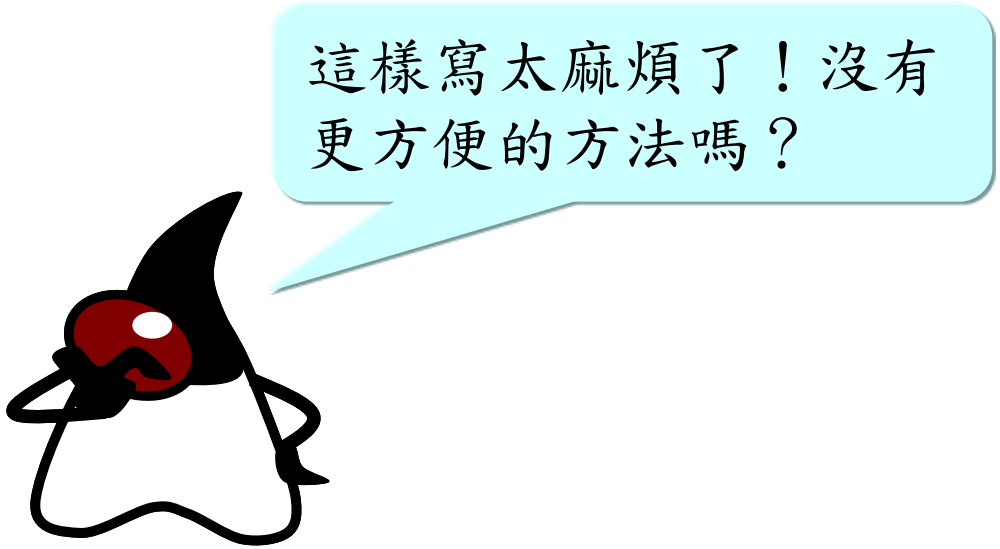
陣列

單元概論

- 單元目標
  - 使用一維陣列（One-dimensional array）
  - 利用length屬性及迴圈來設定陣列的值
  - 使用foreach語法存取陣列元素
  - 建立二維陣列（Two-dimensional array）
- 問題與討論

一維陣列

```
int ageOne = 27;
int ageTwo = 12;
int ageThree = 82;
int ageFour = 70;
int ageFive = 54;
int ageSix = 6;
int ageSeven = 1;
int ageEight = 30;
int ageNine = 34;
int ageTen = 42;
```



宣告一維陣列參考

```
type[] array_identifier;
```

其中：

- type表示儲存在陣列中的值為基本資料型態或是物件型態。
- [ ] 用來告知編譯器，您所宣告的是一維陣列參考名稱。
- array_identifier是您指派給陣列的名稱。

宣告一維陣列參考

以下的程式碼為宣告一個名為status的char一維陣列，及一個名為ages的int一維陣列

```
char[] status;
int[] ages;
```

以下的程式碼宣告了一個名為score的Integer陣列：

```
Integer[] scores;
```

實例化一維陣列

實例化陣列物件的語法為：

```
array_identifier = new type[length];
```

其中：

- array_identifier是您宣告的一維陣列參考名稱。
- type用來表示儲存在陣列中的元素之所屬型態。

length用來表示陣列的大小（元素的數目）。

實例化一維陣列

以下的程式碼用來建立一維char陣列與一維int陣列，並分別指定給status與ages一維陣列參考名稱：

```
status = new char[20];
ages = new int[5];
```

以下的程式碼用來建立一維Integer陣列實例，並指定給score名稱參考：

```
scores = new Integer[5];
```

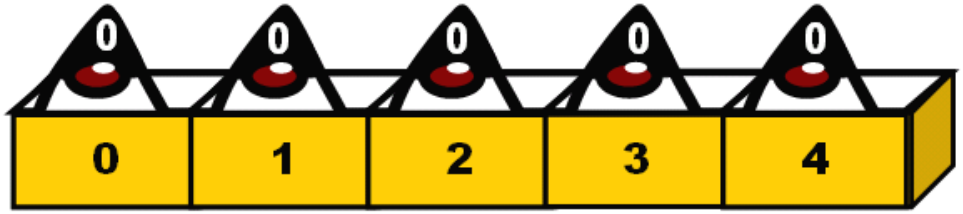
實例化一維陣列

陣列元素初始值

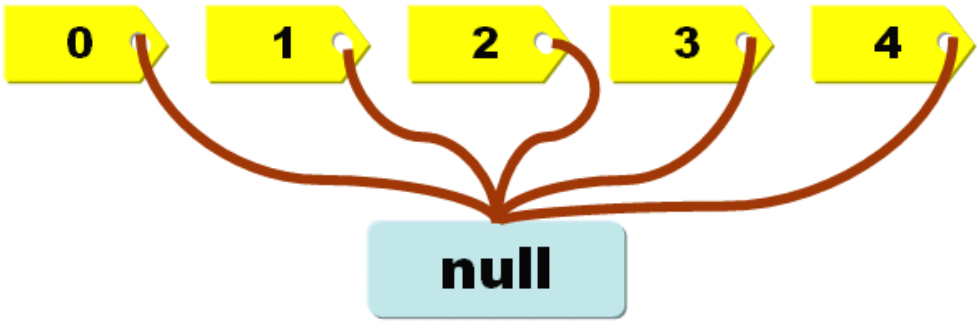
資料型態	初始值
byte	0
short	0
int	0
long	0L
float	0.0F
double	0.0D
char	\u0000
boolean	false
Object	null

實例化一維陣列

```
int[] ages = new int[5];
```




```
Integer scores = new Integer [5];
```



存取一維陣列元素

設定特定值給陣列元素的語法如下：

```
array_identifier[index] = value;
```

其中：

- array_identifier是陣列的名稱。
- index表示該值將被放置在陣列中的位置。

value為要指派至陣列index位置的值。

存取一維陣列元素

以下是從陣列中取出值的程式範例：

```
char s = status[0];
int age = ages[1];
Integer score = scores[2];
```

以下則是設定陣列之元素值的程式範例：

```
status[0] = '3';
ages[1] = 19;
scores[2] = new Integer(10);
```

必須特別注意的是，陣列的索引值是從0開始

宣告、實例化及初始化一維陣列

以下為結合宣告、實例化及設定值的語法：

```
type[] array_identifier = {,分隔的值或運算式} ;
```

其中：

- type用來表示儲存在陣列中元素之型態。
- []用來告知編譯器，此宣告為一維陣列。
- array_identifier是所使用的陣列名稱。

{,分隔的值或運算式}表示一連串您想要儲存在陣列內的值，或一連串的運算式，這些運算式會將其運算的結果儲存在陣列內。

宣告、實例化及初始化一維陣列

以下的程式碼結合陣列的宣告、實例化及初始化：

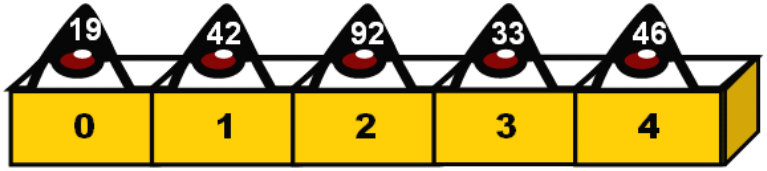
```
int[] ages = {19, 42, 92, 33, 46};
char[] name = {'J', 'a', 'v', 'a'};
```

以下的程式碼示範Integer物件陣列的宣告、實例化及初始化：

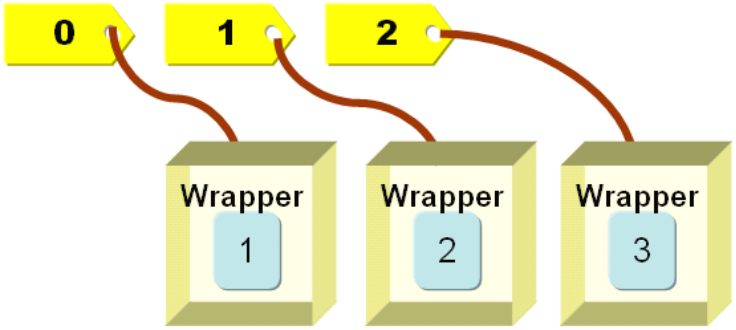
```
Integer[] scores = {new Integer(1),
new Integer(2), new Integer(3)};
```

宣告、實例化及初始化一維陣列

```
int[] ages = {19, 42, 92, 33, 46};
```




```
Integer[] scores = {new Integer(1),
                    new Integer(2),
                    new Integer(3)};
```



宣告、實例化及初始化一維陣列

您也可以在使用new關鍵字建立陣列實例時，一併初始化陣列元素：

```
int[] ages = new int[] {19, 42, 92, 33, 46};
```

這個語法其實作用類似於以下的寫法：

```
int[] ages = {19, 42, 92, 33, 46};
```

length屬性

您可以在陣列名稱之後加上.運算子，之後接著length屬性來取得陣列的長度值，例如下面的程式碼片段將顯示陣列的長度5：

```
int[] ages = {32, 22, 43, 11, 55};
System.out.println(ages.length);
```

存取陣列邊界外的元素，則您將會得到錯誤訊息，程式會丟出ArrayIndexOutOfBoundsException例外。

使用迴圈存取陣列元素

以下的例子說明如何利用迴圈來從陣列的開始到結束逐一設定初值。

```
int[] myArray;
myArray = new int[10];
for(int count = 0;
            count < myArray.length;
            count++) {
    myArray[count] = count;
}
```

使用foreach語法取得陣列元素

- 從JDK 5.0開始新增了foreach語法，又稱之為Enhanced for loop

```
for(type element : array) {
    deal with element;
}
```

其中：

- type宣告的元素型態必須與陣列的元素型態相同。
- array是您想要循序走訪的元素。
- 每一次的迴圈，都會將下一個陣列元素設定給element。

宣告二維陣列參考

以下是宣告二維陣列參考的語法。

```
type[][] array_identifier;
```

其中：

- type用來表示儲存在陣列中的元素之型態。
- []用來告知編譯器，您宣告的為二維陣列參考。
- array_identifier是陣列的名稱。

底下的範例說明如何宣告int型態的二維陣列參考：

```
int[][] yearlySales;
```

實例化二維陣列

以下是實例化二維陣列的語法：

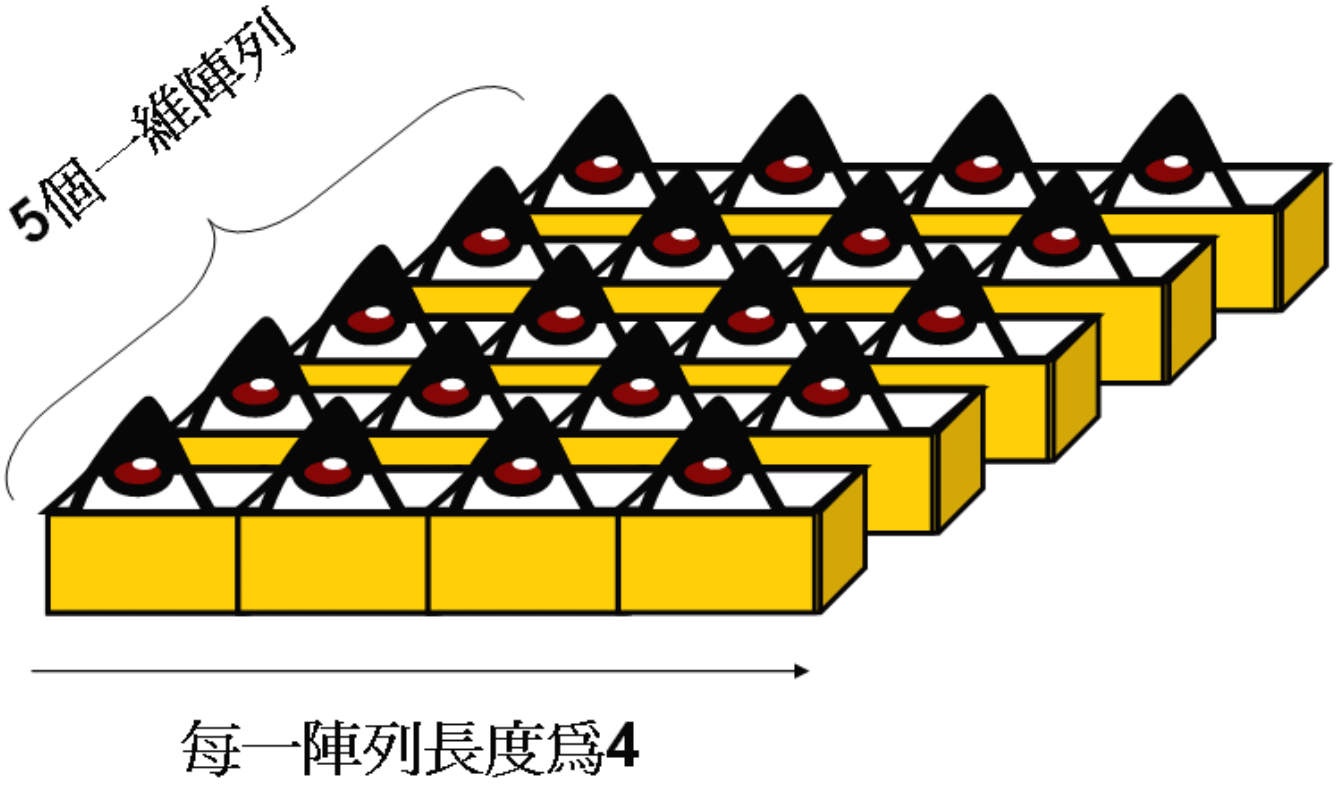
```
array_identifier =
        new type[number_of_arrays] [length];
```

其中：

- number_of_arrays為array_identifier陣列中擁有的一維陣列數量。
- length是array_identifier中每個一維陣列的長度。

實例化二維陣列

```
int[][] yearlySales = new int[5][4];
```



實例化二維陣列

```
Integer[][] yearlySales = new Integer[5][4];
```

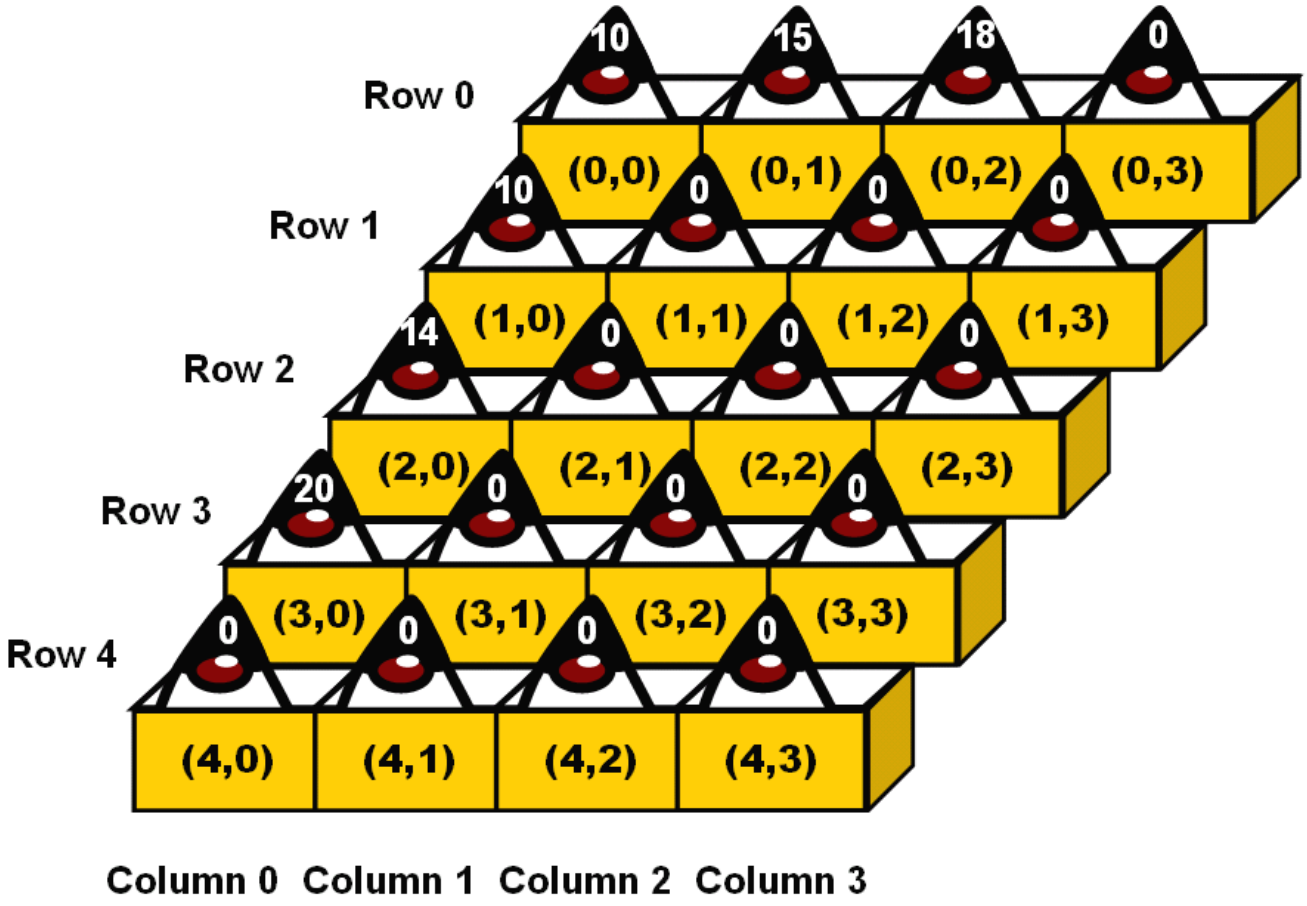
存取二維陣列元素

- 當設定二維陣列中的值時，利用兩個數值當作陣列的索引，其中第一個索引值代表列（Row），而第二個數值代表行（Column）

底下顯示如何設定yearlySales陣列的值：

```
int[][] yearlySales = new int[5][4];
yearlySales[0][0] = 10;
yearlySales[0][1] = 15;
yearlySales[0][2] = 18;
yearlySales[1][0] = 10;
yearlySales[2][0] = 14;
yearlySales[3][0] = 20;
```

存取二維陣列元素



宣告、實例化及初始化二維陣列

以下爲結合宣告、實例化及設定值的語法：

```
type[][] array_identifier =
{{,分隔的值或運算式},
{,分隔的值或運算式}};
```

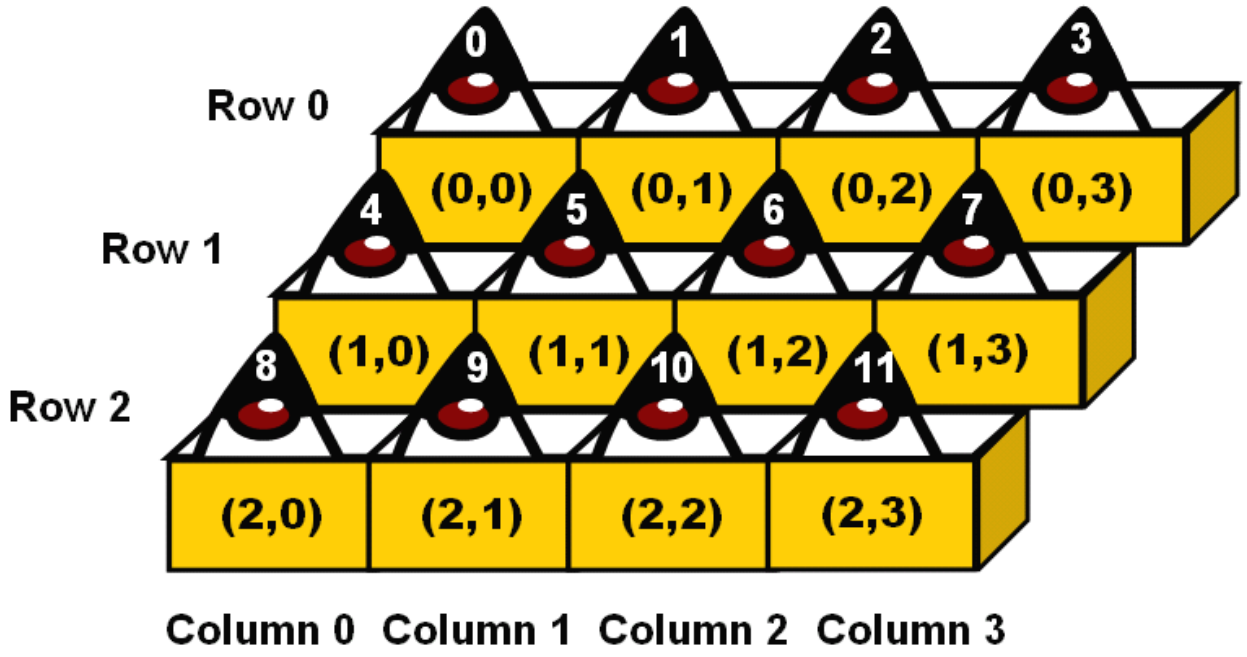
其中：

- type用來表示儲存在陣列中元素之型態。
- []用來告知編譯器，此宣告爲二維陣列。
- array_identifier是所使用的陣列名稱。
- {,分隔的值或運算式}表示一連串您想要儲存在該一維陣列內的值，或一連串的運算式，每個一維陣列以{}及逗號分隔。

宣告、實例化及初始化二維陣列

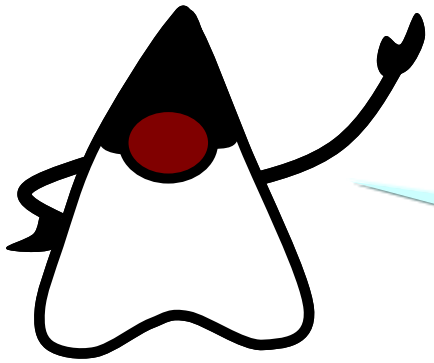
以下的程式碼結合陣列的宣告、實例化及初始化：

```
int[][] yearlySales = {{0, 1, 2, 3},
                       {4, 5, 6, 7},
                       {8, 9, 10, 11}};
```



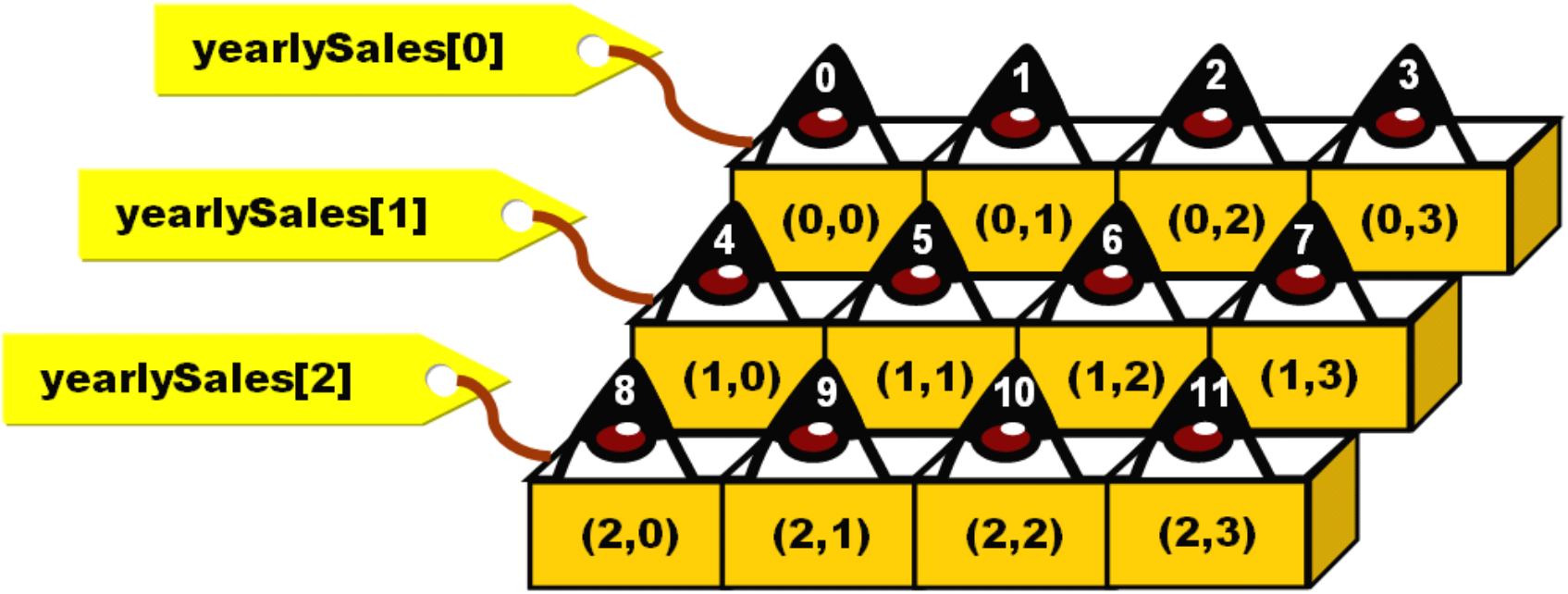
宣告、實例化及初始化二維陣列

```
int[][] yearlySales =
        new int[][]{{0, 1, 2, 3},
                    {4, 5, 6, 7},
                    {8, 9, 10, 11}};
```



不必指定陣列長度，程式會自動判斷長度

length屬性與二維陣列



如果您想要取得二維陣列中某列（Row）一維陣列的長度，則可以使用像yearlySales[0].length、yearlySales[1].length、yearlySales[2].length來取得。

length屬性與二維陣列

```
int[][] yearlySales = {{0, 1, 2, 3},
                       {4, 5, 6, 7},
                       {8, 9, 10, 11}};
for(int i = 0;
          i < yearlySales.length;
          i++) {
    for(int j = 0;
              j < yearlySales[i].length;
              j++) {
        System.out.print(
             yearlySales[i][j] + " ");
    }
}
```

foreach語法與二維陣列

```
for(int[] row : yearlySales) {
    for(int element : row) {
        System.out.print(element + " ");
    }
    System.out.println();
}
```

不規則陣列

下面的程式碼中，arr將管理兩個一維陣列：

```
int[] arr;
arr = new int[2][];
```

分別實例化長度為3與5的一維陣列：

```
arr[0] = new int[3];
arr[1] = new int[5];
```