TI-Nspire™程序編輯指南

請透過 education.ti.com/eguide 的線上說明，瞭解更多有關 TI 技術的資訊。

重要信息

除程序附帶的許可證另有明確規定外，德州儀器不作任何明示或暗示的保證，包括但不限於對特定目的的適銷性和適用性的隱含保證，涉及任何程序或書籍材料，材料只能按“原樣”提供。在任何情況下，德州儀器不對任何人因購買或使用這些材料而造成的特殊，抵押，偶然或後果的損害負責，也不包括德州儀器的唯一和唯一的責任。行動不得超過計劃許可證規定的金額。此外，德州儀器不對任何其他方使用這些材料的任何索賠提出任何責任。

TI-Nspire™軟件使用Lua作為腳本環境。有關版權和許可證信息，請訪問 http://www.lua.org/license.html。

TI-Nspire™軟件使用Chipmunk Physics 5.3.4版作為仿真環境。有關許可證信息，請參見http://chipmunk-physics.net/release/Chipmunk-5.x/Chipmunk-5.3.4-Docs/。

Microsoft®和Windows®是Microsoft Corporation在美國和/或其他國家/地區的註冊商標。

MacOS®，iPad®和OSX®是Apple Inc.的註冊商標。

Unicode®是Unicode，Inc.在美國和其他國家/地區的註冊商標。

產品實物可能與所提供的圖片資料略有不同。

內容

[程式編輯器] 快速入門

您可以在 計算工具輸入列上輸入定義語句，或使用 [程式編輯器]，來建立使用者自行定義的函數或程式。[程式編輯器] 提供了一些好處，將會在本節中介紹。如需更多資訊，請參閱*計算工具*。

- 該編輯器提供程式設計範本和對話方塊，以協助您利用正確的語法來定義函數與程式。
- 該編輯器可讓您輸入多行程式設計語句，而不需要使用特殊的按鍵順序來新增每一行。
- 您可以輕鬆建立自訂和公用的資料庫物件(變數、函數和程式) 。如需詳細資訊，請參閱*資料庫*。

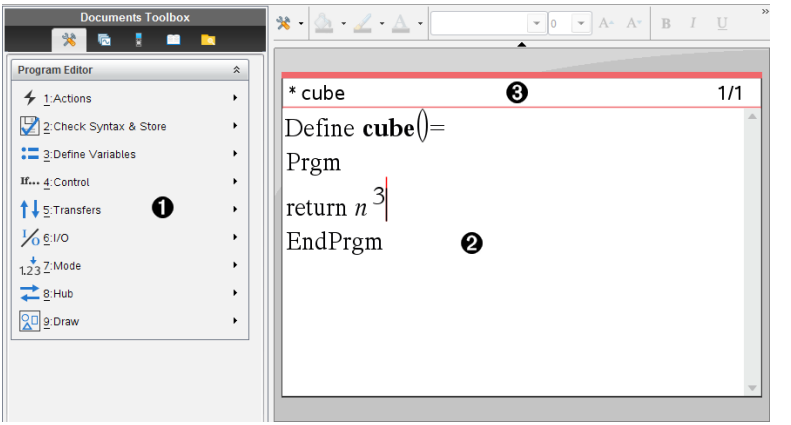
啟動 [程式編輯器]

▶ 若要在目前問題中新增新的 [程式編輯器] 頁面:

在工具列，按一下**插入>程式編輯器>新增**。

計算機: 按 doc▾，然後選取**插入 > 程式 編輯器 >新增**。

附註: 也可透過 [計算工具] 頁面的**功能&程式**功能表存取編輯器。



❶ [程式編輯器] 功能表 - 只要您位於 [程式編輯器] 工作區域並使用「正常」畫面模式，即可隨時使用這個功能表。

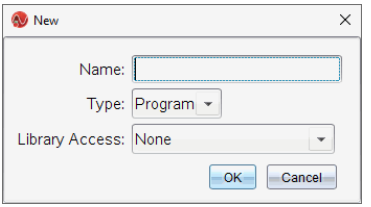
❷ [程式編輯器] 工作區域

❸ 狀態行顯示行號資訊與正在編輯的函數或程式的名稱。星號 (*) 會指明該函數是「已更改」的狀態，表示自從上次檢查其語法並儲存該函數後，已經有所更改。

定義程式或函數

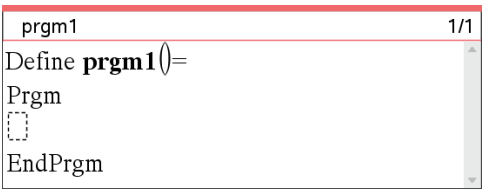
啟動新的 [程式編輯器]

1. 請確認您已開啟要在其中建立程式或函數的文件和問題。
2. 按一下應用程式工具列上的 **[插入]** 按鈕，然後選取 **[程式編輯器] > [新建]**。(在計算機上，按一下 doc，然後選取 **[插入] > [程式編輯器] > [新建]**。



3. 輸入您正在定義的函數或程式名稱。
4. 選取 **[類型]([程式]** 或 **[函數])** 。
5. 設定 **[資料庫存取]**:
   - 若只想使用目前文件與問題中的函數或程式，請選取 **[無]**。
   - 若想從任何文件存取函數或程式，但不在 [目錄] 中顯示，請選取 **LibPriv**。
   - 若想從任何文件存取函數或程式，並在 [目錄] 中顯示，請選取 **LibPub (在目錄中顯示)** 。如需詳細資訊，請參閱*資料庫*。
6. 按一下 **[確定]**。

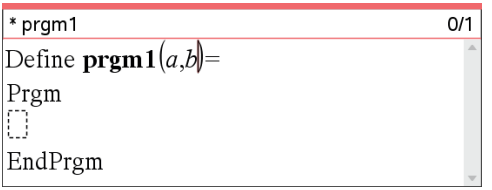
   系統會開啟 [程式編輯器] 的新實例，含有您所選取的相符範本。



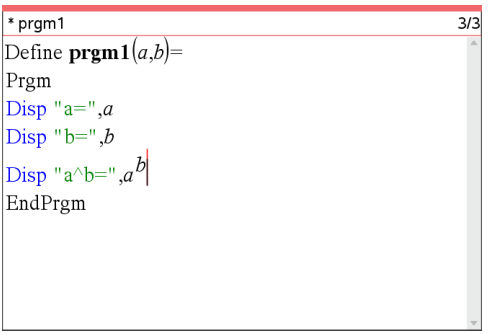
在函數或程式中輸入指令行

[程式編輯器] 不會在您輸入指令或運算式時即執行指令或計算。系統只會在您計算函數或執行程式時才會執行這些動作。

1. 如果您的函數或程式需要使用者提供引數，請在名稱後的括弧內輸入參數名稱。以逗號將參數隔開。



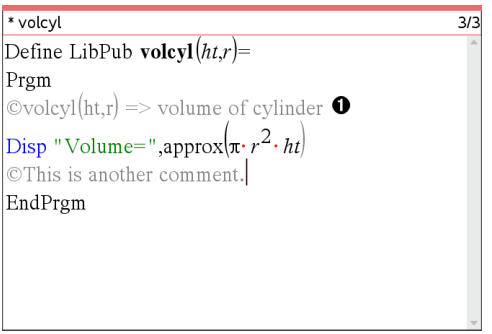
2. 在 Func 與 EndFunc(或 Prgm 和 EndPrgm) 行之間，輸入組成您函數或程式的陳述式。



- 您可以輸入函數或指令的名稱，或從「目錄」中進行插入。
- 某些行的長度可能會超過畫面寬度;在此情況下，您可能需要捲動畫面以檢視完整語句。
- 輸入每一行之後，請按 **Enter**。這會插入新的空白行，並讓您繼續輸入其他行。
- 使用 ◄、►、▲ 和 ▼ 方向鍵捲動以檢視函數或程式的內容，進而輸入或編輯指令。

插入註解

註解對於檢視或編輯程式很有用。程式執行時不會顯示註解，註解亦不會影響程式的流程。© 符號會在註解行開頭處顯示。



❶ 顯示所需語法的註解。由於此資料庫物件可以公用，且此註解位於 Func 或 Prgm 區塊中的第一行，因此註解會以說明形式顯示在 [目錄] 中。如需詳細資訊，請參閱*資料庫*。

插入註解:

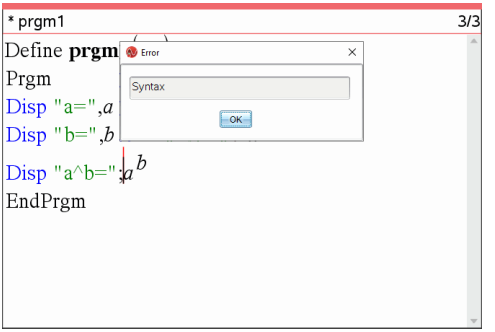
1. 將游標移到您要插入註解的內容行結尾。
2. 在 **[動作]** 功能表中按一下 **[插入註解]**，或按 **Ctrl+T**。
3. 在 © 符號後輸入註解文字。

檢查語法

[程式編輯器] 會讓您檢查函數或程式的語法是否正確。

- 在 **[檢查語法並儲存]** 功能表中，按一下 **[檢查語法]**。

  如果語法檢查器發現任何語法錯誤，會顯示一個錯誤訊息，並嘗試將游標放在第一個錯誤附近，讓您能夠進行更正。



儲存函數或程式

您必須儲存函數或程式才能在之後叫出。[程式編輯器] 會在儲存之前自動檢查語法。

[程式編輯器] 的左上角會顯示一個星號(*)，指明未儲存函數或程式。

- 在 **[檢查語法並儲存]** 功能表中，按一下 **[檢查語法並儲存]**。

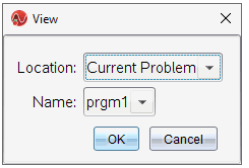
  如果語法檢查器發現任何語法錯誤，會顯示一個錯誤訊息，並嘗試將游標放在第一個錯誤附近。

  如果沒有發現語法錯誤，[程式編輯器] 頂端的狀態行中會顯示「順利儲存成功」訊息。

注意:如果函數或程式定義為資料庫物件，您也必須將文件儲存在指定的資料庫資料夾中，並重新整理資料庫，讓物件可被其他文件使用。如需詳細資訊，請參閱*資料庫*。

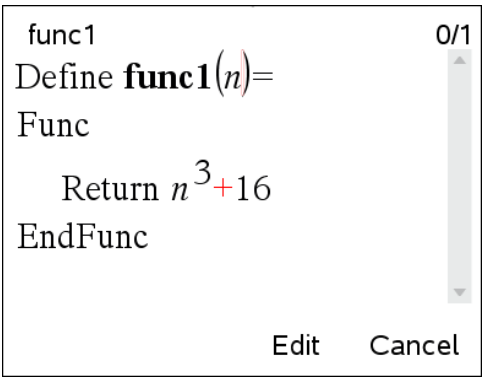
檢視程式或函數

1. 在 **[動作]** 功能表中選取 **[檢視]**。



2. 如果函數或程式屬於資料庫物件，請從 **[位置]** 列表中選取其資料庫。
3. 從 **[名稱]** 列表中選取函數或程式名稱。

   函數或程式會顯示在瀏覽程式中。



4. 使用方向鍵來檢視函數或程式。
5. 如果您想編輯程式，請按一下 **[編輯]**。

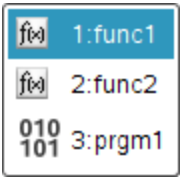
   附註：只有在目前問題中定義的函數與程式才能選取 **[編輯]**。若要編輯資料庫物件，您必須先開啟其資料庫文件。

開啟函數或程式以供編輯

您只能從目前的問題中開啟函數或程式。

附註：您無法修改鎖定的程式或函數。若要解開鎖定物件，請前往計算工具頁面，然後使用 **unLock** 指令。

1. 顯示可用的函數與程式的列表。
   - 在 **[動作]** 功能表中選取 **[開啟]**。

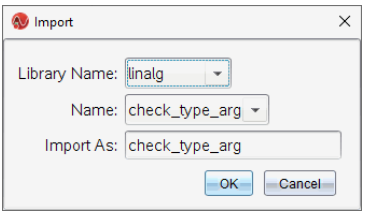


2. 選取要開啟的項目。

從資料庫匯入程式

您可以將定義為資料庫物件的函數或程式匯入目前問題中的 [程式編輯器]。匯入的副本不會被鎖定，即使原來的物件是鎖定狀態。

1. 在 **[動作]** 功能表中選取 **[匯入]**。



2. 選取 **[資料庫名稱]**。
3. 選取物件的 **[名稱]**。
4. 如果您希望使用不同的物件名稱，請在 **[匯入成]** 下方輸入名稱。

建立函數或程式的副本

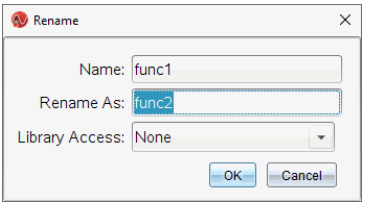
建立新的函數或程式時，您會發現從目前的函數或程式副本開始比較輕鬆。您建立的副本不會被鎖定，即使原來的物件是鎖定狀態。

1. 在 **[動作]** 功能表中選取 **[建立副本]**。
2. 輸入新名稱，然後按一下 **[確定]** 來接受提出的名稱。
3. 如果您想更改存取層次，請選取 **[資料庫存取]**，然後選取新的層次。

重新命名程式或函數

您可以重新命名及(也可以) 更改目前函數或程式的存取層次。

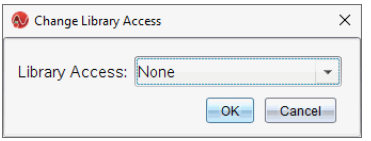
1. 在 **[動作]** 功能表中選取 **[重新命名]**。



2. 輸入新名稱，然後按一下 **[確定]** 來接受提出的名稱。
3. 如果您想更改存取層次，請選取 **[資料庫存取]**，然後選取新的層次。

更改資料庫存取層次

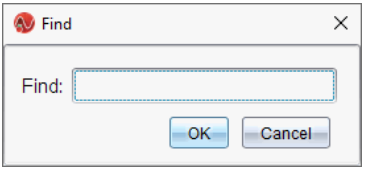
1. 在 **[動作]** 功能表中選取 **[更改資料庫存取權限]**。



2. 選取 **[資料庫存取]**：
   - 若只想使用目前計算工具問題中的函數或程式，請選取 **[無]**。
   - 若想從任何文件叫出不在「目錄」中的函數或程式，請選取 **LibPriv**。
   - 若想從任何文件叫出在「目錄」中的函數或程式，請選取 **LibPub**。

尋找文字

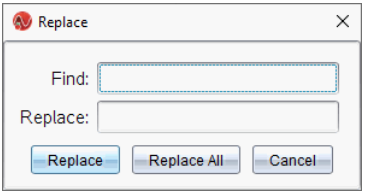
1. 在 **[動作]** 功能表中選取 **[尋找]**。



2. 輸入您想尋找的文字，然後按一下 **[確定]**。
   - 如果系統有找到文字，則會在程式中標示出來。
   - 如果系統沒有找到文字，則會顯示一個通知訊息。

尋找及取代文字

1. 在 **[動作]** 功能表中選取 **[尋找及取代]**。



2. 輸入您要尋找的文字。
3. 輸入取代文字。
4. 按一下 **[取代]** 以取代游標位置之後的第一個出現的項目，或按一下 **[全部取代]** 以取代所有出現項目。

   附註：如果是在數學範本中找到文字，會顯示一個訊息，警告您取代文字將取代整個範本，而不只是找到的文字。

關閉目前的函數或程式

- 在 **[動作]** 功能表中選取 **[關閉]**。

如果函數或程式有未儲存的更改內容，系統會提示您在關閉之前檢查語法並儲存。

執行程式與對函數求值

在定義並儲存程式或函數後，就可在應用程式中使用。所有應用程式都可對函數求值，但只有計算工具及筆記應用程式才能執行程式。

系統會依照順序執行程式語句(即使某些指令會更改程式的流程)。如果有任何輸出，則會顯示在應用程式的工作區域。

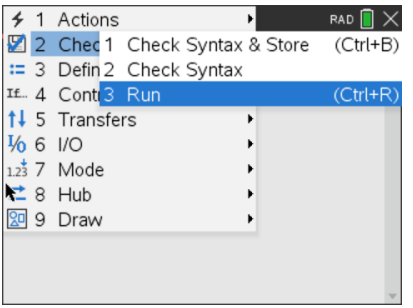
- 程式會繼續執行，直到達到最後一個語句或 **Stop** 指令。
- 函數會繼續執行，直到達到 **Return** 指令。

從程式編輯器執行程式或函數

1. 請確定您已定義程式或函數，且「程式編輯器」為作用中面板(電腦)或頁面(計算機)。

2. 在工具列上，按一下「**文件工具**」按鈕  ，然後選取「**檢查語法並儲存 > 執行**」。
   —或—
   按 **Ctrl+R**。

   計算機:按 menu 2 3，或按 ctrl R。



   這個動作會自動:

   - 檢查語法並儲存程式或函數，
   - 將程式或函數名稱貼到緊接「程式編輯器」的「計算工具」應用程式的第一個可用行。如果該位置沒有「計算工具」，將會插入新的「計算工具」。

3. 如果程式或函數需要您提供一或多個引數，請在括弧內輸入值或變數名稱。
4. 按 enter。

注意:您也可輸入帶括弧的程式名稱以及任何需要的引數，然後按 enter，在「計算工具」或「筆記」應用程式中執行程式或函數。

使用簡短或完整名稱

只要您位於定義物件的相同問題中，就可以輸入物件的簡短名稱(在物件的 **Define** 指令中提供的名稱) 來叫出該物件。這對所有定義的物件來說都適用，包括自訂、公用及非資料庫的物件。

您可以輸入物件的完整名稱，從任何文件中叫出資料庫物件。完整名稱包括物件的資料庫文件名稱，之後接一個反斜線「\」，再接著物件的名稱。例如，在資料庫文件 **lib1** 中定義為 **func1** 的物件的完整名稱是 **lib1\func1**。若要在計算機上輸入「\」字元，請按 shift ÷。

注意:如果您記不住自訂資料庫物件所需的精確名稱或引數順序，可以開啟資料庫文件，或使用「程式編輯器」來檢視物件。您也可以使用 **getVarInfo** 來檢視資料庫中的物件列表。

使用公用資料庫程式或函數

1. 請確認您已在文件的第一個問題中定義物件、已儲存物件、將資料庫文件儲存在 MyLib 資料夾中，並且已重新整理資料庫。
2. 開啟您想在其中使用程式或函數的 TI-Nspire™ 應用程式。

   注意:所有應用程式都能對函數求值，但只有「計算工具」和「筆記」應用程式能執行程式。
3. 開啟「目錄」，然後使用資料庫標籤來尋找並插入物件。
   —或—
   輸入物件名稱。針對程式或函數，一律在名稱後加上括弧。

   ```
   libs2\func1()
   ```
4. 如果程式或函數需要您提供一或多個引數，請在括弧內輸入值或變數名稱。

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

5. 按 enter。

使用自訂資料庫程式或函數

若要使用自訂資料庫物件，您必須知道它的完整名稱。例如，在資料庫文件 **lib1** 中定義為 **func1** 的物件的完整名稱是 **lib1\func1**。

注意:如果您記不住自訂資料庫物件所需的精確名稱或引數順序，可以開啟資料庫文件，或使用「程式編輯器」來檢視物件。

1. 請確認您已在文件的第一個問題中定義物件、已儲存物件、將資料庫文件儲存在 MyLib 資料夾中，並且已重新整理資料庫。
2. 開啟您想在其中使用程式或函數的 TI-Nspire™ 應用程式。

   注意:所有應用程式都能對函數求值，但只有「計算工具」和「筆記」應用程式能執行程式。
3. 輸入物件名稱。針對程式或函數，一律在名稱後加上括弧。

   ```
   libs2\func1()
   ```
4. 如果物件需要您提供一或多個引數，請在括弧內輸入值或變數名稱。

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

中斷執行中的程式或函數

當程式或函數在執行中時，系統會顯示忙碌中的游標 ⧖。

- 停止程式或函數:
  - Windows®:按住 **F12** 鍵並重複按 **Enter** 鍵。
  - Mac®:按住 **F5** 鍵並重複按 **Enter** 鍵。
  - 計算機:按住 ⌂ on 鍵並重複按 · 鍵。

  系統會顯示一則訊息。若要在「程式編輯器」中編輯程式或函數，請選取「**前往**」。游標會出現在發生中斷的指令處。

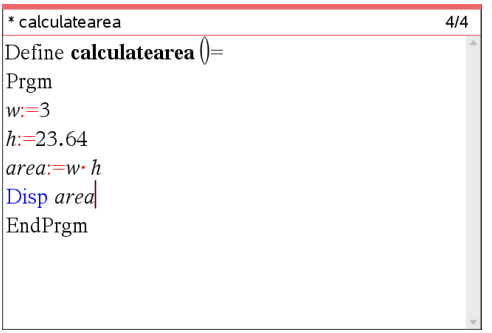
將值輸入程式

您有數種方法可以選擇函數或程式要在計算中使用的值。

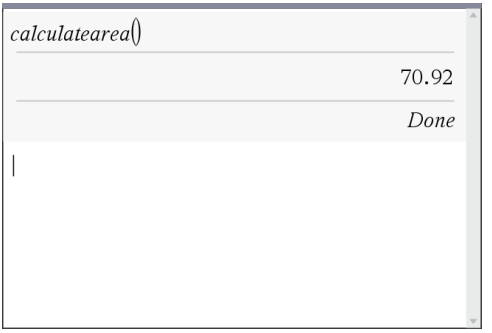
將值嵌入程式或函數

這種方法主要對於每次使用程式或函數時，值必須相同的情況很有用。

1. 定義程式。



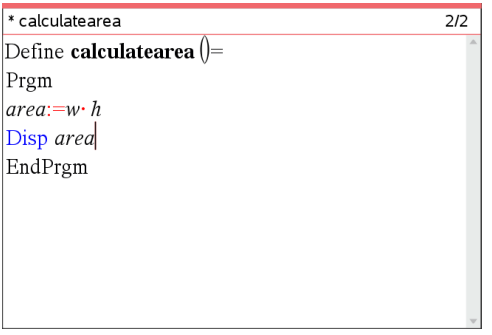
2. 執行程式。



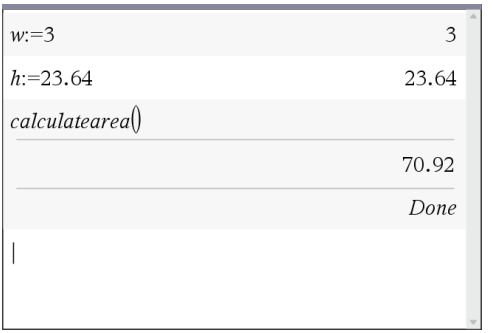
讓使用者為變數指定值

程式或函數可參照之前建立的變數。這種方法需要使用者記得變數名稱，並在使用物件之前為這些變數指定值。

1. 定義程式。



2. 提供變數，然後執行程式。

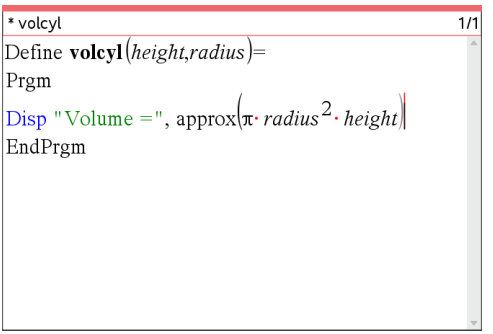


讓使用者提供值作為引數

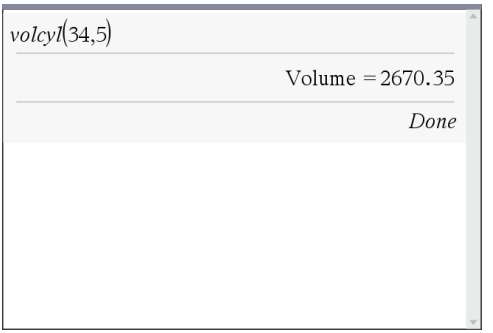
這種方法讓使用者傳遞一或多個值，作為會呼叫程式或函數的運算式內的引數。

下列程式 **volcyl** 會計算圓柱的體積。它需要使用者提供兩個值：圓柱的高度與半徑。

1. 定義 **volcyl** 程式。



2. 執行程式以顯示高度 34 公厘與半徑 5 公厘的圓柱體積。



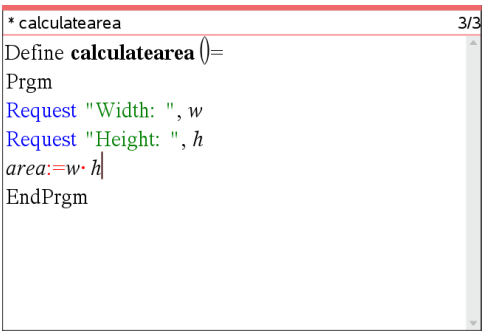
附註：當您執行 **volcyl** 程式時，不需要使用參數名稱，但必須提供兩個引數(作為值、變數或運算式)。第一個引數必須代表高度，而第二個必須代表半徑。

要求使用者提供值(僅限程式)

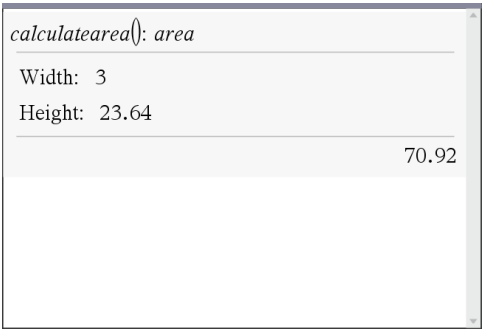
您可以使用程式中的 **Request** 與 **RequestStr** 指令來暫停程式，並顯示一個對話方塊來提示使用者提供資訊。這種方法不需要使用者記住變數名稱或按照順序輸入。

您無法在函數中使用 **Request** 或 **RequestStr** 指令。

1. 定義程式。



2. 執行程式並回應要求。

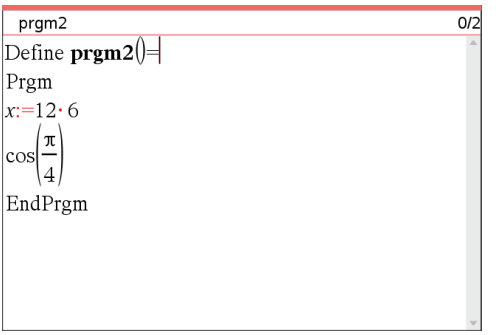


當您希望程式將使用者的回應解譯為字元字串而非數學運算式時，請使用 **RequestStr** 來取代 **Request**。這可以讓使用者不必用引號("") 來括住回應。

顯示資訊

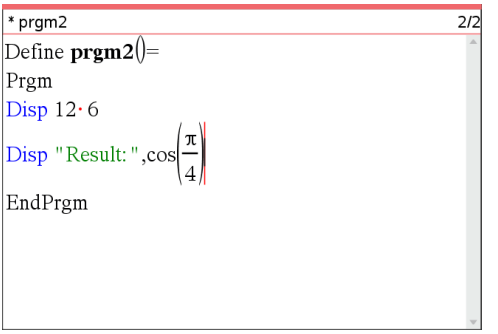
執行中的函數或程式不會顯示中間的計算結果，除非您在其中包括要顯示結果的指令。這是在輸入列上執行計算以及在函數或程式中執行計算的重要差異。

例如，下列計算不會在函數或程式中顯示計算結果(即使他們會從輸入列顯示)。



在歷史記錄中顯示資訊

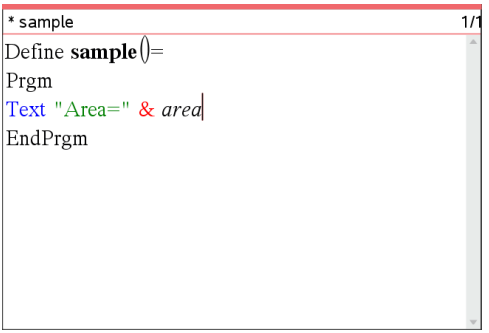
您可以在程式或函數中使用 **Disp** 指令，在歷史記錄中顯示包括中間計算結果的資訊。



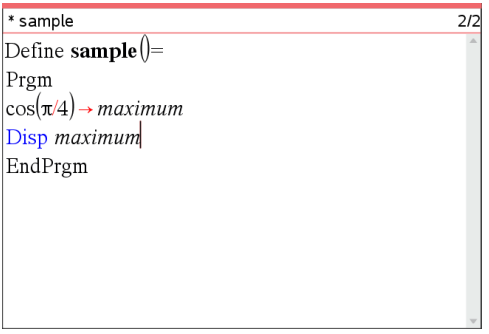
在對話方塊中顯示資訊

您可以使用 **Text** 指令來暫停執行中的程式，並在對話方塊中顯示資訊。使用者可選取 [確定] 以繼續進行，或選取 [取消] 來停止程式。

您無法在函數中使用 **Text** 指令。



附註：利用 **Disp** 或 **Text** 顯示計算結果並不會將結果儲存下來。如果您預期在之後會參照該結果，請將其儲存至全域變數。

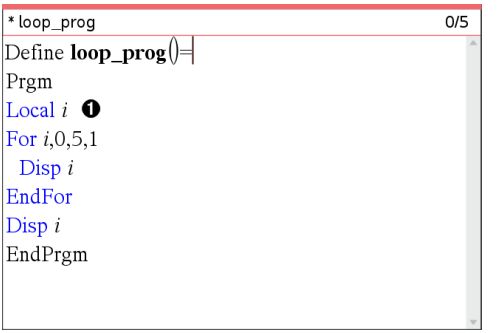


使用區域性變數

區域性變數是暫時性的變數，只於正在計算使用者自行定義的函數，或正在執行使用者自行定義的程式時存在。

區域變數的範例

下列程式區段顯示 **For...EndFor loop(** 會在稍後於此模組中討論**)**。變數 *i* 是迴圈計數器。在大多數情況下，變數 *i* 只會用於正在執行程式時。



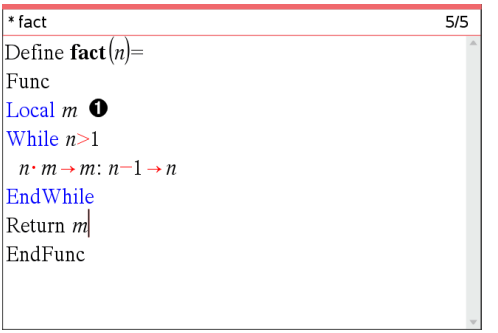
❶ 將變數 *i* 宣告為區域性變數。

附註：可以的話，將任何只在程式內使用，且不需要在程式停止後仍可供使用的變數，都宣告為區域性變數。

導致未定義變數錯誤訊息的原因

當您計算使用者自行定義的函數，或執行使用者自行定義的程式，而該函數或程式會參照未被賦予初值**(** 被指定值**)** 的區域性變數時，便會顯示 **Undefined** 變數錯誤訊息。

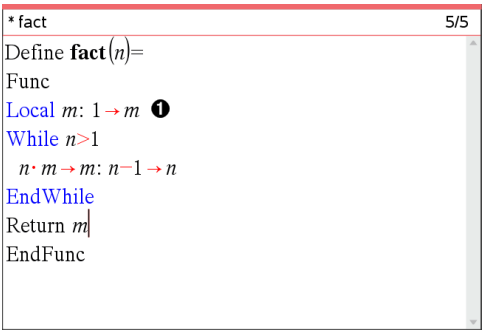
例如：



❶ 區域性變數 *m* 沒有被指定初值。

初始化區域變數

所有的區域變數在被參照之前，都必須被賦予初值。



❶ 1 會被儲存為 *m* 的初值。

附註 (CAS) : 函數和程式無法使用區域性變數來執行符號計算。

CAS: 執行符號計算

如果您希望函數或程式執行符號計算，則必須使用全域變數而非區域性變數。不過，您必須確認全域變數尚未存在於程式外部。下列是有益的方法。

- 一般以二個以上的字元來參照全域變數名稱，而該變數名稱應該未存在於函數或程式的外部。
- 在程式中包括 **DelVar** 來刪除存在的全域變數，再參照該變數。(**DelVar** 不會刪除鎖定或連結的變數。)

函數與程式間的差異

定義於 [程式編輯器] 中的函數類似於建置於 TI-Nspire™ 軟體中的函數。

- 函數必須傳回計算結果，可繪製成圖形或輸入表格中。程式無法傳回計算結果。
- 您可以在運算式內使用函數，但無法使用程式。例如：**3 • func1(3)** 是有效的，但 **3 • prog1(3)** 則無效。
- 只能從[計算工具]和[筆記]應用程式執行程式。不過，可以在[計算工具]、[筆記]、[序列 & 試算表]、[函數繪圖 & 幾何作圖]和[數據 & 統計]中計算函數。
- 函數可以參照任何變數；不過，它只能將值儲存到區域性變數。程式可以儲存到區域性及全域變數。

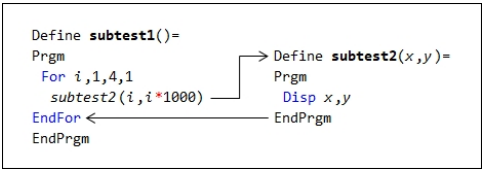
  附註：用來將值傳遞給函數的引數，會自動被視為區域性變數。如果您想儲存至任何其他的變數，則必須從函數內將它們宣告為 **Local**。
- 函數無法以副程式的方式呼叫程式，但可以呼叫其他使用者自行定義的函數。
- 您無法在函數內定義程式。
- 函數無法定義全域函數，但可以定義區域性函數。

從其他程式呼叫某個程式

一個程式可以副程式的方式呼叫其他程式。副程式可以是外部(個別的程式)或內部的(包括於主程式內)。當程式需要在數個不同位置重複一組相同指令時，副程式會很有用。

呼叫個別程式

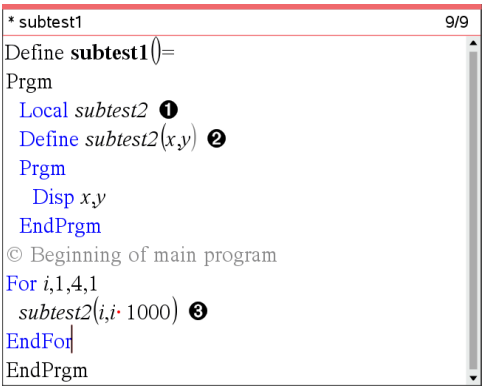
若要呼叫個別的程式，請使用您用來從輸入列執行程式的相同語法。



定義與呼叫內部副程式

若要定義內部副程式，請使用 **Define** 指令搭配 **Prgm...EndPrgm**。由於副程式必須在被呼叫前加以定義，最好能在主程式的開頭就定義副程式。

內部副程式被呼叫與執行的方式與個別程式相同。



❶ 將副程式宣告為區域性變數。

❷ 定義副程式。

❸ 呼叫副程式。

附註：使用 [程式編輯器] 的 **Var** 功能表來輸入 **Define** 與 **Prgm...EndPrgm** 指令。

使用副程式的注意事項

在副程式的結尾，會回到呼叫該副程式的程式繼續執行。若要在任何其他時候離開，請使用不包含引數的 **Return** 。

副程式無法存取在呼叫的程式中宣告的區域性變數。同樣地，呼叫的程式也無法存取在副程式中宣告的區域性變數。

Lbl 指令對它們所在的程式中是區域性的。因此，呼叫的程式中的 **Goto** 指令無法分枝到副程式中的標籤，反之亦然。

避免循環定義錯誤

在計算使用者自行定義的函數或執行程式時，您可以指定一個引數，以包括用來定義函數或建立程式的相同變數。不過，若要避免循環定義錯誤，您必須為用於計算函數或執行程式的變數指定值。例如：

```
x+1→x ❶
```

– 或 –

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

❶ 如果 x 或 i 沒有值，則會導致 **[循環定義]** 錯誤訊息。如果 x 或 i 已經有指定值，則不會發生該錯誤。

控制函數或程式的流程

當您執行程式或計算函數時，會按照順序執行程式中的每一行。不過，有些指令會更改程式流程。例如：

- 像是 **If...EndIf** 指令的控制結構便會使用條件檢定來決定要執行程式的哪一部分。
- 像是 **For...EndFor** 的迴圈指令會重複一組指令。

使用 If、Lbl 和 Goto 來控制程式流程

If 指令與數個 **If...EndIf** 結構可讓您依條件執行某個語句或語法區段，也就是以檢定的結果為基礎(比如 x>5) 。**Lbl**(標籤) 與 **Goto** 指令可讓您在函數或程式中，從一個位置分枝或跳到不同位置。

If 指令與數個 **If...EndIf** 結構位於 [程式編輯器] 的 **[控制]** 功能表。

當您插入像是 **If...Then...EndIf** 這樣的結構時，會在游標位置插入一個範本。游標會移到讓您可以輸入條件檢定的位置。

If 指令

若要在條件檢定的結果為「是」時執行單一指令，請使用一般形式：

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

❶ 僅在 x>5 時執行，否則會跳過。

❷ 一律顯示 x 的值。

在此範例中，您必須在執行 **If** 指令前將值儲存到 x。

If...Then...EndIf 結構

若要在條件檢定的結果為「是」時執行一組指令，請使用下列結構：

```
If x>5 Then
  Disp "x is greater than 5" ❶
  2·x→x ❶
EndIf
Disp x ❷
```

❶ 僅在 x>5 時執行。

❷ 顯示下列計算的值：
2x if x>5
x if x≤5

附註：EndIf 會標記當條件為「是」時所執行的 **Then** 區段結尾。

If...Then...Else...EndIf 結構

若要在條件檢定的結果為「是」時執行一組指令，且在結果為「非」時執行另一組指令，請使用下列結構：

```
If x>5 Then
  Disp "x is greater than 5" ❶
  2·x→x ❶
Else
  Disp "x is less than or equal to 5" ❷
  5·x→x ❷
EndIf
Disp x ❸
```

❶ 僅在 x>5 時執行。

❷ 僅在 x≤5 時執行。

❸ 顯示下列計算的值：
2x if x>5
5x if x≤5

If...Then...ElseIf...EndIf 結構

If 指令的更複雜形式可讓您檢定多個條件。假設您希望程式檢定一個代表四個選項之一由使用者提供的引數。

若要檢定每一個選項(If Choice=1、If Choice=2，依此類推)，請使用 **If...Then...ElseIf...EndIf** 結構。

Lbl 和 Goto 指令

您也可以使用 **Lbl**(標籤) 和 **Goto** 指令來控制流程。這些指令位於 [程式編輯器] 的 **[傳輸]** 功能表。

使用 **Lbl** 指令為函數或程式中的特定位置加上標籤(指定名稱) 。

Lbl *labelName*	要指定給此位置的名稱(使用相同的命名慣例作為變數名稱)

您可以接著在函數或程式中的任意點使用 **Goto** 指令，以分枝到對應於所指定標籤的位置。

Goto *labelName*	指定要分枝的目標 **Lbl** 指令

由於 **Goto** 指令是沒有條件的(一律會分枝到指定的標籤) ，所以經常會搭配 **If** 指令使用，讓您可以指定條件檢定。例如：

```
If x>5
  Goto GT5 ❶
Disp x

.....

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

❶ 如果 x>5，則直接分枝到標籤 GT5。

❷ 以此範例而言，程式必須包括可避免在 x≤5 時執行 **Lbl** GT5 的指令(比如 **Stop**) 。

使用迴圈來重複一組指令

若要接連重複相同的一組指令，請使用其中一個迴圈結構。有數種迴圈可供使用。每一種都提供不同的方式，依照條件檢定的結果來離開迴圈。

迴圈與迴圈相關的指令位於 [程式編輯器] 的 **[控制]** 和 **[傳輸]** 功能表。

當您插入其中一個迴圈結構，會將其範本插入游標的位置。您接著可以開始輸入將在迴圈內執行的指令。

For...EndFor 迴圈

For...EndFor 迴圈會使用計數器來控制迴圈重複的次數。下列為 **For** 指令的語法：

附註：如果增減的值為負數，則結束的值可少於開始的值。

For *variable*, *begin*, *end* [, *increment*]

❶ ❷ ❸ ❹

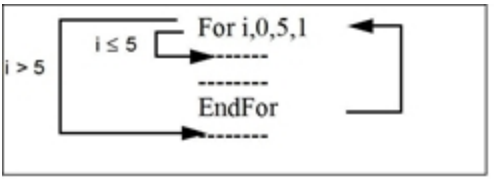
❶ *Variable* 用來作為計數器

❷ 第一次執行 **For** 時要使用的計數器值

❸ 當 *variable* 超過此值則離開迴圈

❹ 每次連續執行 **For** 時會新增到計數器(如果省略此非必要的值，則 *increment* 為 1。)

執行 **For** 時，會將 *variable* 值與 *end* 值進行比較。如果 *variable* 沒有超過 *end*，則會執行迴圈；否則，控制會跳到接在 **EndFor** 之後的指令。



附註：For 指令會自動增減計數器變數，讓函數或程式可以在特定的重複次數後離開迴圈。

在迴圈的結尾處(**EndFor**)，控制會跳回至 **For** 指令，其中系統會增減變數並與 *end* 進行比較。

例如：

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

❶ 顯示 0、1、2、3、4 和 5。

❷ 顯示 6。當 *variable* 增減至 6 時，便不會執行迴圈。

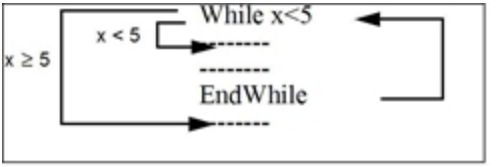
附註：如果您不需要在函數或程式停止後儲存計數器變數，可以將它宣告為區域性變數。

While...EndWhile 迴圈

While...EndWhile 迴圈會重複一個區段的指令，直到指定的條件為「是」。下列為 **While** 指令的語法：

While *condition*

執行 **While** 時，會計算 *condition*。如果 *condition* 為「是」，則會執行迴圈；否則，控制會跳到接在 **EndWhile** 之後的指令。



附註：While 指令不會自動更改條件。您必須包括允許函數或程式離開迴圈的指令。

在迴圈的結尾處(**EndWhile**)，控制會跳回至 **While** 指令，其中條件會經過重新計算。

要首次執行迴圈，條件一開始必須為「是」。

- 條件中所參照的任何變數必須設定在 **While** 指令之前。(您可以將值建置到函數或程式中，或可以提示使用者輸入值。)
- 迴圈必須包含在條件中更改值的指令，最終讓條件成為「非」。否則，條件會一律為「是」，而函數和程式則無法離開迴圈(稱為無限迴圈) 。

例如：

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

❶ 初次設定 x。

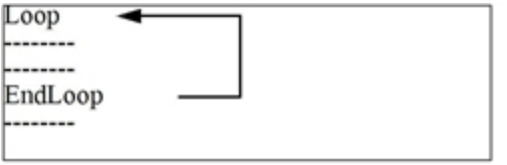
❷ 顯示 0、1、2、3 和 4。

❸ 增減 x。

❹ 顯示 5。當 x 增減至 5 時，便不會執行迴圈。

Loop...EndLoop 迴圈

Loop...EndLoop 會建立一個無限迴圈，無止盡地重複下去。**Loop** 指令沒有任何引數。



一般而言，您會在迴圈中插入讓程式離開迴圈的指令。下列為一般常用的指令：**If**、**Exit**、**Goto** 和 **Lbl** (標籤) 。例如：

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

❶ **If** 指令會檢查條件。

❷ *x* 增減到 6 時，則離開迴圈並跳到此處。

附註：Exit 指令從目前的迴圈離開。

在此範例中，**If** 指令可位於迴圈中的任何位置。

當 If 指令：	**迴圈為：**
在迴圈開頭	僅在條件為「是」時執行。
在迴圈結尾	執行至少一次並僅在條件為「是」時重複。

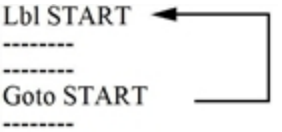
If 也可以使用 **Goto** 指令將程式控制傳輸給指定的 **Lbl(** 標籤 **)** 指令。

立即重複迴圈

Cycle 指令會立即將程式控制傳輸給迴圈的下一個迭代 (在完成目前的迭代前) 。此指令與 **For...EndFor°BWhile...EndWhile** 和 **Loop...EndLoop** 搭配使用。

Lbl 和 Goto 迴圈

雖然嚴格來說 **Lbl(** 標籤 **)** 和 **Goto** 指令不算迴圈指令，但仍可以用它們來建立無限迴圈。例如：



與使用 **Loop...EndLoop** 時相同，迴圈應包含可讓函數或程式離開迴圈的指令。

更改模式設定

函數和程式可使用 **setMode()** 函數來暫時設定特定的計算或計算結果模式。[程式編輯器] 的 **[模式]** 功能表可讓您輕鬆輸入正確的語法，而不必記住數字代號。

附註：在函數或程式定義中對模式進行的更改不會影響到函數或程式的外部設定

設定模式

1. 將游標放在您想插入 **setMode** 函數的位置。
2. 從 **[模式]** 功能表選取要更改的模式，然後選取新設定。

   系統會在游標位置插入正確的語法。例如：

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

除錯程式與處理錯誤

在您編寫完函數或程式之後，可以使用數種技術來找尋並更正錯誤。您也可以建置一個錯誤處理指令到函數或程式本身。

如果您的函數或程式會讓使用者從數個選項中進行選取，請務必先執行並測試每一個選項。

除錯技術

執行階段錯誤訊息會找出語法錯誤，但不會找出在程式邏輯上犯的錯誤。下列為有用的技術。

- 暫時插入 **Disp** 指令來顯示重要變數的值。
- 若要確認以正確的次數執行迴圈，請使用 **Disp** 來顯示計數器變數或條件檢定中的值。
- 若要確認有執行副程式，請使用 **Disp**，在副程式的開頭與結尾顯示像是「正在進入副程式」和「正在離開副程式」的訊息。
- 手動停止程式或函數：
  - **Windows®：**按住 **F12** 鍵並重複按 **Enter** 鍵。
  - **Macintosh®：**按住 **F5** 鍵並重複按 **Enter** 鍵。
  - **計算機：**按住 [on] 鍵並重複按 [enter]。

錯誤處理指令

指令	說明
Try...EndTry	定義一個區段，讓函數或程式執行指令，並在必要時從該指令產生的錯誤中恢復。
ClrErr	清除錯誤狀態，並將系統變數 *errCode* 設為零。若需使用 *errCode* 的範例，請參考 *《參考手冊》*中的 ***Try* 指令**。
PassErr	將錯誤傳遞到 **Try...EndTry** 區段的下一級。

一般資訊

線上說明

education.ti.com/eguide

選擇您的國家/地區以取得更多產品資訊。

連絡 TI 技術支援部門

education.ti.com/ti-cares

選擇您的國家/地區以取得技術和其他支援資源。

服務與保固資訊

education.ti.com/warranty

選擇您的國家/地區，即可瞭解保固期間與條款或產品服務的相關資訊。

這保證不會影響您的法定權利。

Texas Instruments Incorporated

12500 TI Blvd.

Dallas, TX 75243