# Unity 教學與學習

<https://www.gameislearning.url.tw/article_more.php?foo=9997&pgg=1&firstPG=1>

**Unity Script 常用語法教學(unity課程入門學習筆記)**

[**王啟榮**](https://www.gameislearning.url.tw/member_profile.php?mbr=1) 發表於 2018/08/20\_21:22　 ( 最後修改 2020/04/15\_15:56 )

● 在 Console 面板輸出訊息  
　Debug.Log("your message");　或　print("your message");  
　// print("<color=red>這樣能輸出紅色訊息</color>");  
  
● 鍵盤按鍵觸發 (通常寫在 Update 中)  
　if ( Input.GetKey("a") ){　}　// A 鍵被按住時持續觸發(每個 frame 執行一次)  
　if ( Input.GetKeyDown("a") ){　}　// A 鍵被按下瞬間觸發一次  
　if ( Input.GetKeyUp("a") ){　}　// A 鍵被放開瞬間觸發一次  
　※ 請參閱 [**鍵盤控制物體移動、旋轉**](https://www.gameislearning.url.tw/article_content.php?getb=32&foog=9997) 實作範例  
  
● 滑鼠按鍵觸發 (通常寫在 Update 中)  
　if ( Input.GetMouseButton(0) ){　}　// 滑鼠左鍵被按住時持續觸發(每個 frame 執行一次)  
　if ( Input.GetMouseButtonDown(0) ){　}　// 滑鼠左鍵被按下瞬間觸發一次

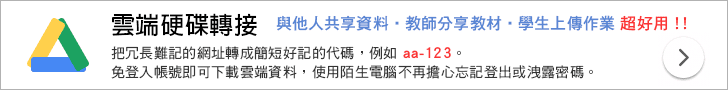
　if ( Input.GetMouseButtonUp(0) ){　}　// 滑鼠左鍵被放開瞬間觸發一次  
　0、1、2 分別代表滑鼠 左、右、中 鍵  
  
  
  
● Transform 相關  
　transform.Translate( x, y, z );　// 移動  
　transform.Rotate( x, y, z );　// 旋轉  
　transform.position = new Vector3( x, y, z );　// 設定位置(世界座標)  
　transform.localPosition = new Vector3( x, y, z );　// 設定位置(相對於母物體)

　transform.eulerAngles = new Vector3( x, y, z );　// 設定角度  
　transform.localEulerAngles = new Vector3( x, y, z );　// 設定角度(相對於母物體)  
　transform.LookAt( otherGameObject.transform );　// 讓物件的 Z 軸朝(指)向另一個物件的軸心  
  
　transform.forward　// 代表物件的 z 軸方向( unity以z軸為前方，建模時最好符合此規則，讓z軸朝前。)  
　　例如要讓物件朝著另一個物件的前面(z 軸方向)移動：  
　　transform.Translate ( otherGameObject.transform.forward );  
  
● 設定與解除連結(階層關係)  
　A.transform.parent = B.transform;　// 設定 B 為 A 的母物體  
　A.transform.parent = null;　// 解除 A 的母物體  
  
● 抓取物件、元件  
　　請參閱「[**unity 抓取物件(GameObject)和元件(Component)**](https://www.gameislearning.url.tw/article_content.php?getb=30&foog=9997)」這篇文章。  
  
  
  
● 啟用(或停用)物件、元件  
　yourGameObject.SetActive( true );　// 啟用物件  
　yourComponent.enabled = true;　// 啟用元件  
  
● 複製、刪除物件  
　Instantiate( yourObject );　// 動態產生 object  
　Destroy( yourObject );　// 刪除 object  
  
● 呼叫其他物件的函數  
　GameObject.Find("物件名稱").GetComponent<script名稱>().函數名稱();　  
　被呼叫函數的前面要加 public，例如： public void abc(){ }  
  
● 抓取其他 script 的變數(例如要在 script\_A 抓取 script\_B 的變數)  
　在 script\_B 宣告變數時，前面加上 static public (例如 static public int a )，  
　之後在 script\_A 打 script\_B.a ，即可取得 script\_B 的變數值。  
  
● 抓取(設定) Animator 中的參數(Parameter)  
　GetComponent<Animator>().GetBool( "參數名稱" );　// 取得參數的布林值  
　GetComponent<Animator>().SetBool( "參數名稱", 值 );　// 設定參數的布林值  
　把 Bool 改成 Integer 就是整數、改成 Float 就是浮點數數  
  
● 定時呼叫函數 (常用來製作計時器)  
　Invoke("函數名稱", 2);　// 2 秒後呼叫函數  
　InvokeRepeating("函數名稱", 1.5f, 0.3f);　// 1.5 秒後呼叫函數，之後每隔 0.3 秒都會呼叫函數  
　以上兩者的差別在於 Invoke 只會呼叫一次，而 InvokeRepeating 會重複呼叫。  
　CancelInvoke();　// 停止(取消)定時呼叫  
  
● 物理學(Physics)  
　Physics.gravity = new Vector3(0, -9.81f, 0);　// 設定重力  
　GetComponent<Rigidbody>().AddForce( x, y, z, 力量模式 );  
　// 對 Rigidbody 施加力量，(x,y,z)為世界座標，代表力量的強度和方向  
　// 力量模式填 ForceMode.Force 適用於持續的力、填 ForceMode.Impulse 適用於瞬間的力  
  
● 物件碰撞偵測  
　　請參閱「[**unity Collision Trigger 碰撞偵測與觸發對應表**](https://www.gameislearning.url.tw/article_content.php?getb=17&foog=9997)」這篇文章。  
  
● 重新開始、結束遊戲  
　SceneManager.LoadScene(0);　// 開啟索引號為 0 的場景，常用來重新啟動遊戲  
　場景索引號可在 Build Setting 視窗查看、script 開頭要加 using UnityEngine.SceneManagement;  
　Application.Quit();　// 退出遊戲、關閉 app  
  
● 一些 MonoBehaviour 常用的函數  
　void Start(){}　// 只有第一個 frame 會呼叫  
　void Update(){}　// 每個 frame 都會呼叫  
　void OnMouseDown(){}　// 套用 script 的 collider 物件被滑鼠點擊瞬間會呼叫  
  
● 材質、貼圖相關  
　public Texture2D myPic;  
　GetComponent<Renderer>().material.mainTexture = myPic;　// 設定材質的主要貼圖  
　GetComponent<Renderer>().material.color = new Color( r, g, b );  
　// 設定材質色彩( RGB 值為 0~1 )  
  
◎ Random.Range( min, max );　// 產生亂數(若都填整數，回傳值最大只會是 max-1)  
◎ GetComponent<RectTransform>().sizeDelta = new Vector2 ( x, y );  
　// 設定 RectTransform (UI物件)的寬、高

**unity C# 語法教學入門**

[**王啟榮**](https://www.gameislearning.url.tw/member_profile.php?mbr=1) 發表於 2018/08/27\_22:04　 ( 最後修改 2020/03/29\_17:58 )

● 註解： // 單行註解　　　/\* 多行註解 \*/  
  
● 宣告變數：　int a = 6;　// 宣告一個名稱為 a 的變數，資料類型為整數、值為 6  
　　　　　　　要把 a 加 1 可寫 a=a+1 或 a++;　　　要把 a 加 5 就寫 a+=5;  
  
● 常用資料類型：　int -> 整數  
　(注意大小寫)　　 float -> 浮點數(有小數點的數。數字後要加 f，例如 3.6f )  
　　　　　　　　　string -> 字串(前後都要加"，例如"abc")  
　　　　　　　　　bool -> 布林(值只有 true 或 false 兩種)  
　　　　　　　　　GameObject -> 遊戲物件  
　　　　　　　　　Vector3 -> 簡單來說就是一組(x,y,z)  
  
　◎ 字串與數字轉換：　int.Parse("67");　// string 轉 int  
　　　　　　　　　　　 float.Parse("25.36");　// string 轉 float  
　　　　　　　　　　　 65.ToString();　// 數字轉 string  
　　　　　　　　　　　 yourFloat.ToString("0.00");　// float 轉 string 並取到小數第2位  
  
　◎ 字串相關：　string[] words = yourString.Split('\_');　// 分割字串(以\_符號為例)

　　　　　　　　 yourString.Substring( 0, 2 );　// 抓取子字串(從第1個字開始連續抓2個)  
　　　　　　　　 yourString.Length;　// 抓取字串長度  
　　　　　　　　 yourString.ToLower();　// 把字串中所有大寫英文字母轉成小寫  
　　　　　　　　 yourString.ToUpper();　// 把字串中所有小寫英文字母轉成大寫  
  
　◎ 字串反轉：　// 要加 using System;  
　　　　　　　　string strA = "unity c#";  
　　　　　　　　char[] myArray = strA.ToCharArray();  
　　　　　　　　Array.Reverse(myArray);  
　　　　　　　　string strB = new string(myArray);　// 結果 strB 為 #c ytinu  
  
**[](https://www.gameislearning.url.tw/file-link.php)**  
  
● if 判斷式：  
　if ( n > 50 ){　}　// 如果 n 大於 50 就執行  
　else if ( n == 10 ){　}　// 如果 n 等於 10 就執行  
　else {　}　// 以上條件都不符合，就執行 else  
　同一個判斷式最多只會執行一個結果(最先符合條件者)，撰寫時要留意先後順序。  
  
　關係運算子:　等於==　不等於!=　大於>　大於等於>=　小於<　小於等於<=  
　邏輯運算子:　且&&　或||  
　例如: if ( a>5 && b==3 ){ }　代表要同時符合 a>5 和 b==3 這兩個條件才行  
　　　　if ( a>5 || b==3 ){ }　代表只要符合 a>5 或 b==3 其中一個條件即可  
  
　判斷布林值為 true 或 false 時:  
　　　if ( a==true )　可寫為　if ( a )  
　　　if ( a==false )　可寫為　if ( !a )  
  
● 陣列(array)：  
　　請參閱「[**unity Array 陣列資料存取**](https://www.gameislearning.url.tw/article_content.php?getb=29&foog=9997)」這篇文章。  
  
● 迴圈：  
　◎ for 迴圈  
　　for (int i=0; i<5; i++) {　print(i);　}  
　　說明：i 的啟始值為 0， i 小於 5 就繼續跑迴圈， i 每次加 1  
　　　　　結果會依序輸出 0、1、2、3、4  
  
　◎ while 迴圈  
　　int x = 0;  
　　while (x < 3) {　 x++;　print(x);　 }　// 結果會依序輸出 1、2、3  
  
  
  
● 函數(function)：  
　◎ 基本語法  
　　void test(){　}　// 宣告一個名稱為 test 的函數(前面加 void 代表沒有回傳值)  
　　test();　// 呼叫函數 test  
　　函數不會主動執行，被呼叫時才會執行。  
  
　◎ 呼叫加傳值  
　　void test( int a, string b ){　}　// 括號內的兩個參數，用來接收傳過來的值(資料類型要對應)  
　　test( 7, "hello" );　// 呼叫函數，且傳兩個值(引數)過去  
  
　◎ 回傳值  
　　int test(){　// 宣告函數時前面加 int，代表會回傳整數  
　　　　return 123;　// 回傳整數 123 (return 必需加在函數的最後)  
　　}  
　　int c = test();　// 呼叫函數，且宣告一個變數 c 以儲存回傳值(結果 c 為 123)  
　※ 若要多個回傳值，請參閱「[**呼叫函數返回多個回傳值**](https://www.gameislearning.url.tw/article_content.php?getb=4&foog=9997)」這篇文章。  
  
　◎ 傳址呼叫  
　　void myTest( ref int a ){　a = 12;　}  
　　int a = 5;  
　　myTest ( ref a );  
　　print ( a );　// 結果 a 為 12。 引數和參數前面加 ref，代表傳址而非傳值。  
  
● 類別(class)與物件(object)：  
　◎ 宣告類別  
　　新增一個名為 myCharacter 的 script，內容如下：  
　　public class myCharacter {　 public int \_hp;　 }　// 例如帶有 hp 值的角色藍圖  
  
　◎ 建立物件  
　　在另一個 script 輸入：　  
　　myCharacter npc = new myCharacter();  
　　npc.\_hp = 90;　// 設定 npc 物件的 hp 值  
　　※ 若要進一步了解，請參閱「**[封裝(Encapsulation)與資料存取](https://www.gameislearning.url.tw/article_content.php?getb=18&foog=9997)**」這篇文章。

**unity 按鍵控制物件移動和旋轉**

[**王啟榮**](https://www.gameislearning.url.tw/member_profile.php?mbr=1) 發表於 2019/02/22\_23:33

使用鍵盤控制操作物體，是最基本又常用的互動方式，這個例子將示範如何透過按鍵操控物體移動和旋轉。  
  
1. 新增一個 unity 專案。  
  
2. 點擊 GameObject > 3D Object > Cube 新增一個立方體。  
  
3. 新增一個 c# script 掛到立方體上。  
  
4. 在 script 中輸入以下程式即完成。

void Update () {  
   
 if ( Input.GetKey("up")) { transform.Translate( 0, 0, 0.1f ); }  
 // 按住 上鍵 時，物件每個 frame 朝自身 z 軸方向移動 0.1 公尺  
  
 if ( Input.GetKey("down")) { transform.Translate( 0, 0, -0.1f ); }  
 // 按住 下鍵 時，物件每個 frame 朝自身 z 軸方向移動 -0.1 公尺  
  
 if ( Input.GetKey("left")) { transform.Rotate( 0, -3, 0 ); }  
 // 按住 左鍵 時，物件每個 frame 以自身 y 軸為軸心旋轉 -3 度  
  
 if ( Input.GetKey("right")) { transform.Rotate( 0, 3, 0 ); }  
 // 按住 右鍵 時，物件每個 frame 以自身 y 軸為軸心旋轉 3 度  
  
 }

**unity 抓取物件(GameObject)和元件(Component)**

[**王啟榮**](https://www.gameislearning.url.tw/member_profile.php?mbr=1) 發表於 2018/03/14\_15:31

　　使用 unity 開發遊戲時，往往需要在程式( C# Script )中表達某個遊戲物件或元件。例如有一個 script 掛在 A 物件，但你卻想要在這個 script 寫程式控制 B 物件，這時候就得懂得抓取物件才行。以下列舉一些常用的方法。  
  
　　● 抓取物件(GameObject)  
　　　GameObject.Find("場景中的物件名稱");  
  
　　　例如要旋轉場景中名為 kitty 的物件：  
　　　GameObject.Find("kitty").transform.Rotate( 0, 2, 0 );  
  
　　　GameObject.Find 無法抓取原本不在場景中或未啟用(active為false)的物件。  
　　　若要取得這類物件，就先在 script 中 public GameObject myObj;  
　　　然後到 Inspector 面板為 public 變數指定物件。(例如從 Project 面板把 prefab 拉進去)  
　　　**[](https://www.gameislearning.url.tw/article_content.php?getb=30&foog=9997)**  
  
　　　◎ 抓取 tag 物件  
　　　　GameObject[] myObjArray;  
　　　　myObjArray = GameObject.FindGameObjectsWithTag("foo");  
　　　　// 所有 tag 為 foo 的物件，都會被抓出來存到變數(陣列) myObjArray 中。  
　　　　可參閱「[**unity Array 陣列資料存取**](https://www.gameislearning.url.tw/article_content.php?getb=29&foog=9997)」這篇文章。  
  
　　　◎ 抓取父物件  
　　　　transform.parent.gameObject;  
  
　　　◎ 抓取子物件　※ 只包含下一層級物體，不含孫輩以下  
　　　　transform.GetChild(2).gameObject;  
　　　　// 抓取第 3 個子物體(索引號是從 0 開始)，可透過 transform.childCount 取得子物體的數量。  
  
　　● 抓取元件(Component)  
　　　GetComponent<元件類型>();  
  
　　　例如要對 Rigidbody 元件使用 AddForce 方法：  
　　　GetComponent<Rigidbody>().AddForce( 0, 0, 5 );  
　　　除了 transform 可以直接打，其他元件都要用 GetComponent 去抓。  
  
　　　◎ 抓取所有子物件(含孫輩以下)的元件  
　　　　Component[] rdrs = yourGameObject.GetComponentsInChildren( typeof(Renderer), true );  
　　　　foreach (Renderer \_renderer in rdrs){　\_renderer.enabled = false;　}  
　　　　// 停用所有子物體的 Mesh Renderer 元件  
  
  
　　為了效能考量，不要在 Update 中重複抓取物件或元件，而是在 Start 中抓取一次儲存到變數中。例如：  
  
　　　GameObject box;  
　　　Rigidbody box\_rb;  
　　　void Start () {  
　　　　　box = GameObject.Find("Cube");  
　　　　　box\_rb = box.GetComponent<Rigidbody>();  
　　　}

**Unity Text 動態改變文字內容(以按鍵加分為例)**

[**王啟榮**](https://www.gameislearning.url.tw/member_profile.php?mbr=1) 發表於 2018/02/19\_23:27

動態文字的應用層面很廣，例如顯示在畫面上的分數、倒數計時秒數、玩家間的對話...等，這些文字內容都必需即時改變而非固定不變。這個例子將示範動態改變 unity text 的內容，每按一次空格鍵，畫面中的數字就會加 1。  
  
1. 新增一個 unity 專案。  
  
2. 點擊 GameObject > UI > Text 新增一個文字物件。  
  
3. 把文字內容改為 0 (原本為 New Text)，並調整好位置確認顯示在 Game 視窗中。  
  
4. 新增一個 c# script 掛到文字物件上。  
  
5. 在 script 中輸入以下程式即完成。

using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
using UnityEngine.UI; // 記得加這行  
  
public class NewBehaviourScript : MonoBehaviour {  
  
 int a = 0;  
  
 void Update () {  
 if (Input.GetKeyDown("space")) {  
 a = a + 1; // 也能簡寫為 a++; 或 a+=1;  
 GetComponent<Text>().text = "" + a; // 前面加空字串，是為了把 整數a 轉為 字串。  
 }  
 }  
}

**unity Array 陣列資料存取**

[**王啟榮**](https://www.gameislearning.url.tw/member_profile.php?mbr=1) 發表於 2018/12/09\_20:28

Array 可用來儲存同一資料類型、固定長度的資料，基本存取語法如下：  
  
string[] myArray;　// 宣告一個陣列，用來儲存字串資料  
myArray = new string[9];　// 定義陣列長度(同時會清空陣列)  
myArray = new string[] { "hello", "world", "game", "learning", "unity" };　// 定義陣列的值和長度  
myArray.Length;　// 陣列長度  
myArray[3];　// 取得陣列中第 4 筆資料的值(陣列的索引號是從 0 開始)  
myArray[2] = "book";　// 寫入陣列中第 3 筆資料的值  
foreach(string s in myArray){　print( s );　}　// 迴圈抓取陣列中所有資料  
  
　　在 unity 使用陣列，可以先在 script 中 public string[] myArray; ，然後到 Inspector 面板設定 Array，在 Size 欄輸入陣列長度、Element 欄輸入資料值。



**unity 限制旋轉角度**

[**王啟榮**](https://www.gameislearning.url.tw/member_profile.php?mbr=1) 發表於 2018/11/26\_20:48

　　物體旋轉是常見的動態，而許多情況必需對物件旋轉角度做限制，例如開關門片的角度、坦克車的砲管...等，以下為簡單的方塊旋轉範例供參考。  
  
1. 新增一個 unity 專案。  
  
2. 點擊 GameObject > 3D Object > Cube 新增一個立方體。  
  
3. 新增一個 c# script 掛到立方體上。  
  
4. 在 script 中輸入以下程式即完成。

void Update () {  
  
 if (transform.eulerAngles.y < 120) { transform.Rotate(0, 1, 0); }  
 // 如果 y 軸角度小於 120 度就旋轉  
  
 }

**呼叫函數返回多個回傳值**

chijung 發表於 2016/08/29\_00:29

void Start () {  
　　int a;　　string b;  
　　myFunction ( out a, out b );　// 在引數前加 out  
　　// 結果 a=12 、 b="hello"  
}  
  
void myFunction( out int a, out string b ){　// 在參數前加 out  
　　a = 12;　　b = "hello";  
}

**封裝(Encapsulation)與資料存取**

chijung 發表於 2017/02/07\_00:21

方法1：(透過方法存取)  
  
◎ 類別(class)內容：  
　　public class myCharacter {  
　　　　private int \_HP;  
　　　　public int getHP(){　return \_HP;　}  
　　　　public void setHP( int HP ){  
　　　　　　if (HP < 0) {　HP = 0;　}　// 限制 \_HP 最小為 0  
　　　　　　\_HP = HP;  
　　　　}  
　　}  
  
◎ 建立物件(object)：  
　　myCharacter npc = new myCharacter();  
　　npc.setHP ( 96 );  
　　print ( npc.getHP() );  
  
  
方法2：(透過屬性存取)  
  
◎ 類別(class)內容：  
　　public class myCharacter {  
　　　　private int \_HP;  
　　　　public int HP{  
　　　　　　get {　 return \_HP;　 }  
　　　　　　set {　 if ( value < 0 ) {　value = 0;　}　// value 為傳入的值  
　　　　　　　　　　\_HP = value;　 }  
　　　　}  
　　}  
  
◎ 建立物件(object)：  
　　myCharacter npc = new myCharacter();  
　　npc.HP = 33;  
　　print ( npc.HP );

**unity 載入外部文字檔(txt)**

[**王啟榮**](https://www.gameislearning.url.tw/member_profile.php?mbr=1) 發表於 2019/11/18\_11:32

從裝置本機或網路下載.txt檔的文字內容：  
  
using UnityEngine.Networking;  
  
void Start(){　StartCoroutine( LoadTxt() );　}  
  
IEnumerator LoadTxt() {  
　　string url = "d:/test/abc.txt";  
　　UnityWebRequest www = UnityWebRequest.Get(url);  
　　yield return www.SendWebRequest();  
  
　　if (www.isNetworkError || www.isHttpError){　　Debug.Log( www.error );　　}  
　　else{　　Debug.Log( www.downloadHandler.text );　　}  
}

**UnityWebRequest 下載雲端檔案**

chijung 發表於 2020/01/07\_01:06

Unity從網路伺服器下載檔案(各種雲端資源)的方法：  
  
using UnityEngine.Networking;  
  
void Start(){　StartCoroutine( Download\_File() );　}  
  
IEnumerator Download\_File(){　  
　　string url = "https://www.yourServer.com/unity.jpg";　// 欲下載圖片的網路位址  
　　string savePath = "d:/unity.jpg";　// 圖片下載後的儲存路徑  
  
　　var uwr = new UnityWebRequest(url);  
　　uwr.method = UnityWebRequest.kHttpVerbGET;  
　　var dh = new DownloadHandlerFile(savePath);  
　　dh.removeFileOnAbort = true;  
　　uwr.downloadHandler = dh;  
　　yield return uwr.SendWebRequest();  
　　if (uwr.isNetworkError || uwr.isHttpError) { Debug.Log(uwr.error); }  
　　else { Debug.Log("檔案已下載到:" + savePath); }  
}