

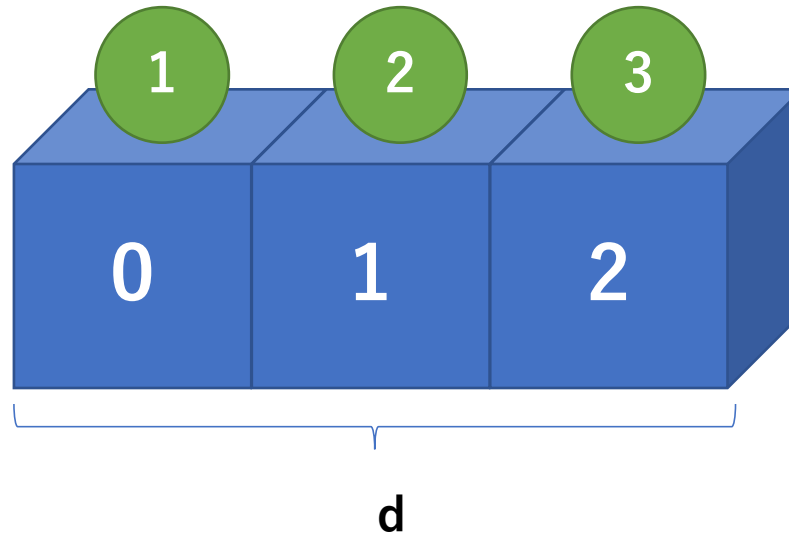


# 配列変数

基本編5日目

# 配列（はいれつ）

- **配列（はいれつ）**は、変数の箱が連なったもの
- その連なった箱1つ1つを「**要素（ようそ）**」と言う
- 複数のデータを扱うために必要
- 要素には一意の数（**添字（そえじ）**）がついている



# 配列のサンプル①

## Sample502

```
double[] d = new double[3];  
d[0] = 1.2;  
d[1] = 3.7;  
d[2] = 4.1;  
double sum, avg; // 合計値、平均値を入れる変数  
sum = 0.0;  
for(int i = 0; i < d.Length; i++){  
    Console.Write(d[i] + " ");  
    sum += d[i];  
}
```

# 配列の初期化

Sample502

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

↓ この処理は、以下の処理に相当

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

```
n[0] = 5;
```

```
n[1] = 4;
```

```
n[2] = 3;
```

```
n[3] = 2;
```

```
n[4] = 1;
```

# 配列の大きさ

- Lengthは、配列変数の成分の数を表す  
配列の長さ

(配列変数名).Length

# 多次元配列

- **多次元配列**…複数の添字がついた配列変数
- **二次元配列**はよく使われる
  - 表
  - 座標などのデータ
- 表計算の処理は、二次元配列を用い実現できる
- 二次元、三次元…といった多次元の配列変数を作ることとは可能だが実際用いられる多次元配列は、ほとんどが二次元配列
- **Sample503**参照

# 二次元配列のイメージ

Sample503

```
int[,] a= new int[3,4]
```

	0	1	2	3
0	a[0,0]	a[0,1]	a[0,2]	a[0,3]
1	a[1,0]	a[1,1]	a[1,2]	a[1,3]
2	a[2,0]	a[2,1]	a[2,2]	a[2,3]

# 多次元配列の宣言とアクセス

- 二次元配列 **int[,] a = new int[3,4]; → a[1,2] = 2;**
- 三次元配列 **int[,,] a = new int[3,4,5]; → a[1,2,3] = 2;**
- 四次元配列 **int[,,,] a = new int[3,4,5,6]; → a[1,2,3,4] = 2;**



# ジャグ配列

Sample504

	0	1	2	3
0	a[0][0]	a[0][1]		
1	a[1][0]			
2	a[2][0]	a[2][1]	a[2][2]	a[2][3]

```
int[][] a = new int[][]  
{new int[]{0,1},  
new int[]{2},  
new int[]{3,4,5,6}};
```

# ジャグ配列の成分の長さ

Sample504

