**宣告常數透過Const或Readonly**

**何謂常數:**

* **固定值，不會被程式進行修改的值。**
* **Example: PI = 3.1415926…,永遠不會變。**

**Const使用方式:**

**static** **void** Main(**string**[] args)

{

**decimal** r = 2.0M;

//計算面積時，取出Math Class的常數PI來計算

//const修飾詞也有類似static的屬性，不須new就可直接使用

**decimal** area = r \* r \* Math.PI;

Console.WriteLine("面積大小為:{0}", area);

Console.Read();

}

**class** Math

{

**public** **const** **decimal** PI = 3.14159M;

}

**假設要再Main Class中重新指定Math.PI的值會無法編譯，因為PI加入const常數修飾子，無法被assign新的值。**

**Readonly使用方式:**

**static** **void** Main(**string**[] args)

{

**decimal** r = 2.0M;

//計算面積時，取出Math Class的常數PI來計算

//因為PI有static的屬性，不須new就可直接使用

**decimal** area = r \* r \* Math.PI;

Console.WriteLine("面積大小為:{0}", area);

Console.Read();

}

**class** Math

{

**public** **static** **readonly** **decimal** PI = 3.14159M;

}

Readonly可以加上static，方便在別的class被使用，以上結果和const範例相同，若要再Main Class中重新assign PI的值，也會報錯無法進行編譯。

**Const與Readonly的差異**

* Const與Readonly都有限制某值為常數的效果。
* Const只能使用於數值或字串，Readonly可以用在任何型別。
* Const效能稍加。
* Const是在編譯時產生的，Readonly是在程式運行時產生。
* Const能在Method內使用，Readonly不行

**結語:**

當要設定常數的時候我會以static readonly優先，因為Readonly是在程式執行時取得值相對於Const是在編譯時產生日後要改值會比較方便，且readonly適用各種型別。會讓我考慮使用Const設定常數只有一種情況，就是要在Method設常數之時(但要再method設常數的情況應該不多)。

**參考資料:**

<http://slmtsite.blogspot.tw/2016/10/c-36-constant.html>  
<https://dotblogs.com.tw/yc421206/archive/2011/06/06/27232.aspx>