

fChart 6 使用手冊

<https://fchart.github.io>

- ◆ 安裝、設定與啟動 fChart
- ◆ 建立第一個 fChart 流程圖
- ◆ fChart 流程圖的直線、L 型和 U 型連接線
- ◆ 開啟與執行 fChart 流程圖
- ◆ fChart 分類 150 個流程圖專案
- ◆ 編輯 fChart 流程圖專案
- ◆ fChart 流程圖符號的對話方塊
- ◆ fChart 程式碼編輯器與 Blockly 積木程式

安裝、設定與啟動 fChart

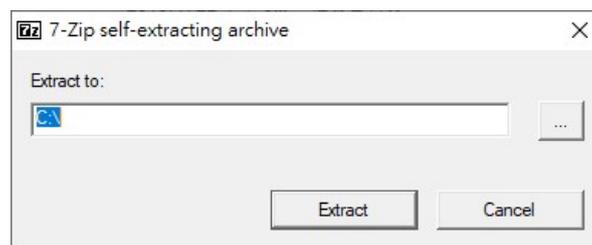
fChart 6.0 版不只支援多國語言的使用介面，更重新改寫流程圖的連接線功能，在新版本共支援五種連接線：直線、先水平再垂直 L 型線、先垂直再水平 L 型線、順時鐘和反時鐘的 U 型線。

請注意！雖然新版相容舊版專案，但新版建立的流程圖專案，在舊版 fChart 開啟時不會錯誤，但是因為只支援直線，所有 L 或 U 型線都只能顯示直線，其顯示的流程圖可能會有很大的差異。

fChart 程式語言教學工具是一套程式設計的教學工具，也是一套輕量級的整合開發環境（內建 Blockly 積木程式），主要分成兩大工具：流程圖直譯器和程式碼編輯器。

安裝 fChart

fChart 新版本已經改為 7-Zip 格式的自解壓縮檔，以標準版下載的 fChart6.exe 為例，只需執行下載的執行檔，即可將相關程式檔案解壓縮至指定目錄（工具並不需安裝，解壓縮後即可使用），如下圖所示：



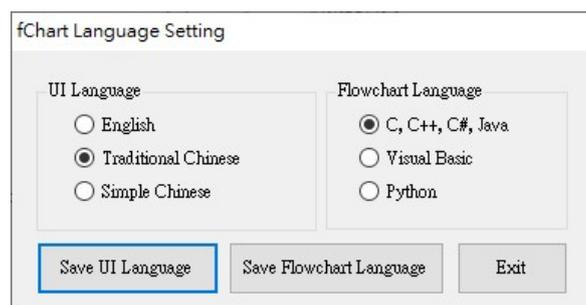
上述對話方塊是 7-Zip 自解壓縮對話方塊，因為壓縮檔已經包含「fChart6」目錄，在欄位只需輸入硬碟【C:\】或【D:\】即可，按【Extract】鈕即可解壓縮至此硬碟下的「fChart6」目錄，以 C:\為例，

標準版是「C:\fChart6」目錄；Python 版預設是「C:\fChartPython6」；Node 版是「C:\fChartNode6」，在目錄下的主要檔案說明，如下所示：

- **RunfChart.exe**：使用系統管理員身份啟動 fChart 流程圖直譯器。
- **FlowProgramming_Edit.exe**：fChart 流程圖直譯器的執行檔，因為權限問題，直接執行檔案，需在檔名上執行【右】鍵快顯功能表的【以系統管理員身份執行】命令，使用系統管理員身份來執行。
- **fChartSetting.exe**：fChart 使用介面和流程圖語言的設定程式。
- **fChartCodeEditor.exe**：fChart 程式碼編輯器的執行檔。

設定 fChart 介面語言和流程圖的程式語言

fChart 支援多國語言使用介面，預設介面是繁體中文，流程圖是 C、C++和 Java 語言。請執行安裝目錄下的【fChartSetting.exe】來設定使用的介面語言，支援繁體中文、簡體中文和英文，如下圖所示：



在上述對話方塊的左邊是使用介面語言，在選擇後，請按【Save UI Language】鈕儲存介面語言設定；右邊是流程圖使用的語言（影響運

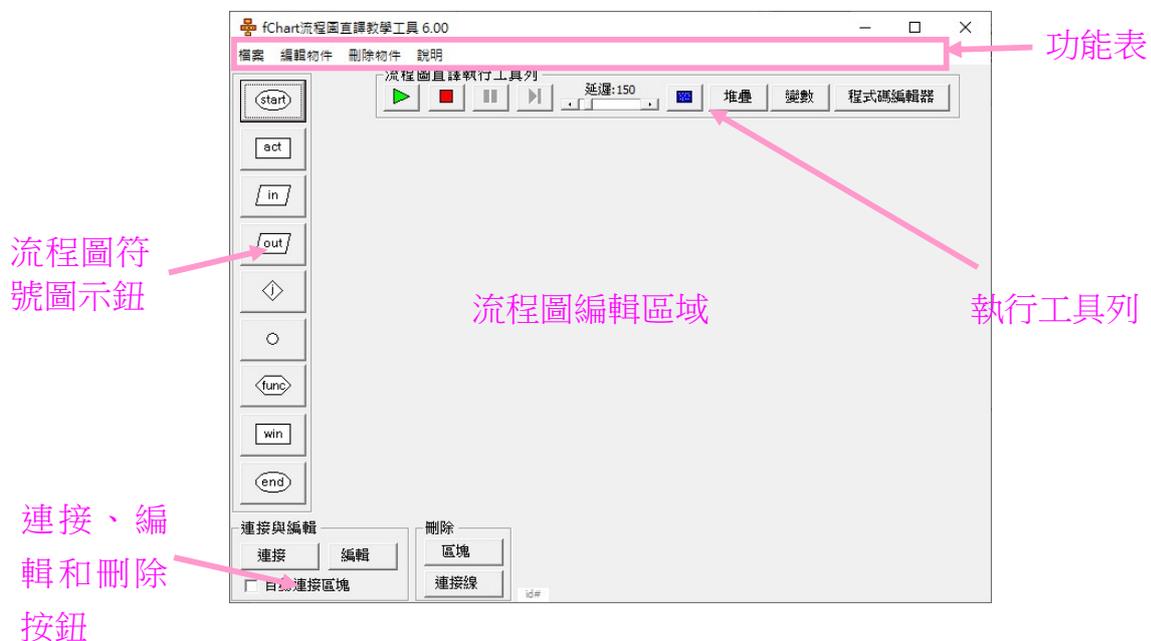
算子)，在選擇程式語言後，請按【Save Flowchart Language】鈕儲存設定，在完成設定後，請按【Exit】鈕結束設定程式。

啟動 fChart 流程圖直譯器

在解壓縮和設定 fChart 後，我們可以馬上在 Windows 作業系統執行 fChart 流程圖直譯器（從此工具可以啟動 fChart 程式碼編輯器），其步驟如下所示：

Step 1：請開啟 fChart 所在的「\fChart6」目錄，執行【RunfChart.exe】後，按【是】鈕啟動 fChart 流程圖直譯器。

Step 2：在成功啟動 fChart 流程圖直譯器後，可以看到流程圖編輯的使用介面。



上述圖例是 fChart 流程圖直譯器的使用介面，在上方是功能表，功能表下方是執行工具列，可以執行我們繪出的流程圖，左邊是建立流程圖符號圖示的按鈕工具列，在下方是連接、編輯和刪除圖示符號的按鈕，位在中間區域就是編輯建立流程圖的編輯區域。

結束 fChart 流程圖直譯器

請執行「檔案>結束」命令結束 fChart 流程圖直譯器，或是按視窗右上角【X】鈕關閉流程圖直譯器。

建立第一個 fChart 流程圖

在啟動 fChart 流程圖直譯器後，我們可以馬上開始繪製第一個流程圖，fChart 流程圖直譯器提供十分容易的方式來繪製流程圖。

流程圖的基本繪製原則

為了繪製良好的流程圖，一些繪製流程圖的基本原則，如下所示：

- 流程圖請使用標準的圖示符號，以方便閱讀、溝通和小組討論。
- 在每一個流程圖符號的說明文字需力求簡潔、扼要和明確可行。
- 流程圖只能有一個起點，和至少一個終點。
- 流程圖的繪製方向是從上而下；從左至右。
- 決策符號有兩條出去連接線的流程符號；終止符號不允許有出去連接線的流程符號。
- 流程圖連接線的流程符號應避免交叉或太長，如果有多條進入的連接線，可以使用連接符號來進行連接。

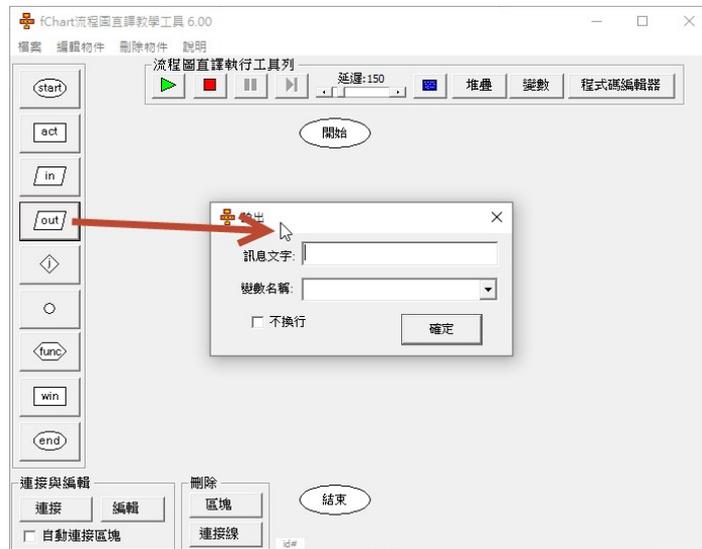
建立第一個 fChart 流程圖

我們準備建立第 1 個 fChart 流程圖來顯示一段文字內容，即傳統程式語言最常見的 Hello World 程式，其步驟如下所示：

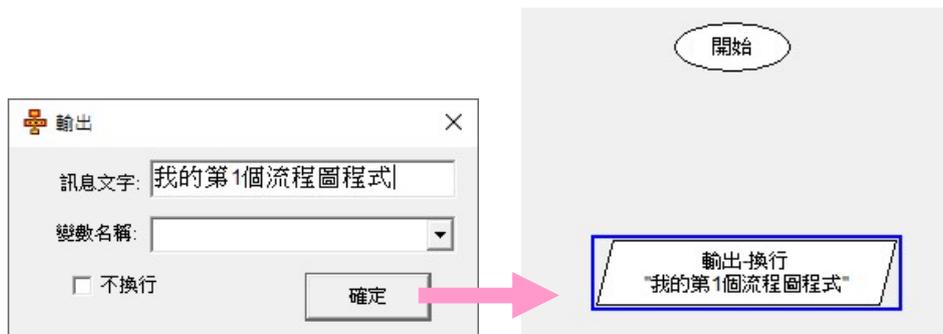
Step 1：請啟動 fChart 流程圖直譯器，執行「檔案>新增流程圖專案」命令，可以看到新增的流程圖專案，預設新增開始和結束 2 個符號。



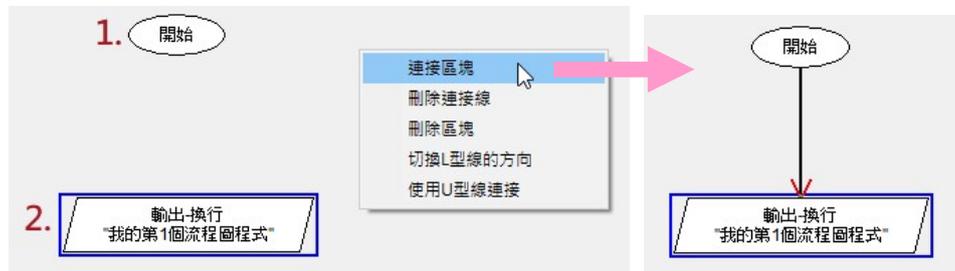
Step 2：在左邊垂直工具列，選第 4 個 out 輸出符號後，拖拉至插入位置，點選一下，開啟「輸出」對話方塊。



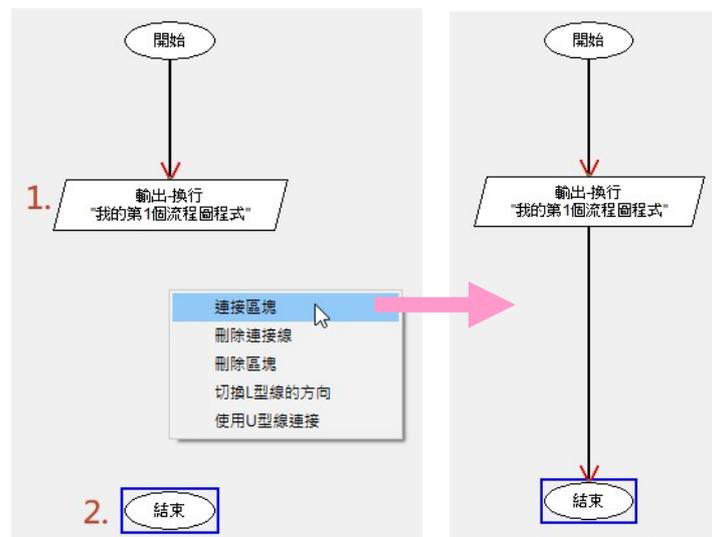
Step 3：在【訊息文字】欄輸入欲輸出的文字內容【我的第 1 個流程圖程式】，如果有輸出變數值，請在下方【變數名稱】欄位輸入或選擇變數名稱，按【確定】鈕，可以看到新增的輸出符號。



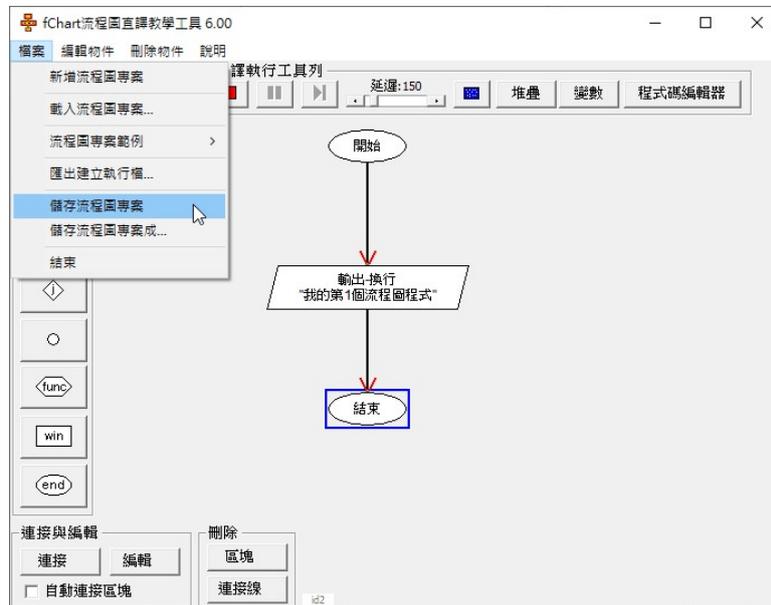
Step 4：接著連接流程圖符號，請先點選「開始」符號，然後是「輸出」符號，在沒有符號的區域，執行滑鼠【右】鍵快顯功能表的【連接區塊】命令，可以新增開始至輸出符號間的連接線，紅色箭頭是執行方向。



Step 5：然後點選輸出符號，再選結束符號，在沒有符號區域，執行滑鼠【右】鍵快顯功能表的【連接區塊】命令，新增輸出至結束符號間的連接線，紅色箭頭是執行方向。



Step 6：在拖拉調整流程圖符號的位置後，即可完成 fChart 流程圖的繪製。

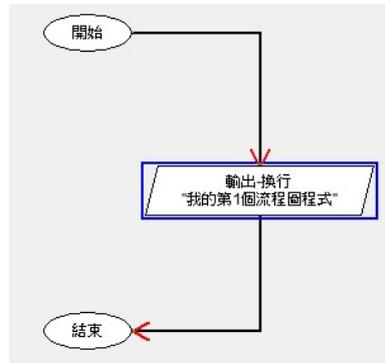


Step 7：請執行「檔案>儲存流程圖專案」命令儲存流程圖專案，可以看到「另存新檔」對話方塊，請切換路徑和輸入檔案名稱【FirstProgram.fpp】後，按【存檔】鈕儲存專案，副檔名是.fpp。

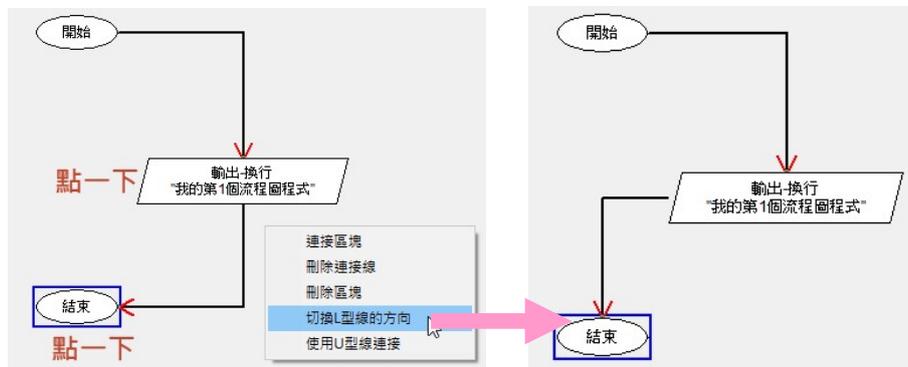
fChart 流程圖的直線、L 型和 U 型連接線

fChart 預設能夠自動依據符號位置來調整是否使用 L 型連接線（自動 L 型線），如果符號太接近，空間不足是使用直線；空間足夠使用 L 型線，當在起點座標已經有其他連接線（fChart 預設不允許出去和進入連接線是重疊線；多條進入線是可以是重疊線），就自動切換成相反方向的 L 型線，如下所示：

- **自動 L 型線**：當 2 個符號位置如果是終點符號是位在起點符號的右下方；L 型線預設是先水平再垂直 90 度；如果是位在右上方；左下方；左上方，預設是先垂直再水平 90 度的 L 型線，如下圖所示：

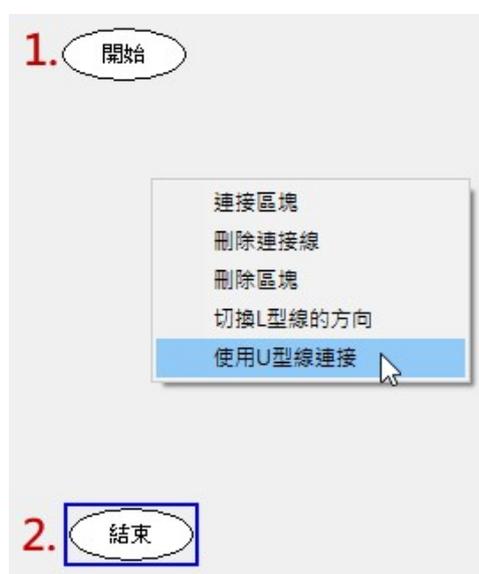


- 切換 L 型線的方向：在連接線的 2 個符號各點選一下（順序沒有關係）後，執行「編輯物件>切換 L 型線的方向」命令，或【右】鍵快顯功能表的【切換 L 型線的方向】命令，可以切換成先水平再垂直 L 型線或先垂直再水平的 L 型連接線，如下圖所示：

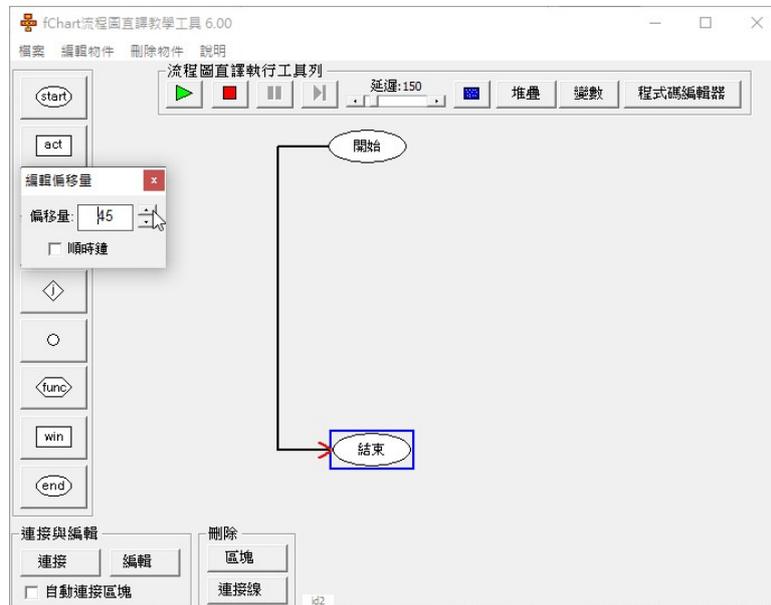


請注意！L 型連接線會自動判斷符號的起點是否已經有其他進入的連接線（有重疊連接線），如果有，就會自動切換至反方向的 L 型線，此時，因為反方向 L 型線的起點有重疊連接線，fChart 預設不允許出去和進入連接線是重疊線，所以不能手動切換 L 型線的方向，如需切換方向，請先調整其他符號的連接線，移開造成相同點的其他連接線後，即可自動更新或手動切換 L 型連接線的方向。

- **U 型線**：如同其他連接線，在欲連接的 2 個符號各點選一下（順序是先開始；然後結束）後，執行「編輯物件>使用 U 型線連接」命令，或【右】鍵快顯功能表的【使用 U 型線連接】命令，預設建立使用偏移量 40 建立反時鐘連接的 U 型連接線，同時顯示對話方塊來調整偏移量，最小值是 20，如下圖所示：



U 型線預設是使用反時鐘方向來連接 2 個符號，同時顯示「編輯偏移量」對話方塊，我們可以調整 U 型線的偏移量，如果勾選下方【順時鐘】的核取方塊，可以切換成順時鐘方向的 U 型線連接，如下圖所示：



在建立 U 型線後，如需編輯 U 型線，我們可以在連接線的 2 個符號各點選一下（順序沒有關係）後，執行「編輯物件>編輯 U 型線的偏移量」命令來更改偏移量和 U 型線的連接方向。

- **設定使用直線**：如果一定需要使用直線，不使用 L 型或 U 型線，請在欲設定的 2 個符號各點選一下（順序沒有關係）後，執行「編輯物件>使用直線的連接線」命令，就可以改用直線的連接線來連接 2 個符號，
- **將 U 型線或直線重設為自動 L 型線**：如果連接線已經指定成 U 型或直線，欲重設成預設的自動 L 型線，請在欲設定連接的 2 個符號各點選一下（順序沒有關係）後，執行「編輯物件>切換 L 型線的方向」命令，可以再切換成預設的自動 L 型連接線。

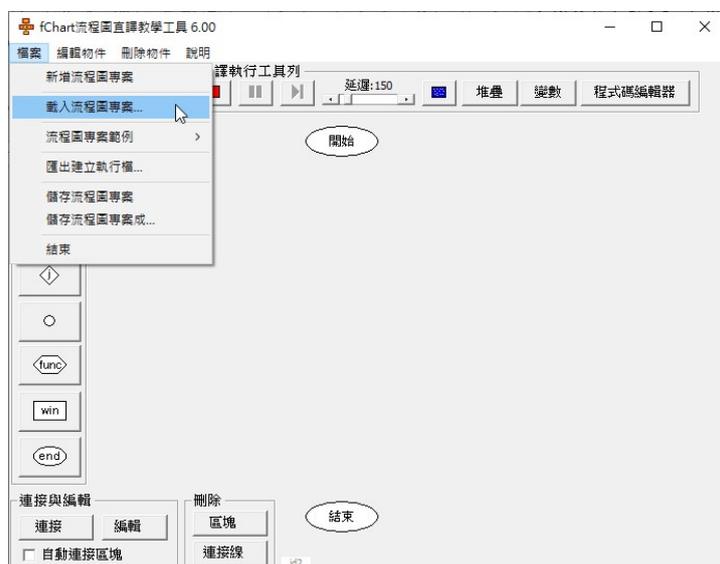
開啟與執行 fChart 流程圖

fChart 流程圖直譯器不只可以編輯繪製流程圖，還可以使用動畫來完整顯示流程圖的執行過程和結果，輕鬆幫助我們驗證演算法是否可行，和訓練初學程式設計者的程式邏輯。

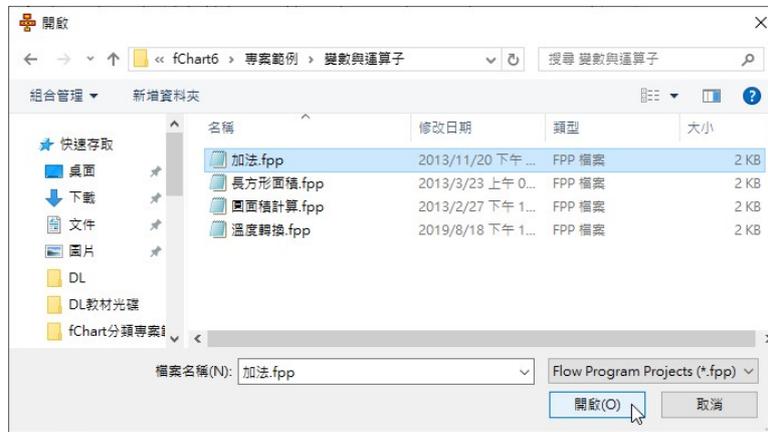
開啟 fChart 專案執行流程圖

對於已經存在的 fChart 流程圖專案，我們可以載入流程圖專案來執行，其步驟如下所示：

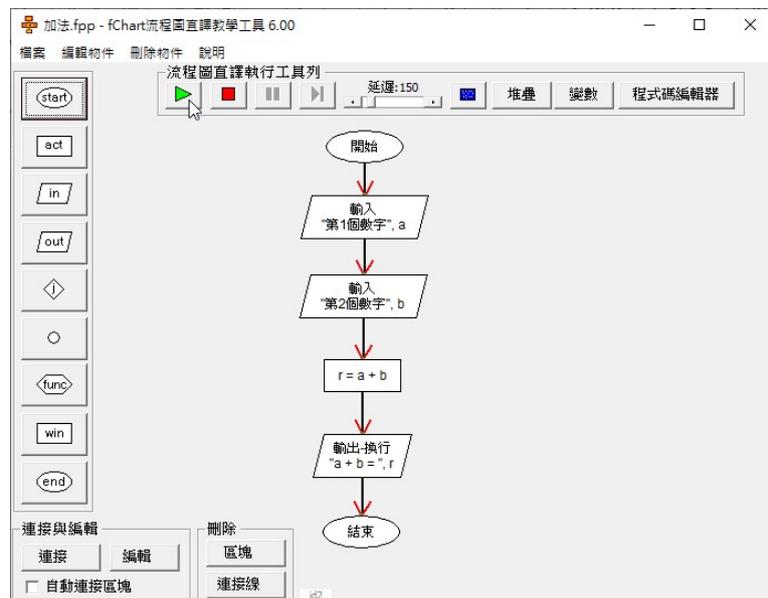
Step 1：請啟動 fChart 流程圖直譯器，可以進入流程圖編輯的使用介面。



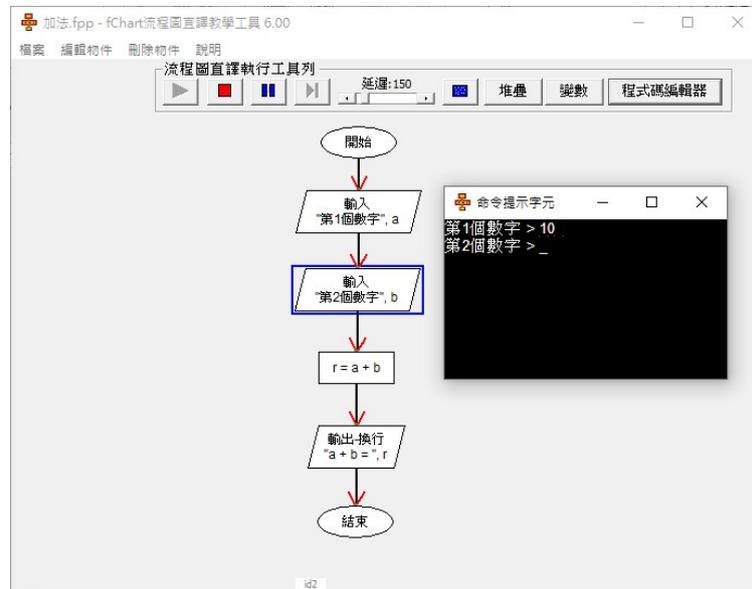
Step 2：請執行「檔案>載入流程圖專案」命令，可以看到「開啟」對話方塊。



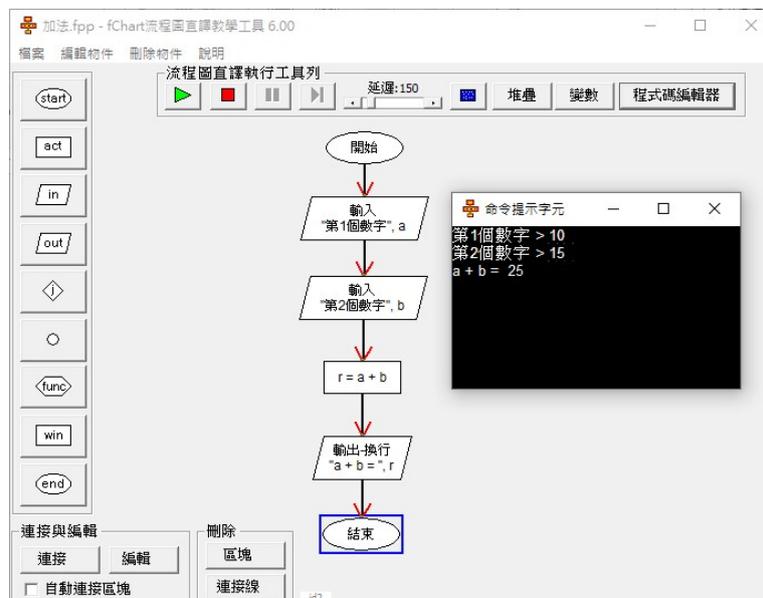
Step 3：切換至「\fChart6\專案範例\變數與運算子」路徑，選【加法.fpp】，按【開啟】鈕載入流程圖。



Step 4：按上方執行工具列的第 1 個【執行】鈕，可以看到動畫移動藍色框來執行流程圖，因為執行到輸入符號，所以顯示「命令提示字元」視窗，和輸入第 1 個數字的提示文字。



Step 5：請輸入 10，按 **Enter** 鍵，可以看到輸入第 2 個數字，請輸入 15，按 **Enter** 鍵，可以看到流程圖繼續執行，和顯示執行結果 25。



Step 6：我們可以再次執行流程圖，並且輸入不同值，就可以看到不同的執行結果。

流程圖直譯執行工具列

fChart 流程圖直譯器是使用上方執行工具列按鈕來控制流程圖的執行，我們可以調整執行速度和顯示相關輔助資訊視窗，如下圖所示：



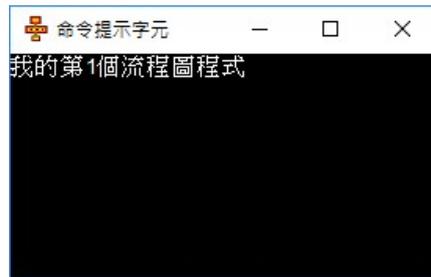
上述執行工具列按鈕從左至右的說明，如下所示：

- **執行**：按下按鈕開始執行流程圖，這是使用延遲時間定義的間隔時間來一步一步自動執行流程圖，如果流程圖需要輸入資料，就會開啟「命令提示字元」視窗讓使用者輸入資料（在輸入資料後，請按 **Enter** 鍵）例如：【加法.fpp】，如下圖所示：

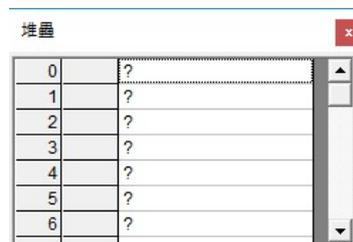
```
命令提示字元
第1個數字 > 34
第2個數字 > 56
a + b = 90
```

- **停止**：按此按鈕停止流程圖的執行。

- **暫停**：當執行流程圖時，按此按鈕暫停流程圖的執行。
- **逐步執行**：當延遲時間捲動軸調整至最大時，就是切換至逐步執行模式，此時按【執行】鈕執行流程圖，就是一次一步來逐步執行流程圖，請重複按此按鈕來執行流程圖的下一步。
- **調整延遲時間**：使用捲動軸調整執行每一步驟的延遲時間，如果調整至最大，就是切換成逐步執行模式。
- **顯示命令提示字元視窗**：按下此按鈕可以顯示「命令提示字元」視窗的執行結果，例如：FirstProgram.fpp，如下圖所示：



- **顯示堆疊視窗**：在「堆疊」視窗顯示函數呼叫保留的區域變數值，如下圖所示：



- **顯示變數視窗：**在「變數」視窗顯示執行過程的每一個變數值，包含目前和之前上一步的變數值，例如：**【加法.fpp】**，如下圖所示：

	RETURN	PARAM	a	b	r	RET-OS
目前變數值:		PARAM	43	56	99	
之前變數值:		PAR-OS				

- **程式碼編輯器：**啟動 fChart 程式碼編輯器。

fChart 分類 150 個流程圖專案

fChart 在 GitHub 檔案庫提供有 150 個分類範例，請啟動瀏覽器進入 6.0 版 fChart 分類 150 個流程圖專案的 GitHub 檔案庫，如下所示：

- <https://github.com/fchart/fChartExamples2>

Why GitHub? Enterprise Explore Marketplace Pricing Search Sign in Sign up

fchart / fChartExamples2 Watch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Security Insights

Join GitHub today
Dismiss
GitHub is home to over 40 million developers working together to host and review code, manage projects, and build software together.
Sign up

fChart 6.0以上版本的分類範例

32 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Find File Clone or download

fchart Update README.md Latest commit 9c2437a 2 days ago

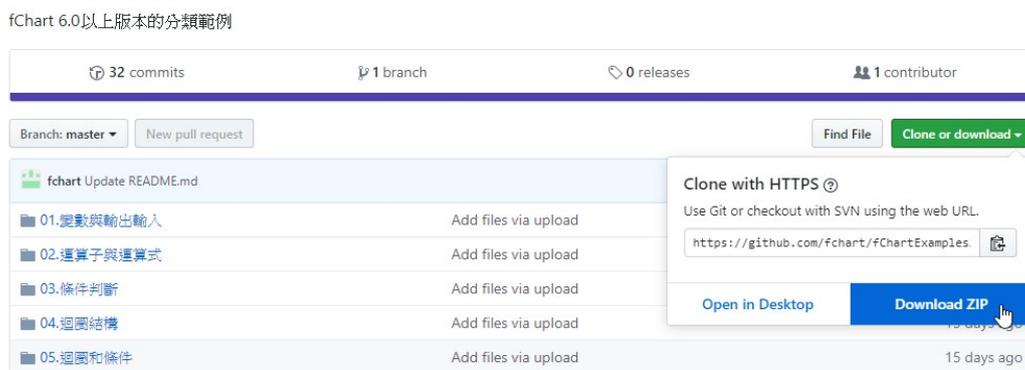
- 01.變數與輸出輸入 Add files via upload 15 days ago
- 02.運算子與運算式 Add files via upload 15 days ago
- 03.條件判斷 Add files via upload 15 days ago

上述 GitHub 檔案庫提供分成十類約 150 個 fChart 流程圖專案，可以作為程式設計的學習單、作業或考試題目，因為提供的是可執行的流程圖，即程式的演算法，所以適用各種程式語言：C/C++、C#、Python、Java、JavaScript、VB 和 PHP 等，如下圖所示：

- 01.變數與輸出輸入
- 02.運算子與運算式
- 03.條件判斷
- 04.迴圈
- 05.迴圈和條件
- 06.巢狀迴圈
- 07.函數
- 08.陣列
- 09.演算法
- 10.遞迴函數

下載整個 GitHub 檔案庫

請在檔案庫首頁，點選【Clone or download】鈕，再選【Download ZIP】，即可下載整個 GitHub 檔案庫，如下圖所示：

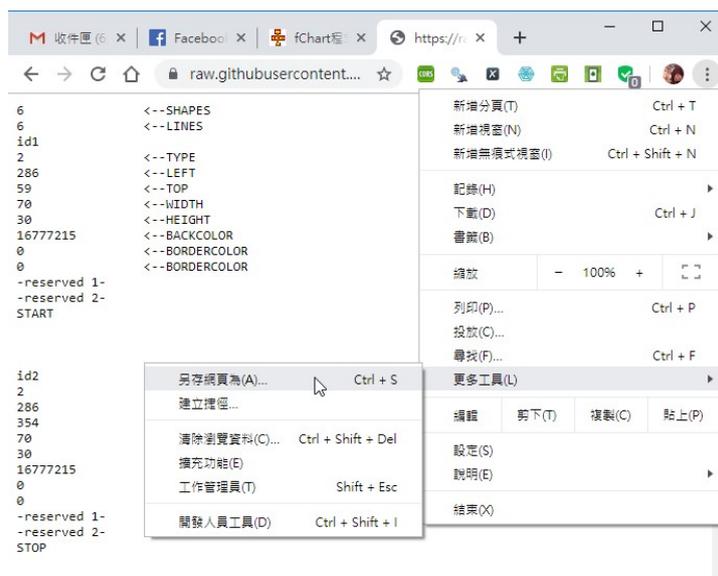


下載指定的 fChart 流程圖專案

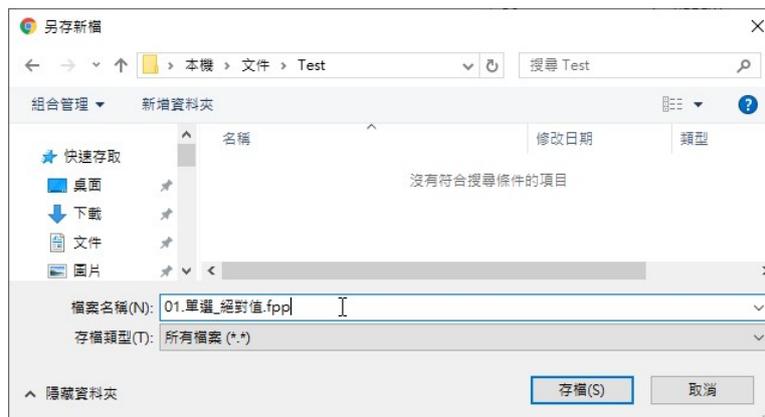
如果是下載特定的 fChart 流程圖專案，例如：【01.單選_絕對值.fpp】，請點選目錄切換至此.fpp 檔案，點選【Raw】鈕，如下圖所示：



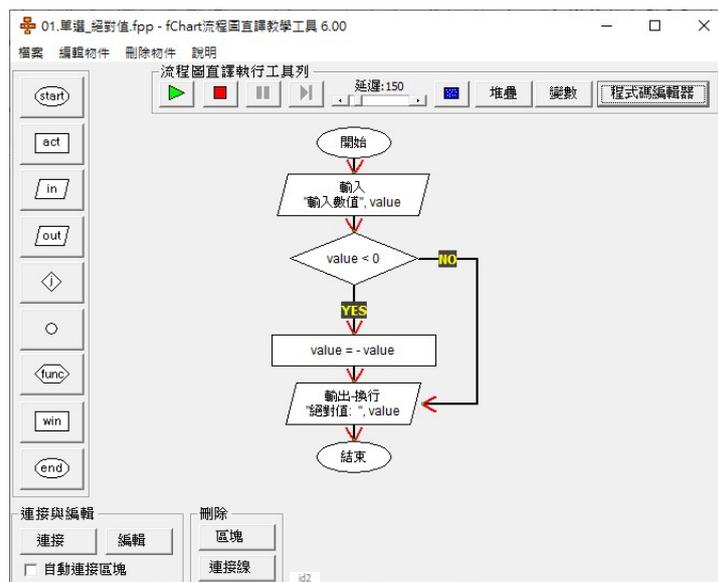
在顯示.fpp 的檔案內容（.fpp 檔案格式是文字檔案），請執行 Chrome 瀏覽器的「更多工具>另存網頁為」命令，如下圖所示：



在「另存新檔」對話方塊切換至儲存路徑，【存檔類型】欄選【所有檔案 (*.*)】，在【檔案名稱】欄刪除最後的【.txt】，按【存檔】鈕儲存成副檔名.fpp 的 fChart 流程圖專案檔，如下圖所示：



在成功下載 GitHub 檔案庫的 fChart 流程圖專案檔.fpp 後，我們就可以在 fChart 開啟此專案檔，如下圖所示：

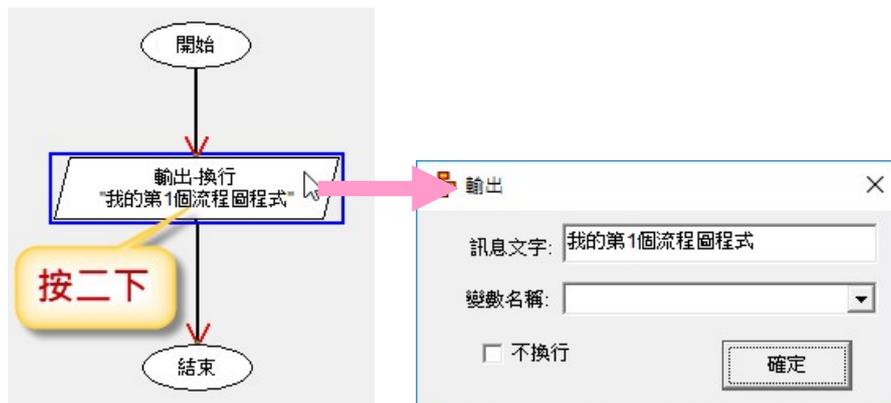


編輯 fChart 流程圖專案

我們只需開啟已經建立的 fChart 流程圖專案，即可重新編輯專案的流程圖。

編輯流程圖符號

在流程圖編輯區域建立的流程圖符號，只需按二下符號圖示，就可以開啟符號的編輯對話方塊，重新編輯流程圖符號，如下圖所示：



連接兩個流程圖符號

在 fChart 新增連接 2 個流程圖符號之間的連接線，請在欲連接的 2 個符號各點選一下（順序是先點選開始符號，然後是結束符號）後，我們共有二種方式來建立 2 個符號之間的連接線，紅色箭頭是執行方向，如下所示：

- 請按左下方「連接與編輯」框的【連接】鈕來新增連接線，如下圖所示：



- 在沒有符號區域，執行滑鼠【右】鍵快顯功能表的【連接區塊】命令來新增連接線（執行【使用 U 型線連接】命令是使用 U 型線）。

如果在左下方「連接與編輯」框勾選【自動連接區塊】，在新增符號圖示後，就會自動新增符號圖示之間的連接線，如下圖所示：



刪除符號間的連接線

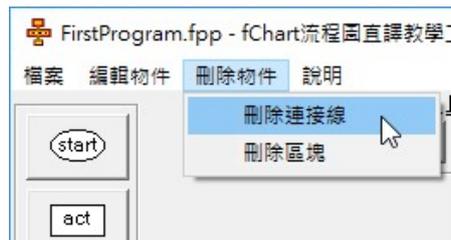
刪除連接線請分別點選一下連接線兩端的流程圖符號（順序沒有關係），我們共有三種方式來刪除連接線，如下所示：

- 按左下方「刪除」框的【連接線】鈕刪除之間的連接線。



- 在沒有符號區域，執行滑鼠【右】鍵快顯功能表的【刪除連接線】命令來刪除連接線。

- 執行「刪除物件>刪除連接線」命令刪除連接線，如下圖所示：



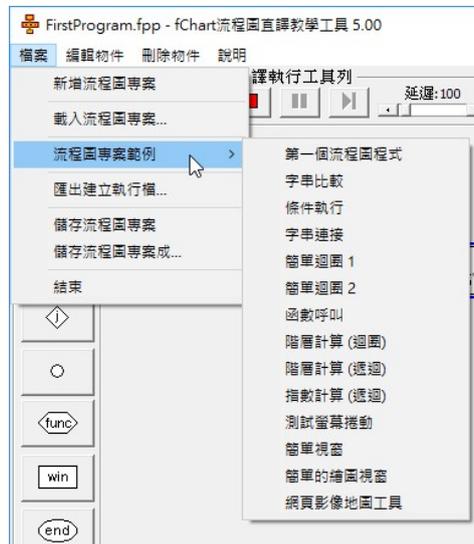
刪除流程图符號

當流程图符號沒有任何連接線時，我們才可以刪除流程图符號，請點選一下欲刪除符號後，我們共有三種方式刪除流程图符號，如下所示：

- 按左下方「刪除」框的【區塊】鈕來刪除流程图符號。
- 在沒有符號區域，執行滑鼠【右】鍵快顯功能表的【刪除區塊】命令。
- 執行「刪除物件>刪除區塊」命令。

流程图專案範例

在「檔案>流程图專案範例」功能表的子選單提供多個內建流程图專案的範例，如下圖所示：



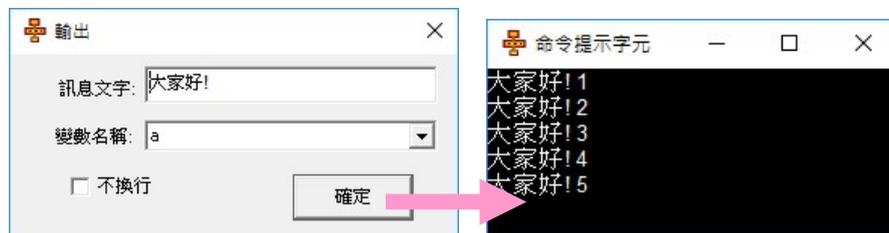
請直接執行選項命令，就可以馬上載入範例的 fChart 流程圖專案，在 fChart 安裝目錄下的「專案範例」子目錄提供更多流程圖範例專案。

fChart 流程圖符號的對話方塊

在 fChart 流程圖直譯器左邊工具列點選欲新增的流程圖符號後，移動符號圖示至編輯區域的欲插入位置，點選一下，可以開啟編輯符號的對話方塊來編輯符號內容，各種符號對話方塊的說明，如下所示：

輸出符號

輸出符號是用來顯示程式的執行結果，請在「輸出」對話方塊的【訊息文字】欄輸入欲輸出的文字內容，在下方【變數名稱】欄位可以輸入或選擇輸出的變數值，例如：運算結果的變數 a，如下圖所示：



上述圖例沒有勾選【不換行】，如果勾選，可以看到輸出不會換行，如下圖所示：

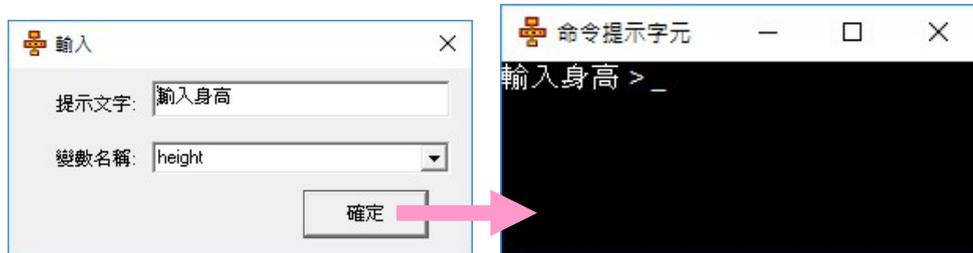


在輸出符號的【訊息文字】欄位的輸出字串可以使用{0}符號（位在中間是數字 0）標示下方【變數名稱】欄變數值顯示的位置，換句話說，變數值不只可以顯示在最後，還可以顯示在字串之中，例如：

輸出{0}元
第一節: {0} 分

輸入符號

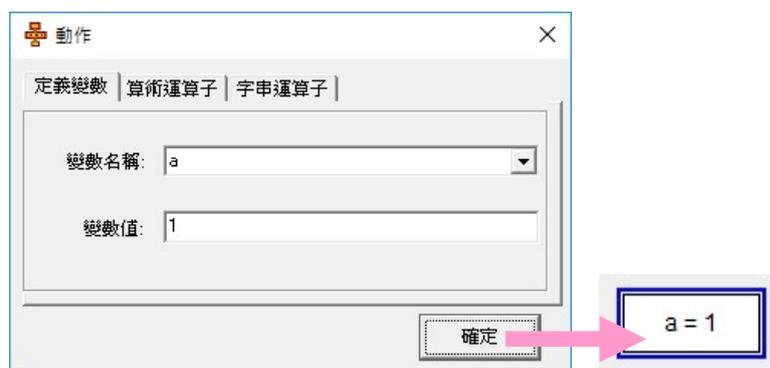
輸入符號可以讓使用者輸入資料，我們是在「輸入」對話方塊的【提示文字】欄輸入提示說明文字，下方【變數名稱】欄位輸入或選擇輸入的變數名稱，例如：讓使用者輸入的資料儲存至下方變數 `height`，如下圖所示：



動作符號

動作符號可以定義變數值、陣列變數，或建立擁有 2 個運算元的算術和字串運算式，如下所示：

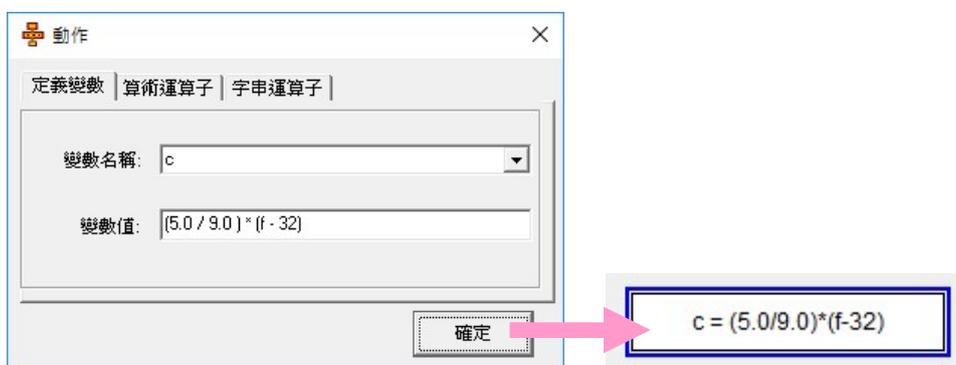
- 定義變數：在【定義變數】標籤新增變數和指定初值，我們可以在【變數名稱】欄輸入新增的變數名稱（或選擇目前已經新增過的變數），【變數值】欄輸入變數值（也可以是其他變數名稱 b ，即將其他變數值指定給變數，例如： $a = b$ ），如下圖所示：



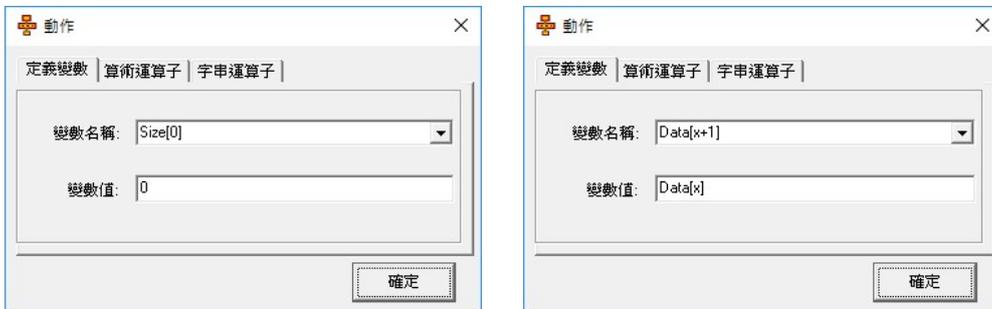
- 變數值是運算式：除了指定變數的值是常數值或其他變數值外，我們還可以在【變數值】欄位輸入完整的算術運算式，即指定變數是此運算式的運算結果，支援「+」、「-」、「*」、「/」、「\」（整數除法）、「^」（指數）、「%」（餘數）運算子和「（）」括

號，運算元可以使用整數、浮點數或數學函數。目前支援的數學函數，如下表所示：

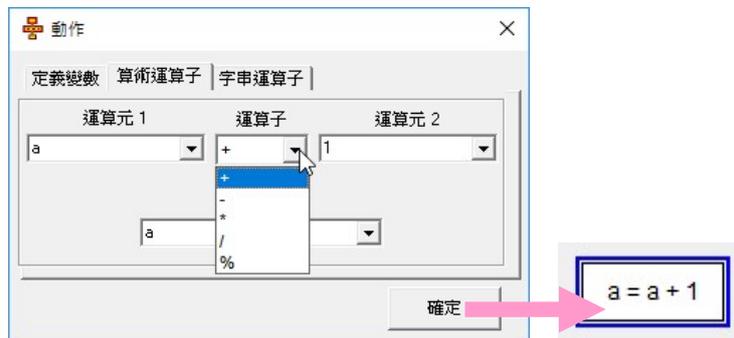
數學函數	說明
abs(exp)	絕對值函數
int(exp)、fix(exp)	取得整數值
sin(rad)、cos(rad)、tan(rad)、atn(rad)	三角函數，參數是徑度 $\text{deg} * 3.1415926 / 180$
sqr(exp)	開平方根
factorial(exp)	階乘函數



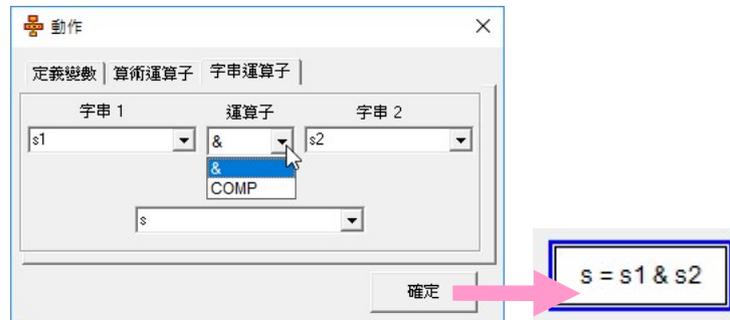
- 建立陣列變數：在【定義變數】標籤可以建立陣列變數和指定初值，fChart 並不用宣告陣列，可以馬上使用陣列變數（因為 fChart 將陣列變數視為是一個擁有索引的變數），請使用方框「[]」建立，在方框中也可以使用運算式，如下圖所示：



- 建立二元運算式：在【算術運算子】標籤可以使用選擇方式來建立二元的算術運算式，在中間可以選擇使用的運算子：「+」（加）、「-」（減）、「*」（乘）、「/」（除）、和「%」（餘數），如下圖所示：

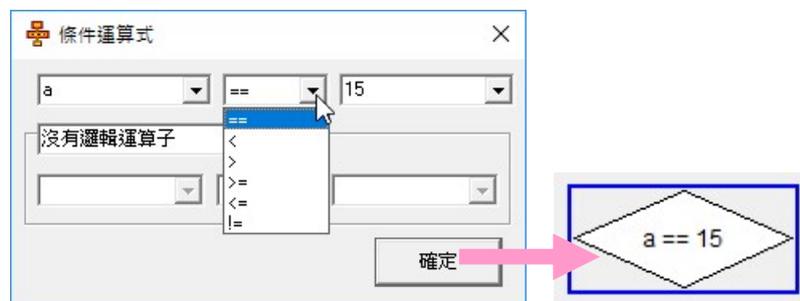


- 建立字串運算式：在【字串運算子】標籤可以建立字串運算式，可以使用「&」（字串連接）和「COMP」（字串比較）運算子，如下圖所示：

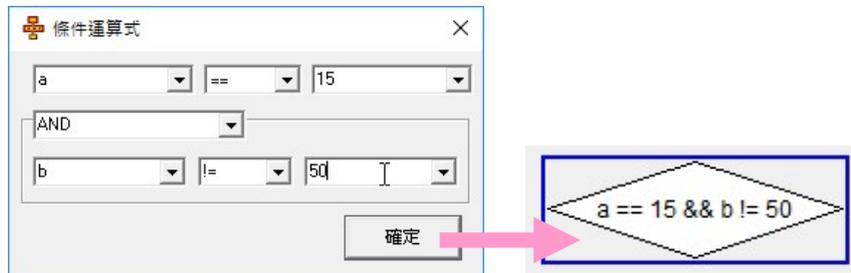


決策符號

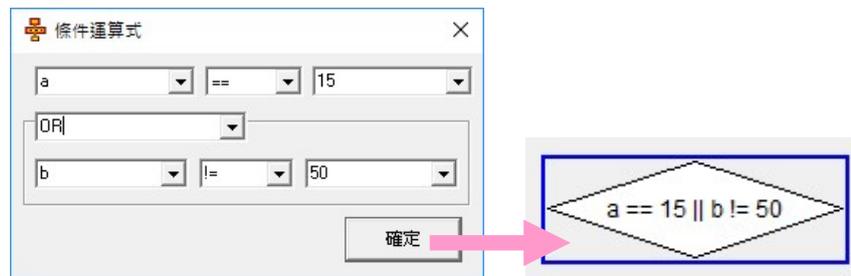
決策符號是用來建立條件運算式，可以建立 2 個運算元的比較運算式，在中間可以選擇條件運算子：「==」（等於）、「<」（小於）、「>」（大於）、「>=」（大於等於）、「<=」（小於等於）和「!=」（不等於），如下圖所示：



fChart 流程圖直譯器支援邏輯運算子 AND 和 OR，可以建立 2 個比較運算式作為運算元的邏輯運算式。首先是 AND 邏輯運算子，如下圖所示：

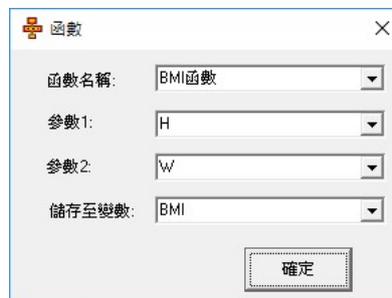


然後是 OR 邏輯運算子，如下圖所示：

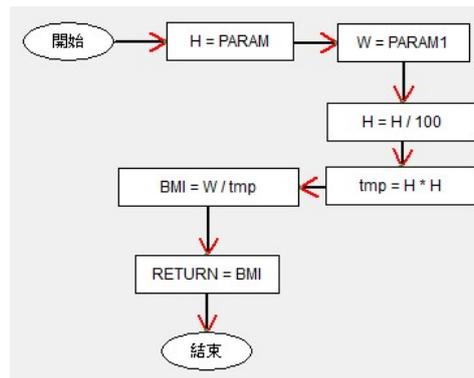


函數符號

fChart 流程圖的.fpp 檔可以是主程式；也可以是一個同名的函數，我們可以傳遞預設名為 PARAM 和 PARAM1 兩個參數進入函數，然後指定 RETURN 變數值來回傳值，如下圖所示：



上述函數名稱是指【BMI 函數.fpp】檔，傳遞 2 個參數，和指定回傳值至變數 BMI，【BMI 函數.fpp】的流程圖，如下圖所示：



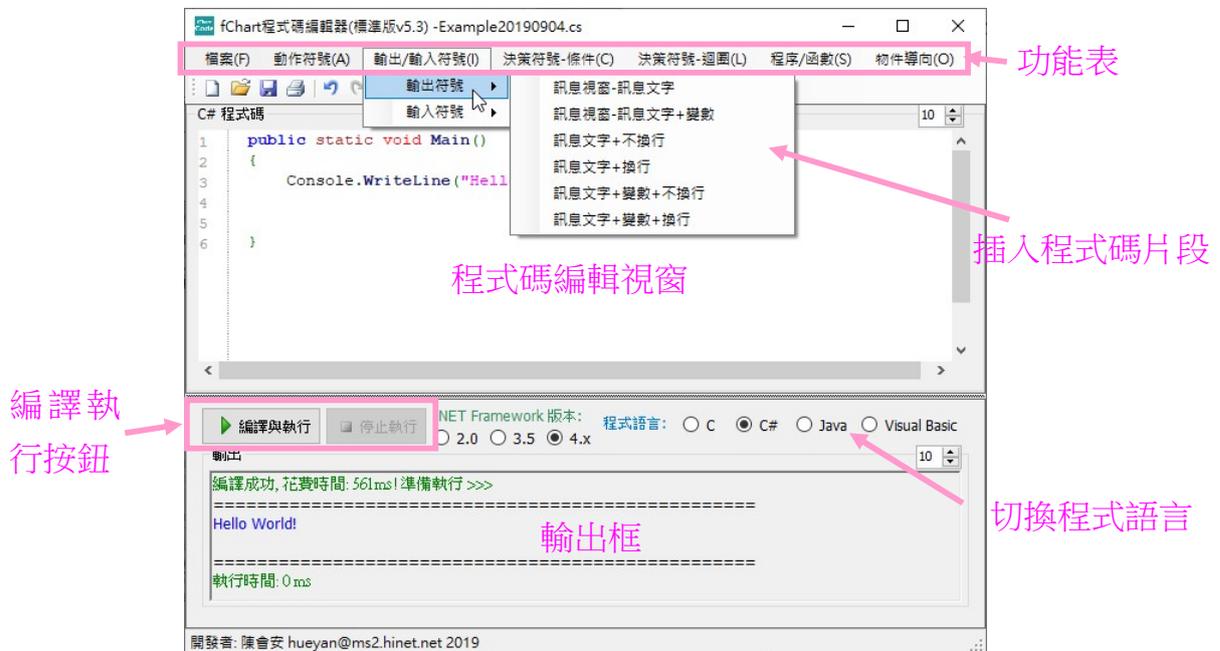
上述傳入函數的第 1 個參數是 PARAM；第 2 個是 PARAM1，最後指定 RETURN 的值，這就是回傳的 BMI 值。

fChart 程式碼編輯器與 Blockly 積木程式

fChart 程式設計教學工具內建輕量級整合開發環境和 Blockly 積木程式編輯器，當我們使用 fChart 流程圖直譯器了解流程圖的程式邏輯後，可以馬上拼出或寫出演算法的程式碼，各版本支援的程式語言有：C/C++、Java、C#、VB、Python、JavaScript 和 Node.js。

fChart 程式碼編輯器

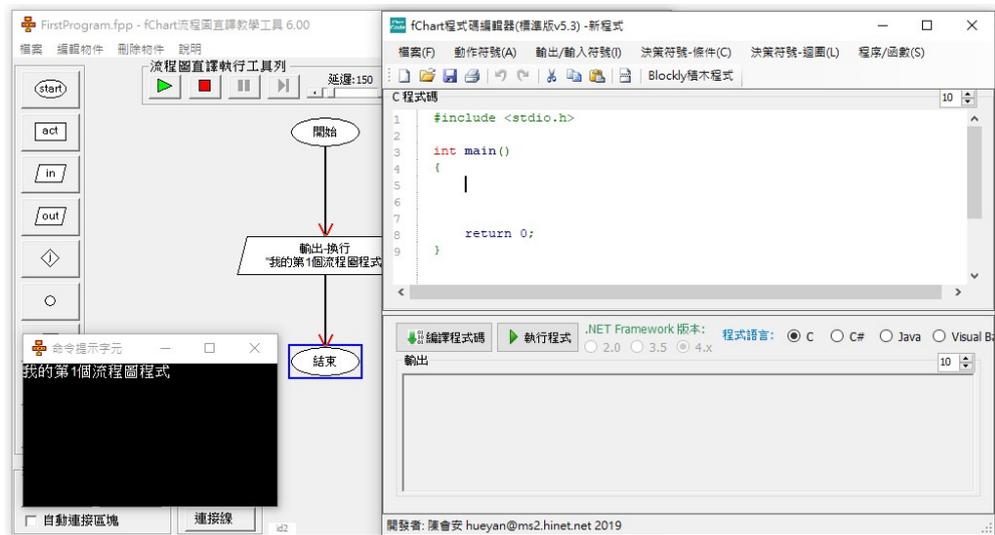
在 fChart 流程圖直譯器的執行工具列，按最後【程式碼編輯器】鈕，可以啟動 fChart 程式碼編輯器的整合開發環境，程式碼編輯器的介面，如下圖所示：



上述執行畫面上方是功能表，這些功能表命令是對應流程圖符號，可以幫助初學者建立程式碼和流程圖符號的對應關係，在下方是程式碼編輯視窗，右下方選擇使用的程式語言；左下方是編譯和執行按鈕，位在最下方是輸出框，可以輸出編譯訊息或 VB 和 C#程式的執行結果。

在 fChart 各版本的程式碼編輯器都內建 TCC 的 C 語言編譯器，可以馬上編輯、編譯和執行 C 程式，也可以編譯執行 C#和 VB 程式。以 C 語言為例，我們除了自行使用鍵盤輸入 C 程式碼外，也可以使用功能表命令來快速插入 C 程式碼片段後，然後小部分修改來完成 C 程式碼的輸入，其步驟如下所示：

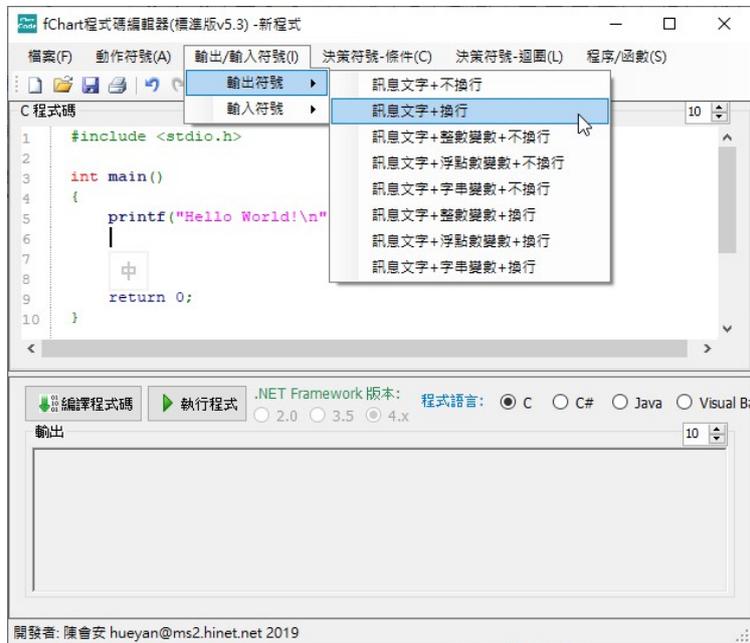
Step 1：請在 fChart 流程圖直譯器的上方工具列，按最後【程式碼編輯器】鈕啟動 fChart 程式碼編輯器，然後在右下方選項選【C】，切換成 C 程式語言，如下圖所示：



上述介面位在程式碼編輯區域右上角的數字是目前的字型尺寸，我們可以使用箭頭馬上調整程式碼顯示的字型尺寸。

Step 2：因為流程圖是使用輸出符號輸出一段訊息文字，請先在 `main()` 函數程式區塊中點一下作為插入點。

Step 3：請執行「輸出/輸入符號>輸出符號/訊息文字+換行」命令，可以插入 C 語言 `printf()` 函數的輸出程式碼，"`\n`"是換行。



Step 4：請將原來的字串內容「Hello World!」改為「我的第 1 個流程圖程式」。

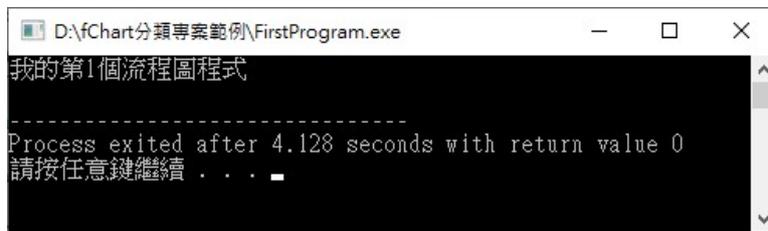


Step 5：執行「檔案>儲存」命令儲存檔案，可以開啟「另存新檔」對話方塊，請切換路徑後，在【檔案名稱】欄輸入檔名 FirstProgram.c，按【存檔】鈕儲存成 C 程式檔案（執行「檔案>開啟」命令可以開啟存在的 C 程式檔案）。

Step 6：請按下方【編譯程式碼】鈕編譯 C 程式，如果沒有錯誤，可以在下方顯示成功編譯的綠色訊息文字；錯誤是紅色的錯誤訊息文字。



Step 7：然後按【執行程式】鈕執行 C 程式，可以開啟「命令提示字元」視窗來顯示執行結果。



Blockly 積木程式編輯器

Blockly 是 Google 公司支援的開源專案，一套在客戶端執行的視覺化積木程式編輯器，源於 Scratch 和 App Inventor 的積木觀念，可以讓使用者不用撰寫文字內容的程式碼，直接拖拉積木來組合出積木程式。fChart 工具支援 2 種 Blockly 積木程式編輯器，如下所示：

- Cake Blockly for C：基於 Blockly 的積木程式編輯器，原是韓國 Joshua 團隊開發的 C 語言版本（<https://github.com/cra16/cake->

core)，筆者已經中文化、修正錯誤和增強功能，可以幫助初學程式者輕鬆拖拉積木來學習 C 程式設計，如下圖所示：



- **Blockly 積木程式：Python 和 Node 版支援**，這是修改源於"雲林 SONG"的精簡版本，只保留 Code 部分，目前版本已經修改支援整數和 Python 3（不再支援 Python 2），如下圖所示：

