因為SWIFT是非常注重安全性的語言，所以他預期設計師很清楚知道自己在做甚麼，不希望出現不該變動的變數被改變，所以將變數分成可改變 和不可改變 兩種，在一開始宣告就要明確地寫出來，看一下例子

特別的是，這是宣告的方式，在swift裡不需要一開始就明確地宣告變數的type，因為swift有型別辨識( type inteferr)，會自動去判斷店數所儲存的制式甚麼樣的值，而給定他的type (看例子)

另外因為swift有承襲C和objective C所以type很像

另外String 跟java有點像，可以用+或是 \( ) 來做字串的相接和差值，另外也有countElement( ) 計算字串的長度

另外比較特別的是tuple這個type，是可以把多個不同型別的值，組合成一個

很像可以放不同值的array，他好用的地方就在，當function要return value時，可以用tuple一次return多個想要的值

再來就是array和dictionary，他的array跟其他語言差不多，一樣index從0開始，dictionary就很像之前perl的 hash array，適用key來當索取資料的辦法，資料間是沒有順序的放在記憶體中 (看例子)，dictionary只要在前面加上dictionary就行

再來是struct和class，因為有承襲objective C 所以有物件的概念存在，可以用struct和class有彈性地做很多事，而取值時一樣是用 . ，比較要注意的是struct是value type ， class是reference type ，value type是直接複製一份，reference type是指向原本的變數，所以在改變第二個變數值時，value type的第一個變數不會改變，但是reference type會被改變 (看例子)

雖然 Dynamic Scoping 功能很強大，但是 programmer 比較難預期結果如何，所以很容易發生錯誤，因此 注重安全性的Swift 跟大多數語言一樣採用的是 Static scoping

最後提兩個特別的點，是swift裡面沒有pointer ，因為pointer很容易會發生很多問題，像是dangling pointer 等等，所以swift將這個部分拿掉，取而代之就是剛剛所講的class的方式，做call reference的動作

Optional 是swift裡特有的語法，因為swift不希望看到非預期性的狀況發生，所以有時候會對可能不存在的狀況用optional來表示 ( 看例子 ) 在字串轉數字時，有可能發生無發轉換的情形，所以在這裡，將無法轉換的字串應轉成亂碼，swift乾脆用optional處理成 nil

讓programmer 更好去處理