**浮點數型別**

1. **為甚麼會有浮點數?**

試問一個問題 : 0到1之間共有幾個整數?

答案很簡單，就是0個，非常明確!

再問一個問題 : 0到1之間共有幾個實數?

如果明白實數的定義，這題應該也不難，因為實數包含小數，所以這題的答案會是無限多個。

以上這兩個問題如果再繼續延伸，可以試想一下，我們需要多大的記憶體，才能夠有辦法儲存所有的實數呢?換個例子說，我們需要多大的記憶體才能夠儲存圓周率(3.14159265…)這種無限小數呢?所以今天要介紹的浮點數就是用來解決以上問題。

1. **浮點數簡介**

因為我們不可能為了可以表示所有實數，就讓記憶體無限擴增，所以想出了使用浮點數這個方法。浮點數簡單的說就是實數的約略值，為了要確保這個實數能夠在有限記憶體容量中被儲存，我們必須犧牲這個實數的有效數字(精度)，舉個例子:

0.0359856這個數字轉換為科學記號為3.59856 \* 10-2

若只能取三位有效數字 3.59856 \* 10-2  會變成 3.59 \* 10-2

而浮點數則會儲存其中的有效數字及指數，也就是359及-2，所以若是使用浮點數型別，實際上儲存的會變成這兩個整數。也因為原本要存的是很長一段實數，改為兩個整數，所以相同的記憶體容量，浮點數型別可以儲存的範圍會大於整數型別。

1. **浮點數的優缺點**

**優點:**

* 相較占用同樣大小的整數格式，浮點數可表示的數值範圍較廣，較不容易溢位。
* 可以表示小數。

**缺點:**

* 浮點數的表示可能只是約略值，在做精密計算時應使用高精度的浮點數型別。
* 浮點數型別相較於整數型別，因還需在浮點數與科學記號中轉換，所以運算時執行效率較差。

1. **C#中的浮點數型別**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **描述** | **精度(有效數字)** | **範圍** |
| **Float** | **32bit單精度** | **7** | **±1.5 × 10-45 to ±3.4 × 1038** |
| **Double** | **64bit倍精度** | **15~16** | **±5.0 × 10 -324 to ±1.7 × 10308** |
| **Decimal** | **128bit高精度** | **28~29** | **±1.0 × 10-28 to ±7.9 × 1028** |

從上表可以看出，decimal的有效位數很大，達到了28位，但是表示的數據範圍卻比float和double類型小。decimal類型並不是C#中的基礎類型，所以使用的時候會對計算時的性能有影響。decimal 類型是適合財務和貨幣計算的 128 位數據類型。

1. **資料來源**

<https://www.youtube.com/watch?v=Q9ViPkXCEtY>

<http://lj.soft.blog.163.com/blog/static/79402481201032210173381/>