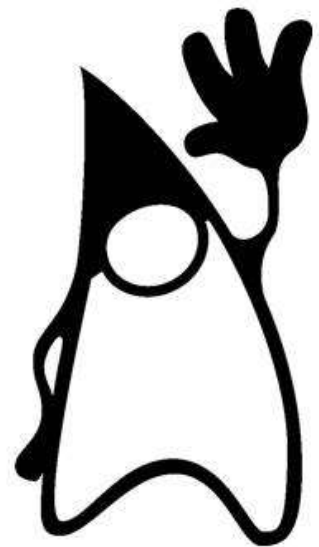




**END** ~~

Thank you!!



Edited by Ruei-Jiun Tzeng



J01-05

# 認識變數與Java基本型別

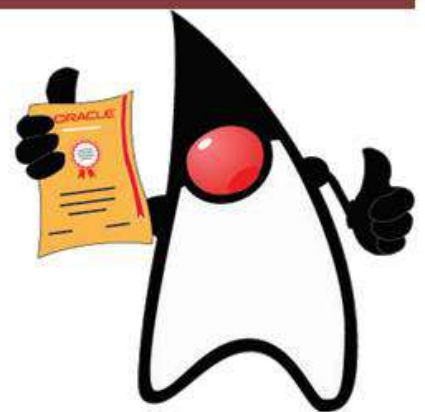
曾瑞君 (Jim\_Tzeng)

# 學習目標

1. 由計算機領域看Java
2. 認識8種Java的基本型別 (primitive data types)與變數 (variables) 、常數(constants)的使用
3. 使用運算子 (operators)
4. 使用 promotion 和 casting



Edited by Ruei-Jiun Tzeng



1/3

**認識8種Java的基本型別 (primitive data types)與變數 (variables) 、常數(constants)的使用**

# 在數學上，你一定聽過變數...

X 是一個變數，而且是一個未知數。

藉由“=”左右兩邊相同，算出X的數值

解一元一次方程式  $4x+5=x+11$

$$\Rightarrow 4x+5 -x = x+11 -x$$

$$\Rightarrow x+5 = 11$$

$$\Rightarrow x+5-5 = 11-5$$

$$\frac{x}{1} = \frac{6}{1}$$

# 現在，一切變得簡單了!!

- 在程式語言裡，X還是變數，但不用思考X是多少。通常都是我們直接給X某個數值，請電腦做一些運算。
- 使用變數前，我們必須先在程式碼中告知變數的**型別**，電腦才能**分配合適的記憶體空間**，才能開始計算。
- 變數宣告為**型別**後，  
可以一直「改變」數值，故名【**變數**】

```
public class MyMath {  
    public static void main(String[] args) {  
  
        int x = 1;  
        System.out.println( x + 2 );  
        System.out.println( x * 4 );  
  
        x = 6;  
        System.out.println( x / 3 );  
  
    }  
}
```



## 變數為何要有「型別」？

---

- 程式在執行的過程中，需要運算/儲存許多的資訊在記憶體空間中。
- 由於資料的型態各不相同，在儲存時所需要的容量不一，不同的資料必須要配給不同的空間大小來儲存，才能有效利用記憶體，因而有了「資料型別」的規範。
- Java 除了【**基本型別 (Primitive Type)**】的變數用於數學邏輯運算外；因為屬於物件導向程式，還有進階的【**參考型別 (Reference Type)**】的變數用於objects的儲存。



# Java 的基本 (Primitive) 型別

型態類型 (types)	關鍵字	位元組 (bytes)	位元數 (bits)	最小值	最大值
整數 Integral	byte	1	8	-128	127
	short	2	16	$-2^{15}$	$2^{15}-1$
	int	4	32	$-2^{31}$	$2^{31}-1$
	long	8	64	$-2^{63}$	$2^{63}-1$
浮點數 Floating point	float	單精確度, 32-bit 浮點數		依據 IEEE 754 標準	
	double	雙精確度, 64-bit 浮點數		依據 IEEE 754 標準	
字元 Textual	char	2	16	'\\u0000' - '\\uffff'	
布林值 Logical	boolean	1	1	true, false	

Edited by Kuei-Juan Tzeng



# 字面常量

- 可以在程式中寫直接寫下1、1.0、3.14159、'T'這樣的數值或文字，這類數值稱之為「字面常量」（Literal constant）或常數：
  - 字元
  - 字串
  - 整數：預設 `int`，若背後緊接 `l` 或 `L`，表示 `long` 型別
  - 浮點數：預設 `double`，若背後緊接 `f` 或 `F`，表示 `float` 型別
  - 符號
  - `true/false`

# 字面常量範例

```
public class LiteralConstantDemo {  
  
    public static void main(String[] args) {  
        System.out.println('J');  
        System.out.println("Java");  
        System.out.println(10);  
        System.out.println(10.10);  
        System.out.println("+");  
        System.out.println(true);  
    }  
}
```

Edited by Kuei-Jian Tzeng

## 用程式語言來計算時....

```
public class WithoutVariable {  
  
    public static void main(String[] args) {  
  
        // 給一個半徑 (r=5), 計算其 圓周 =>  $PI * 2r$   
        System.out.println("圓周 = " + 3.1415 * 2 * 5);  
  
        // 給一個半徑 (r=5), 計算其 面積 =>  $PI * r * r$   
        System.out.println("面積 = " + 3.14159 * 5 * 5);  
  
        // 半徑變2倍, 計算其 圓周  
        System.out.println("2倍半徑的 圓周 = " + 3.141 * 2 * (2*5) );  
  
        // 半徑變2倍, 計算其 面積  
        System.out.println("2倍半徑的 面積 = " + 3.14159 * (2*5) * (2*5) );  
  
    }  
}
```



## 發現不方便的地方了嗎?

---

- 程式裡有很多「字面常量 (Literal Constant)」反覆出現
- $\text{PI} = 3.14159$ ，多打幾次很容易打錯
- 半徑增加2倍，就程式架構看不太出來(如果沒有comment的話...)
- 以後若要改半徑，要改很多地方。漏改、錯改都會造成 bug

```
public class WithVariable {  
  
    public static void main(String[] args) {  
  
        final double PI = 3.14159;  
  
        int r = 5;  
  
        // 給一個半徑 (r=5), 計算其 圓周 =>  $PI * 2r$   
        System.out.println("圓周 = " + PI * 2 * r);  
  
        // 給一個半徑 (r=5), 計算其 面積 =>  $PI * r * r$   
        System.out.println("面積 = " + PI * r * r);  
  
        r = 10;  
  
        // 半徑變2倍, 計算其 圓周  
        System.out.println("2倍半徑的 圓周 = " + PI * 2 * r);  
  
        // 半徑變2倍, 計算其 面積  
        System.out.println("2倍半徑的 面積 = " + PI * r * r);  
  
    }  
}
```

## 也可以再進階...

```
public class WithVariable2 {  
    public static void main(String[] args) {  
        int r = 5;  
        showResult(r);  
  
        r = 10;  
        showResult(r);  
    }  
  
    private static void showResult(int r) {  
        final double PI = 3.14159;  
        System.out.println("圓周 = " + PI * 2 * r);  
        System.out.println("面積 = " + PI * r * r);  
    }  
}
```



## 變數宣告的邏輯意義

- 比較兩者，可以發現結果相同，但程式邏輯卻清楚很多!! 這是「變數宣告」在「程式邏輯」上的意義之一。雖然程式目的達成，但也要讓別人看得懂，也要考慮未來維護的成本：
  1. 字面常量改宣告為變數，放在計算式前面。當 $r$ 變成 $2*r$ ，邏輯清楚。
  2.  $PI$ 被宣告成常數，可以直接拿來使用，更不用擔心值被修改。
  3. 計算式由原本的一堆數字的加減乘除，變成公式，程式邏輯清楚。
  4. 日後改變半徑 $r$ 值時，輕鬆簡單!!



## 變數宣告的物理意義

- 用程式碼做資料運算時，因為CPU沒有記憶的功能，因此剛做完處理的資料馬上便消失，為了能取出剛才的運算結果，繼續加工加料，我們會需要記錄某些資料，可能是文字，也可能是數字，我們把這些資料記錄在記憶體某個位址中。
- 因為這個位址是電腦自己決定，位址本身不容易讓人們記住，所以我們給它一個名稱，讓這個名稱代表這個位址，這就是變數。由於記憶體的容量是有限的，而每個資料所佔用的空間必須定義清楚，因此Java定義了基本變數型態，不同型態佔用大小不等空間。
- 使用方式：變數 = 資料 如： $X=X+1$
- 這裡的“=”並不是指數學上的“等於”，而是“指定複製給”

# 宣告變數

- 「字面常量」是不需經過「宣告」即可在程式中使用；變數 (Variable) 因為會隨程式執行而變動，因此需要經過「宣告」程序：
  - 定義一個英文名稱代表該變數，使用駝峰命名原則
  - 定義其資料型別
  - 若是field，可以再加上 [modifiers]

```
int intNum;      // 宣告一個整數變數
double dblNum;   // 宣告一個倍精度浮點數變數
float x = 10, y = 20; //同時宣告多個變數屬同一型別，型別不重複
```
- 經過「宣告」後，系統就會配置一塊記憶體空間供其使用。

`int x;`



`x = 2;`

`x = 4;`

`x = 6;`

`x = 8;`

`x = 10;`

## 宣告常數

- 有時候一但將數值指定給變數之後，就不允許再「改變」，則可以在宣告變數時使用 **final** 關鍵字來限定，如果程式中有其它程式碼試圖改變這個變數，編譯器會先檢查出這個，例如：

```
final double PI = 3.14;
```

```
PI = 3.14159; // compile error!!
```

- 使用 **final** 來限定的變數，目的通常就是不希望其它的程式碼來變動它的值，例如圓周率PI的指定。



# The **scope** of Variable

**Instance** variable

**Local** variable

```
public class Shirt {  
  
    public int size = 10;  
    public double price = 100.5;  
  
    public void display() {  
        int size = 5;  
        System.out.println(size);  
        System.out.println(price);  
        /* if (5 > 2) {  
            int size = 9; // compile error!  
            System.out.println(size);  
        }*/  
    }  
    public static void main(String[] args) {  
        new Shirt().display();  
    }  
}
```