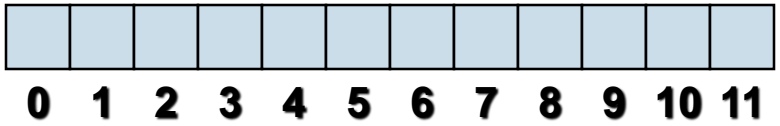
อาร์เรย์หลายมิติ

Multi-Dimensional Arrays

วีระยุทธ คุณรัตนสิริ

มิติของอาเรย์

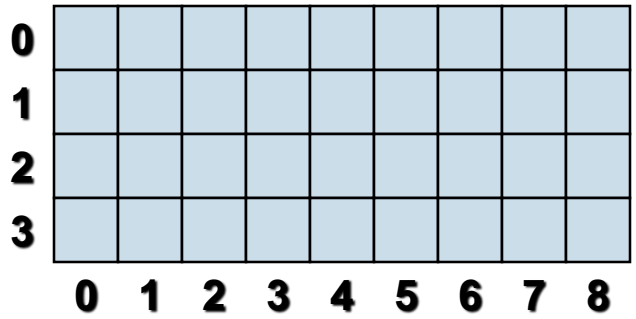
อาเรย์หนึ่งมิติ



ระบุจำนวนเต็มหนึ่งค่าสำหรับเป็นดัชนี

A[7]

อาเรย์สองมิติ



ระบุจำนวนเต็มสองค่าสำหรับเป็นดัชนี

A[1,4]

การประกาศอาเรย์หลายมิติ

สองมิติ

```
<type>[,] <name>;
```

- ตัวอย่าง

```
int[,] Array2D;
```

สามมิติ

```
<type>[,,] <name>;
```

- ตัวอย่าง

```
int[,,] Array3D;
```

การสร้างอาเรย์หลายมิติ

สองมิติ

```
<name> = new <type>[<dim1>,<dim2>];
```

- ตัวอย่าง

```
int[,] Array2D;
Array2D = new int[3,5];
```

สามมิติ

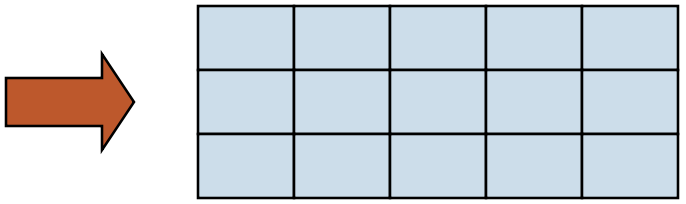
```
<name> = new <type>[<dim1>,<dim2>,<dim3>];
```

- ตัวอย่าง

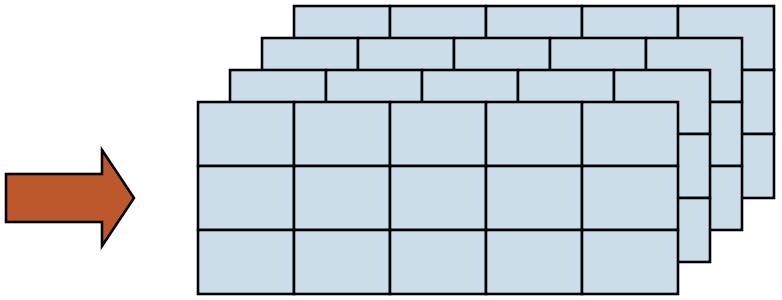
```
int[,,] Array3D;
Array3D = new int[3,5,2];
```

ตัวอย่าง

```
int[,] Array2D;
Array2D = new int[3,5];
```



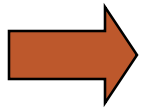
```
int[,] Array3D;
Array3D = new int[4,3,5];
```



การกำหนดค่าเริ่มต้น

แบบที่หนึ่ง เช่น

```
int[,] Array2D;
Array2D = new int[2,3] {{5,4,3},{2,1,0}};
```



5	4	3
2	1	0

แบบที่สอง เช่น

```
int[,] Array2D;
Array2D = new int[,] {
    {5,4,3},
    {2,1,0}
};
```

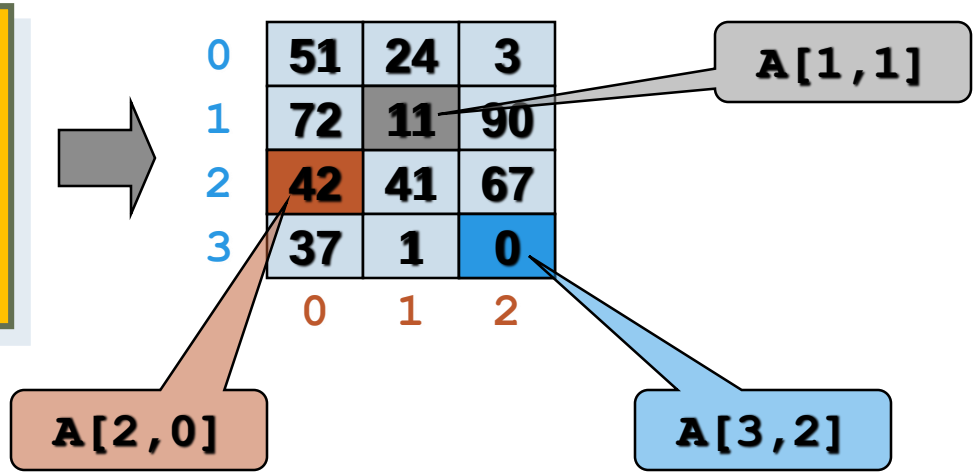
แบบที่สามอย่างย่อ เช่น

```
int[,] Array2D;
Array2D = {
    {5,4,3},
    {2,1,0}
};
```

การเข้าถึงข้อมูลในอาเรย์

จำนวนตัวเลขที่ใช้เป็นดัชนีขึ้นอยู่กับจำนวนมิติของอาเรย์

```
int[,] A = {
    {51,24,3},
    {72,11,90},
    {42,41,67},
    {37,1,0}
};
```

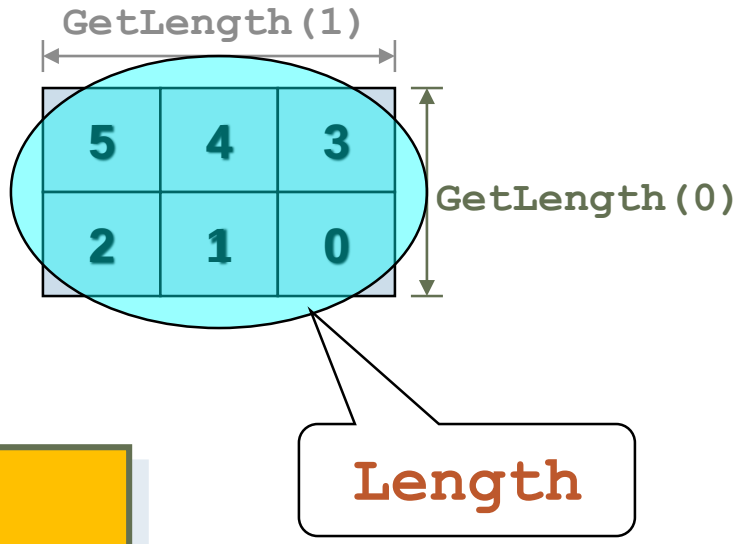


ขนาดของอาเรย์

ทบทวน

- *Length* property
- *GetLength* method

ตัวอย่าง




```
int[,] Array2D = {
    {5,4,3},
    {2,1,0}
};
Console.WriteLine(Array2D.GetLength(0));
Console.WriteLine(Array2D.GetLength(1));
Console.WriteLine(Array2D.Length);
```

แบบฝึกหัด

พิจารณาอาเรย์

```
int[,] A = {
    {5,34,3,1},
    {27,11,90,9},
    {24,41,67,6},
    {73,1,0,4},
    {1,2,3,4}
};
```

จงหาค่าของนิพจน์ต่อไปนี้

A[2,2]	
A[3,0]	
A[1,3]	
A.Length	
A.GetLength(0)	
A.GetLength(1)	

แบบฝึกหัด

พิจารณาอาเรย์

```
int[,] B = {
    {5,2,3,1,0},
    {11,9,10,8,2},
    {-1,20,4,33,12},
    {10,11,8,9,7}
};
```

จงหาค่าของนิพจน์ต่อไปนี้

B[2,2]	
B[3,0]	
B[1,3]	
B.Length	
B.GetLength(0)	
B.GetLength(1)	

การประยุกต์ใช้อาเรย์สองมิติ: *เมตริกซ์*

กำหนดให้ *A* และ *B* เป็นเมตริกซ์ขนาด 3x3

$$A = \begin{bmatrix} 1 & 3 & 2 \\ 9 & 12 & 0 \\ 5 & 4 & 6 \end{bmatrix} \qquad B = \begin{bmatrix} 4 & 1 & 3 \\ 5 & 2 & 6 \\ 6 & 3 & 9 \end{bmatrix}$$

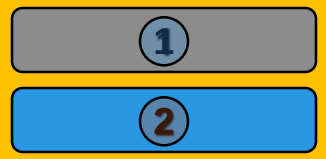
การประยุกต์ใช้อาเรย์สองมิติ: *เมตริกซ์*

คำนวน *A+B*

```
using System;
class Matrix {
   public static void Main() {
      int [,] A = 
      int [,] B = 

      int i,j;

      for (i = 0; i < 3; i++) {
         for (j = 0; j < 3; j++) {
            Console.Write("{0,4}", A[i,j]+B[i,j]);
         }
         Console.WriteLine();
      }
   }
}
```



เมท็อดที่สำคัญสำหรับเมตริกซ์

`ReadMatrix()`

`ShowMatrix()`

เมท็อด *ReadMatrix()*

รับค่าพารามิเตอร์สองค่าคือ

- จำนวนแถว, ***nr***
- จำนวนหลัก, ***nc***

ทำหน้าที่อ่านค่าจำนวนเต็ม ***nr***×***nc*** ค่าจากผู้ใช้

คืนค่าเป็นอาเรย์สองมิติซึ่งแทนเมตริกซ์

เมท็อด *ReadMatrix()*

```
static int[,] ReadMatrix(int nr, int nc) {
  int[,] m = new int[nr,nc];
  for (int i = 0; i < nr; i++) {
    for (int j = 0; j < nc; j++) {
      Console.Write("Element [{0},{1}]: ", i+1, j+1);
      m[i,j] = int.Parse(Console.ReadLine());
    }
  }
  return m;
}
```

เมท็อด *ShowMatrix()*

รับค่าพารามิเตอร์เป็นอาเรย์สองมิติซึ่งแทนเมตริกซ์

แสดงข้อมูลในเมตริกซ์ออกหน้าจอ

ไม่มีการคืนค่าใดใด

```
static void ShowMatrix(int[,] m) {
  for (int i = 0; i < m.GetLength(0); i++) {
    for (int j = 0; j < m.GetLength(1); j++) {
      Console.Write("{0,3}", m[i,j]);
    }
    Console.WriteLine();
  }
}
```

โปรแกรมตัวอย่าง

ตัวอย่างข้างล่างแสดงการเรียกใช้เมท็อด ***ReadMatrix*** และ ***ShowMatrix***

- แสดงเฉพาะส่วน ***Main*** เท่านั้น

```
static void Main() {
  int[,] A;
  Console.Write("Enter #rows: ");
  int nrows = int.Parse(Console.ReadLine());
  Console.Write("Enter #columns: ");
  int ncols = int.Parse(Console.ReadLine());
  Console.WriteLine("Enter Matrix A");
  A = ReadMatrix(nrows, ncols);
  Console.WriteLine("Matrix A is");
  ShowMatrix(A);
}
```

การคำนวนเมตริกซ์

เราสามารถเขียนโปรแกรมเพื่อคำนวนเมตริกซ์ได้เช่น

- การบวกสองเมตริกซ์
- การคูณสองเมตริกซ์
- การคำนวนหาดีเทอร์มิแนนท์
- การหาทรานสโพสของเมตริกซ์

การบวกเมตริกซ์

เมท็อด **AddMatrix** ทำการบวกสองเมตริกซ์ และส่งผลลัพธ์เป็นเมตริกซ์ใหม่ โดยสมมติว่าเมตริกซ์ทั้งสองมีขนาดเท่ากัน

```
static int[,] AddMatrix(int[,] a, int[,] b) {
  int nr = a.GetLength(0);
  int nc = a.GetLength(1);
  int[,] result = new int[nr,nc];
  for (int i = 0; i < nr; i++) {
    for (int j = 0; j < nc; j++) {
      result[i,j] = a[i,j] + b[i,j];
    }
  }
  return result;
}
```

การคูณเมตริกซ์

A เป็นเมตริกซ์ขนาด $m \times n$ และ B เป็นเมตริกซ์ขนาด $n \times p$

$$(A \times B)[i,j] = A[i,1] \times B[1,j] + A[i,2] \times B[2,j] + \ldots + A[i,n] \times B[n,j]$$

เมท็อด **`MulMatrix`** ทำการคูณเมตริกซ์ และส่งผลลัพธ์ในเมตริกซ์ใหม่

- มมติว่า #cols ของเมตริกซ์แรกมีค่าเท่ากับ #rows ของเมตริกซ์ที่สอง

การคูณเมตริกซ์

```
static int[,] MulMatrix(int[,] a, int[,] b) {
  int[,] result = new int[a.GetLength(0), b.GetLength(1)];
  for (int i = 0; i < a.GetLength(0); i++) {
    for (int j = 0; j < b.GetLength(1); j++) {
      int sum = 0;
      for (int k = 0; k < a.GetLength(1); k++)
        sum += a[i,k]*b[k,j];
      result[i,j] = sum;
    }
  }
  return result;
}
```