma

È possibile scegliere tra diversi metodi per fornire i valori per il calcolo a una funzione o un programma.

Inserimento dei valori nella funzione o nel programma

Questo metodo è utile principalmente per valori che devono essere uguali ogni volta che si utilizza il programma o la funzione.

1. Definire il programma.

```
* calculatearea                    4/4
Define calculatearea()=
Prgm
w:=3
h:=23.64
area:=w·h
Disp area
EndPrgm
```

2. Eseguire il programma.

```
calculatearea()
                                  70.92
                                  Done
```

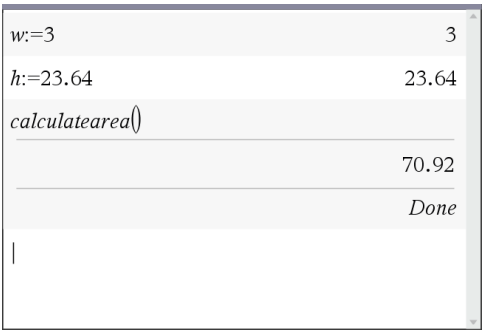
Assegnazione di valori a variabili da parte dell'utente

Un programma o una funzione possono fare riferimento a variabili precedentemente create. Per utilizzare questo metodo l'utente deve ricordare i nomi delle variabili e assegnare loro dei valori prima di utilizzare l'oggetto.

1. Definire il programma.

```
* calculatearea                    2/2
Define calculatearea()=
Prgm
area:=w·h
Disp area
EndPrgm
```

2. Indicare le variabili, quindi eseguire il programma.

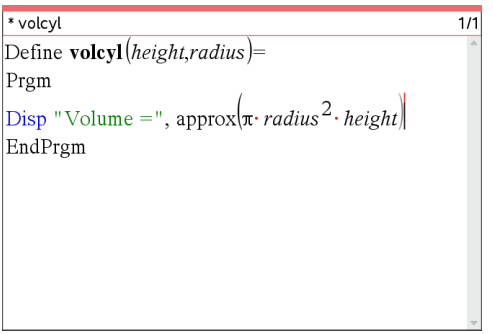


Specifica dei valori come argomenti da parte dell'utente

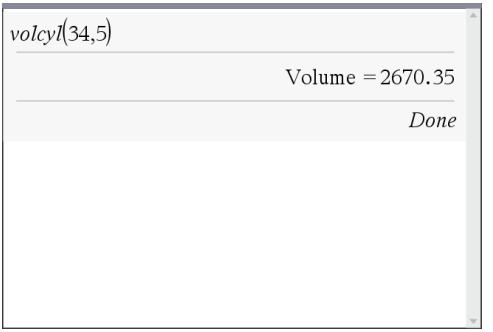
Questo metodo consente all'utente di passare uno o più valori come argomenti all'interno dell'espressione che chiama il programma o la funzione.

Il seguente programma, **volcyl**, calcola il volume di un cilindro. Richiede all'utente di specificare due valori: altezza e raggio del cilindro.

1. Definire il programma **volcyl**.



2. Eseguire il programma per visualizzare il volume di un cilindro avente altezza di 34 mm e raggio di 5 mm.



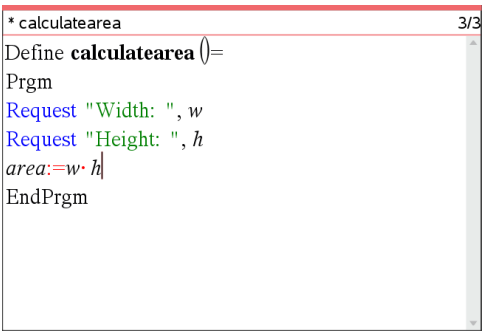
Nota: non occorre utilizzare i nomi dei parametri quando si esegue il programma **volcyl**, ma si devono specificare due argomenti (come valori, variabili o espressioni). Il primo deve rappresentare l'altezza e il secondo il raggio.

Richiesta di valori da parte dell'utente (solo programmi)

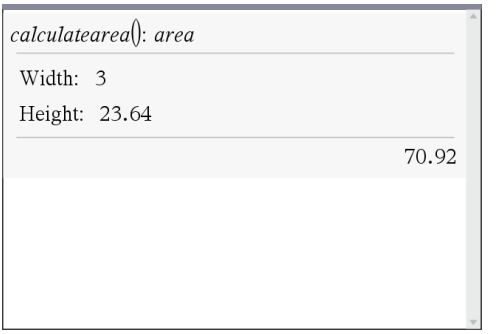
È possibile utilizzare i comandi **Request** e **RequestStr** in un programma per sospenderne l'esecuzione e visualizzare una finestra di dialogo con la richiesta di informazioni all'utente. Questo metodo non prevede che l'utente ricordi i nomi delle variabili o l'ordine in cui sono necessarie.

Non è possibile utilizzare il comando **Request** o **RequestStr** in una funzione.

1. Definire il programma.



2. Eseguire il programma e rispondere alle richieste.

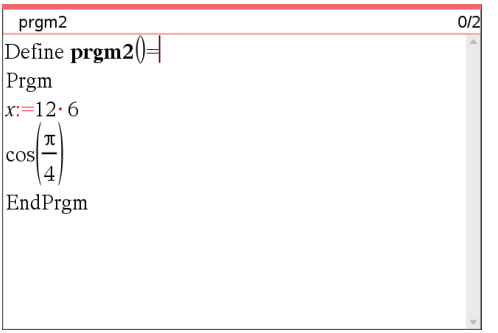


Utilizzare **RequestStr** invece di **Request** quando si desidera che il programma interpreti la risposta dell'utente come una stringa di caratteri anziché come un'espressione matematica. In questo caso l'utente non deve racchiudere la risposta tra virgolette (“”).

Visualizzazione delle informazioni

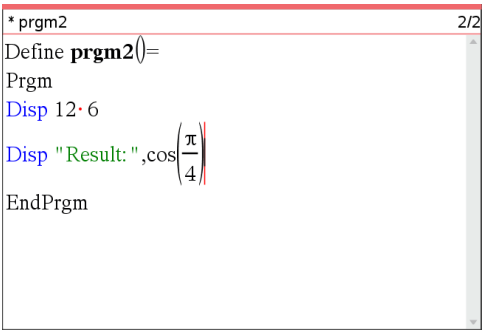
Una funzione o un programma in esecuzione non visualizza risultati intermedi a meno che non si includa un comando per visualizzarli. Questa è una differenza importante tra l'esecuzione di un calcolo sulla riga di introduzione o la sua esecuzione in una funzione o in un programma.

I seguenti calcoli, ad esempio, non visualizzano alcun risultato in una funzione o in un programma (ma se vengono eseguiti sulla riga di comando, il risultato appare).



Visualizzazione delle informazioni nella cronologia

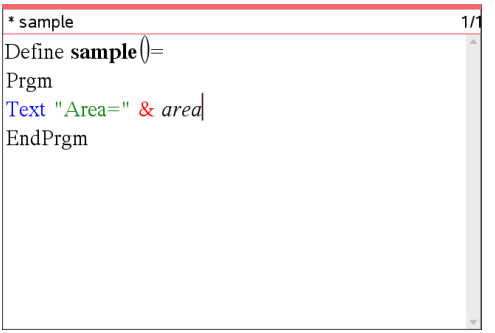
È possibile utilizzare il comando **Disp** in un programma o in una funzione per visualizzare informazioni, compresi i risultati intermedi, nella cronologia.



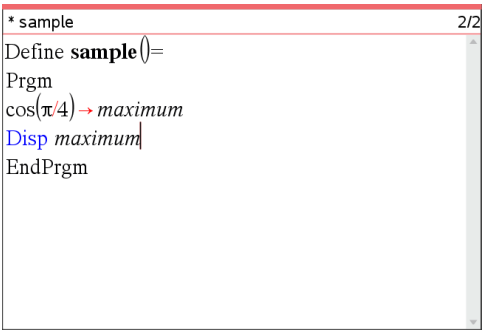
Visualizzazione di informazioni in una finestra di dialogo

È possibile utilizzare il comando **Text** per sospendere un programma in esecuzione e visualizzare informazioni in una finestra di dialogo. L'utente quindi seleziona **OK** per continuare oppure **Annulla** per arrestare il programma.

Non è possibile utilizzare il comando **Text** in una funzione.



Nota: Un risultato visualizzato con **Disp** o **Text** non viene automaticamente memorizzato. Se si pensa di richiamare successivamente un risultato, salvarlo in una variabile globale.

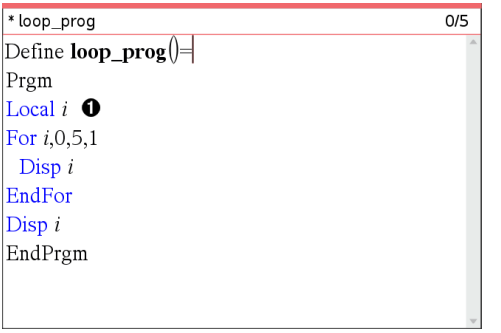


Utilizzo di variabili locali

Una variabile locale è una variabile temporanea che esiste solo durante il calcolo di una funzione definita dall'utente o durante l'esecuzione di un programma definito dall'utente.

Esempio di variabile locale

Il seguente segmento di programma mostra un'istruzione **For...EndFor loop** (descritta più avanti in questo capitolo). La variabile *i* è il contatore di loop. Nella maggior parte dei casi, la variabile *i* viene utilizzata solo mentre il programma è in esecuzione.



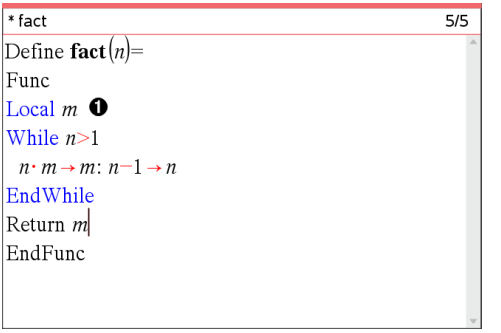
❶ Dichiara la variabile *i* come variabile locale.

Nota: quando possibile, dichiarare come locale qualsiasi variabile che viene utilizzata solo all'interno del programma e che non deve essere disponibile dopo il termine del programma.

Che cosa provoca un messaggio di errore Variabile non definita?

Un messaggio di errore Variabile **non definita** appare quando si calcola una funzione definita dall'utente o si esegue un programma definito dall'utente che chiama una variabile locale non inizializzata (nessun valore assegnato).

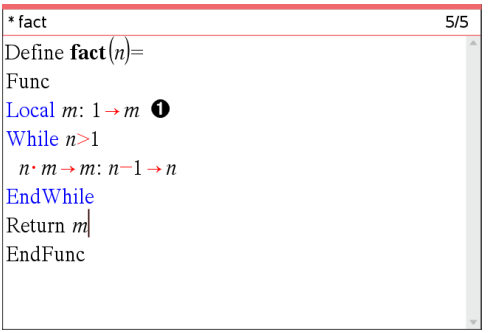
Ad esempio:



❶ Alla variabile locale *m* non è assegnato un valore iniziale.

Inizializzare variabili locali

A tutte le variabili locali deve essere assegnato un valore iniziale prima di essere chiamate.



❶ 1 viene memorizzato come valore iniziale di m.

Nota (CAS): funzioni e programmi non possono usare una variabile locale per eseguire calcoli simbolici.

CAS: Esecuzione di calcoli simbolici

Se si desidera che una funzione o un programma eseguano calcoli simbolici, è necessario utilizzare una variabile globale al posto di una locale. Tuttavia, occorre essere certi che la variabile globale non esista già al di fuori del programma. Possono essere utili i seguenti metodi.

- Utilizzare un nome di variabile globale, di solito con due o più caratteri, che probabilmente non esiste al di fuori della funzione o del programma.
- Includere **DelVar** all'interno di un programma per eliminare una variabile globale, se esiste, prima di chiamarla (**DelVar** non elimina variabili bloccate o collegate).

Differenze tra funzioni e programmi

Una funzione definita nell'Editor di programmi è molto simile alle funzioni create con il software TI-Nspire™.

- Le funzioni devono restituire un risultato, che può essere rappresentato graficamente o inserito in una tabella. I programmi non possono restituire un risultato.
- È possibile utilizzare una funzione (ma non un programma) all'interno di un'espressione. Ad esempio: **3 • func1(3)** è valida, ma non **3 • prog1(3)**.
- È possibile eseguire programmi solo dalle applicazioni Calcolatrice e Notes. Tuttavia, è comunque possibile calcolare funzioni in Calcolatrice, Notes, Foglio elettronico, Grafici e geometria, Dati e statistiche.

- Una funzione può chiamare qualsiasi variabile; tuttavia può memorizzare un valore solo in una variabile locale. I programmi possono essere memorizzati in variabili locali e globali.

  Nota: gli argomenti utilizzati per passare valori a una funzione vengono trattati automaticamente come variabili locali.
  Se si desidera effettuare il salvataggio in altre variabili, occorre dichiararle come **locali** direttamente nella funzione.

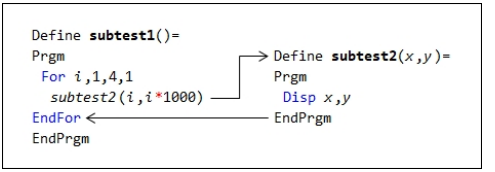
- Una funzione non può chiamare un programma come una sottoroutine, ma può chiamare un'altra funzione definita dall'utente.
- Non è possibile definire un programma all'interno di una funzione.
- Una funzione non può definire una funzione globale, ma può definire una funzione locale.

Chiamata di un programma da un altro programma

Un programma può chiamare un altro programma come una sottoroutine. La sottoroutine può essere esterna (un programma separato) o interna (inclusa nel programma principale). Le sottoroutine sono utili quando un programma deve ripetere lo stesso gruppo di comandi in più punti diversi.

Chiamata di un programma separato

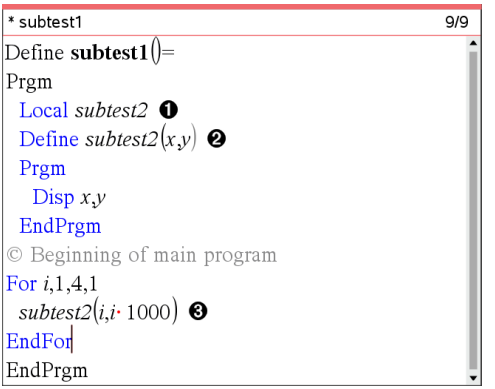
Per chiamare un programma separato, adottare la stessa sintassi che si utilizza per eseguire un programma dalla riga di introduzione.



Definizione e chiamata di una sottoroutine

Per definire una sottoroutine interna, utilizzare il comando **Define** con **Prgm...EndPrgm**. Poiché una sottoroutine deve essere definita prima di essere chiamata, è buona pratica definire le sottoroutine all'inizio del programma principale.

Una sottoroutine interna viene chiamata ed eseguita allo stesso modo di un programma separato.



❶ Dichiara la sottoroutine come variabile locale.

❷ Definisce la sottoroutine.

❸ Chiama la sottoroutine.

Nota: utilizzare il menu **Var** dell'Editor di programmi per inserire i comandi **Define** e **Prgm...EndPrgm**.

Note sull'uso delle sottoroutine

Alla fine di una sottoroutine, l'esecuzione ritorna al programma chiamante. Per uscire da una sottoroutine in qualsiasi momento, utilizzare il comando **Return** senza argomento.

Una sottoroutine non può accedere a variabili locali dichiarate nel programma chiamante. Analogamente, il programma chiamante non può accedere a variabili locali dichiarate in una sottoroutine.

I comandi Lbl sono locali nei programmi in cui sono inseriti. Di conseguenza, un comando **Goto** nel programma chiamante non può passare a un'etichetta in una sottoroutine o viceversa.

Come evitare errori di definizione circolare

Quando si calcola una funzione definita dall'utente o si esegue un programma, è possibile specificare un argomento che includa la stessa variabile che è stata utilizzata per definire la funzione o per creare il programma. Tuttavia, per evitare errori di definizione circolare è necessario assegnare un valore per variabili che vengono utilizzate nel calcolo della funzione o nell'esecuzione del programma. Ad esempio:

$x+1 \rightarrow x$ ❶

– Oppure –

```
For i,i,10,1
  Disp i ❶
EndFor
```

❶ Causa un messaggio di errore **Definizione circolare** se ad x o ad i non è stato assegnato alcun valore. L'errore non si produce se ad x o ad i è già stato assegnato un valore.

Controllo del flusso di una funzione o di un programma

Quando si esegue un programma o si calcola una funzione, le righe del programma vengono eseguite in ordine sequenziale. Tuttavia. alcuni comandi alterano il flusso del programma. Ad esempio:

- Le strutture di controllo, come ad esempio i comandi **If...EndIf**, utilizzano una prova condizionale per decidere quale parte eseguire di un programma.
- I comandi loop, come ad esempio **For...EndFor**, ripetono un gruppo di comandi.

Utilizzo di If, Lbl e Goto per controllare il flusso del programma

Il comando **If** e diverse strutture **If...EndIf** consentono di eseguire un'istruzione o un blocco di istruzioni condizionatamente, ossia in base al risultato di una prova (come ad esempio x>5). **I comandi Lbl** (etichetta) e **Goto** consentono di passare l'esecuzione a un altro comando o di saltare da un punto a un altro in una funzione o in un programma.

Il comando **If** e diverse strutture **If...EndIf** sono elencati nel menu **Strutture di controllo** dell'Editor di programmi.

Quando si introduce una struttura come **If...Then...EndIf**, alla posizione del cursore viene inserito un modello. Il cursore viene posizionato in modo tale che sia possibile inserire una prova condizionale.

Comando If

Per eseguire un solo comando quando una prova condizionale è vera, utilizzare la forma generale:

```
If x>5
  Disp "x is greater than 5" ❶
Disp x ❷
```

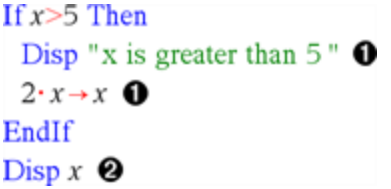
❶ Eseguito solo se x>5; altrimenti, saltato.

❷ Visualizza sempre il valore di x.

In questo esempio, si deve memorizzare un valore in x prima di eseguire il comando **If** .

Strutture If...Then...EndIf

Per eseguire un gruppo di comandi se una prova condizionale è vera, utilizzare la struttura:



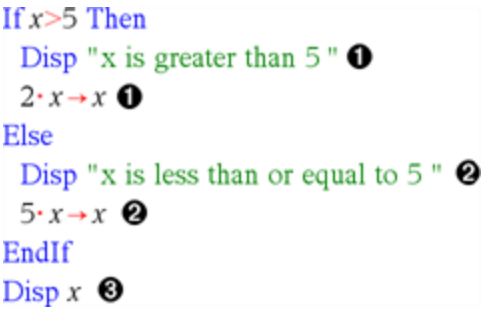
❶ Eseguito solo se x>5.

❷ Visualizza il valore di:
2x if x>5
x if x≤5

Nota: EndIf indica la fine del blocco **Then** che viene eseguito se la condizione è vera.

Strutture If...Then...Else... EndIf

Per eseguire un gruppo di comandi se una prova condizionale è vera e un altro gruppo se la condizione è falsa, utilizzare questa struttura:



❶ Eseguito solo se x>5.

❷ Eseguito solo se x≤5.

❸ Visualizza il valore di:
2x se x>5
5x se x≤5

Strutture If...Then...ElseIf... EndIf

Una forma più complessa del comando If consente di eseguire una prova per condizioni multiple. Si supponga che un programma debba sottoporre a prova un argomento fornito dall'utente con quattro opzioni possibili.

Per eseguire la prova per ciascuna opzione (If Opzione=1, If Opzione=2, e così via), utilizzare la struttura **If...Then...ElseIf...EndIf**.

Comandi Lbl e Goto

È possibile controllare il flusso anche utilizzando i comandi **Lbl** (etichetta) e **Goto**. Questi comandi sono elencati nel menu **Trasferimenti (Transfers)** dell'Editor di programmi.

Utilizzare il comando **Lbl** per etichettare (ossia assegnare un nome a) una posizione specifica nella funzione o nel programma.

Lbl *nomeEtichetta*	nome da assegnare a questa posizione (utilizzare le stesse convenzioni di assegnazione dei nomi di variabile)

Successivamente, sarà possibile utilizzare il comando **Goto** in un punto qualsiasi della funzione o del programma per passare l'esecuzione alla posizione corrispondente all'etichetta specificata.

Goto *nomeEtichetta*	specifica a quale comando **Lbl** deve passare l'esecuzione

Poiché **Goto** è un comando non condizionale (passa sempre l'esecuzione all'etichetta specificata), viene spesso utilizzato con un comando **If** per consentire la specifica di una prova condizionale. Ad esempio:

```
If x>5
  Goto GT5 ❶
Disp x

.............

Lbl GT5 ❷
Disp "The number was > 5"
```

❶ If x>5 passa l'esecuzione direttamente all'etichetta GT5.

❷ In questo esempio, il programma deve includere comandi (come ad esempio **Stop**) che impediscano l'esecuzione di **Lbl** GT5 se x≤5.

Utilizzo di loop per ripetere un gruppo di comandi

Per ripetere lo stesso gruppo di comandi in successione, utilizzare una delle strutture loop. Sono disponibili diversi tipi di loop. Ciascun tipo offre un modo diverso di uscita dal loop, sulla base di una prova condizionale.

I comandi loop e relativi ai loop sono elencati nei menu **Strutture di controllo (Control)** e **Trasferimenti (Transfers)** dell'Editor di programmi.

Quando si inserisce una delle strutture loop, alla posizione del cursore viene inserito il relativo modello. È quindi possibile iniziare a introdurre i comandi che verranno eseguiti all'interno del loop.

Loop For...EndFor

Un loop **For...EndFor** utilizza un contatore per controllare il numero di volte che il loop viene ripetuto. La sintassi del comando **For** è:

Nota: il valore finale può essere minore del valore iniziale, purché l'incremento sia negativo.

For *variabile*, *inizio*, *fine* [, [, *incremento*]
❶ ❷ ❸ ❹

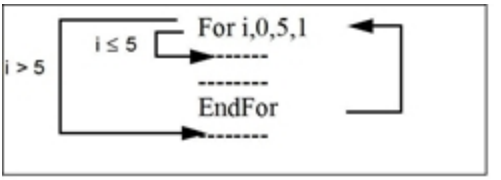
❶ *Variabile* è utilizzata come contatore

❷ Valore del contatore utilizzato la prima volta che viene eseguito **For**

❸ Esce dal loop quando *variabile* supera questo valore

❹ Aggiunto al contatore ad ogni successiva esecuzione di **For** (se questo valore opzionale viene omesso, *incremento* è 1).

Quando viene eseguito **For**, il valore di *variabile* viene confrontato con il valore di *finale*. Se *variabile* non è maggiore di *finale*, il loop viene eseguito; altrimenti l'esecuzione passa al successivo comando **EndFor**.



Nota: il comando **For** incrementa automaticamente la variabile contatore in modo che la funzione o il programma possa uscire dal loop dopo un certo numero di ripetizioni.

Alla fine del loop (**EndFor**), l'esecuzione torna indietro al comando **For**, dove la variabile viene incrementata e confrontata con *finale*.

Ad esempio:

```
For i,0,5,1
  Disp i ❶
EndFor
Disp i ❷
```

❶ Visualizza 0, 1, 2, 3, 4 e 5.

❷ Visualizza 6. Quando *variabile* arriva a 6, il loop non viene eseguito.

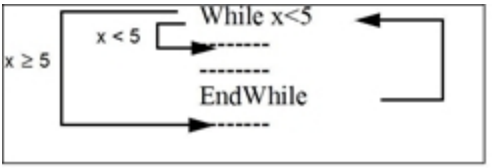
Nota: è possibile dichiarare la variabile contatore come locale se non deve essere salvata al termine della funzione o del programma.

Loop While...EndWhile

Un loop **While...EndWhile** ripete un blocco di comandi fino a quando una condizione specificata non è vera. La sintassi del comando **While** è:

While *condizione*

Quando **While** viene eseguito, la *condizione* viene controllata. Se la *condizione* è vera, il loop viene eseguito; altrimenti l'esecuzione passa al successivo comando **EndWhile**.



Nota: il comando **While** non cambia automaticamente la condizione. È necessario includere comandi che consentano alla funzione o al programma di uscire dal loop.

Alla fine del loop (**EndWhile**), l'esecuzione torna indietro al comando **While**, dove la condizione viene ricontrollata.

Per eseguire il loop la prima volta, la condizione deve essere inizialmente vera.

- Qualsiasi variabile chiamata nella condizione deve essere impostata prima del comando **While** (è possibile integrare i valori nella funzione o nel programma, oppure è possibile richiedere all'utente di inserirli).
- Il loop deve contenere comandi che cambiano i valori nella condizione, causandone alla fine una condizione falsa. Diversamente, la condizione è sempre vera e la funzione o il programma non possono uscire dal loop (loop infinito).

Ad esempio:

```
0→x ❶
While x<5
  Disp x ❷
  x+1→x ❸
EndWhile
Disp x ❹
```

❶ Imposta inizialmente x.

❷ Visualizza 0, 1, 2, 3 e 4.

❸ Incrementa x.

❹ Visualizza 5. Quando x incrementa a 5, il loop non viene eseguito.

Loop Loop...EndLoop

Loop...EndLoop crea un ciclo infinto, che si ripete senza fine. Il comando **Loop** non ha argomento.

In genere, nel loop si inseriscono comandi che consentono al programma di uscire dal loop. I comandi solitamente utilizzati sono: **If**, **Exit**, **Goto** e **Lbl** (etichetta). Ad esempio:

```
0→x
Loop
  Disp x
  x+1→x
  If x>5 ❶
    Exit
EndLoop
Disp x ❷
```

❶ Un comando **If** verifica la condizione.

❷ Esce dal loop e passa direttamente qui quando x incrementa a 6.

Nota: il comando **Exit** esce dal loop corrente.

In questo esempio, il comando **If** può essere in qualsiasi punto del loop.

Quando il comando If è	**Il loop viene**
All'inizio del loop	Eseguito solo se la condizione è vera.
Alla fine del loop	Eseguito almeno una volta e ripetuto solo se la condizione è vera.

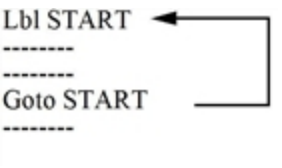
Il comando **If** può utilizzare anche un comando **Goto** per trasferire il controllo del programma a un comando **Lbl** (etichetta) specificato.

Ripetizione immediata di un loop

Il comando **Cycle** trasferisce immediatamente il controllo del programma alla successiva iterazione di un loop (prima che l'iterazione corrente sia completata). Questo comando funziona come **For...EndFor**, **While...EndWhile** e **Loop...EndLoop**.

Loop Lbl e Goto

Anche se i comandi **Lbl** (etichetta) e **Goto** non sono esattamente comandi loop, possono essere utilizzati per creare un loop infinito. Ad esempio:



Come con **Loop...EndLoop**, il loop dovrebbe contenere comandi che consentano alla funzione o al programma di uscire dal loop.

Modifica delle impostazioni di modalità

Funzioni e programmi possono utilizzare la funzione **setMode()** per impostare temporaneamente specifiche modalità di calcolo o risultato. Il menu **Modo (Mode)** dell'Editor di programmi consente di inserire facilmente la sintassi corretta senza dover memorizzare codici numerici.

Nota: i cambi di modalità effettuati nella definizione di una funzione o di un programma non persistono al di fuori della funzione o del programma.

Impostazione di una modalità

1. Posizionare il cursore nel punto in cui si desidera inserire la funzione **setMode**.
2. Nel menu **Mode (Modalità)**, selezionare la modalità da cambiare, quindi selezionare la nuova impostazione.

   La sintassi corretta viene inserita alla posizione del cursore. Ad esempio:

```
setMode(1,3)
```

Debug di programmi e gestione degli errori

Dopo aver scritto una funzione o un programma, è possibile utilizzare varie tecniche per trovare e correggere errori. È possibile anche inserire un comando di gestione degli errori all'interno della funzione o del programma stesso.

Se la funzione o il programma consente all'utente di effettuare una selezione tra diverse opzioni, accertarsi di eseguire e sottoporre a prova ciascuna opzione.

Tecniche di debug

Messaggi di errore run-time possono individuare errori di sintassi, ma non errori nella logica del programma. Possono essere utili le seguenti tecniche.

- Inserire temporaneamente comandi **Disp** per visualizzare i valori delle variabili critiche.
- A conferma dell'esecuzione di un loop il numero di volte corrette, utilizzare il comando **Disp** per visualizzare la variabile contatore o i valori nella prova condizionale.
- A conferma dell'esecuzione di una sottoroutine, utilizzare il comando **Disp** per visualizzare messaggi quali "Entering subroutine" e "Exiting subroutine" all'inizio e alla fine della sottoroutine.
- Per arrestare manualmente un programma o una funzione:
  - Windows®: Tenere premuto il tasto **F12** e premere **Enter** più volte.
  - Macintosh®: Tenere premuto il tasto **F5** e premere **Enter** più volte.
  - Palmare: Tenere premuto il tasto [on] e premere [enter] più volte.

Gestione dei comandi di errore

Comando	Descrizione
Try...EndTry	Definisce un blocco che consente a una funzione o a un programma di eseguire un comando e, se necessario, di recuperare da un errore generato da tale comando.
ClrErr	Cancella lo stato di errore e azzera il numero di errori nella variabile di sistema *errCode*. Per un esempio dell'utilizzo di *errCode*, vedere il comando **Try** nella *Guida di riferimento*.
PassErr	Passa un errore al livello successivo del blocco **Try...EndTry**.

Informazioni Generali

Guida online

education.ti.com/eguide

Selezionare il proprio Paese per maggiori informazioni sul prodotto.

Contattare l'assistenza TI

education.ti.com/ti-cares

Selezionare il proprio Paese per assistenza tecnica e altre risorse.

Informazioni su servizi e garanzia

education.ti.com/warranty

Selezionare il proprio Paese per informazioni sulla durata e sui termini della garanzia o sull'assistenza ai prodotti.

Garanzia limitata. La presente garanzia non pregiudica i diritti spettanti per legge.

Texas Instruments Incorporated

12500 TI Blvd.

Dallas, TX 75243