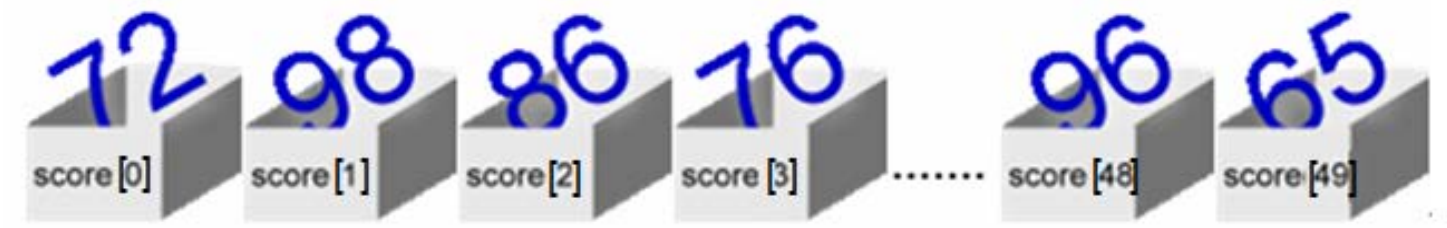
Array and Method

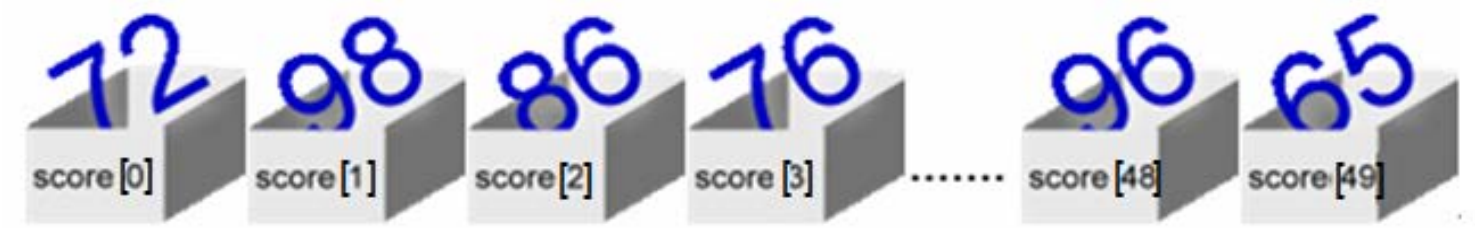
4-1 Array

- Array object can be simply called array
- Store data – like variable
- One variable only stores one data
- Array object stores many data in a series of memory area
- Array object is a set of data which has the same data type。



4-1 Array

- Combined with array elements
- Use array_name[index] to stand for an array element
- An array element is as same as a variable
- Use [] to identify every element
- Assign index, then C# automatically figures out the real position



4-2 1-D Array

- Declaration is required before usage
- Target:
  1. Assign array’s name and data type
  2. Determine memory space used by array
  3. Store number of data

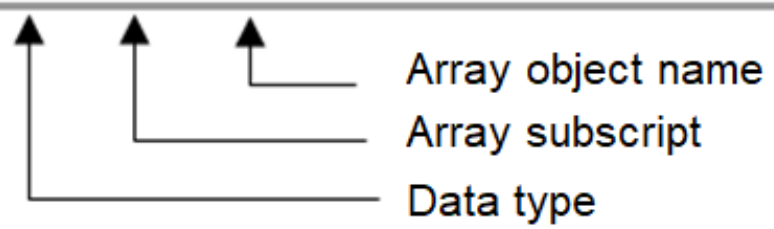
4-1.1. Array Declaration

- **1 [] ⇨ 1-D array**
- **2 [,] ⇨ 2-D array**

1. Declaration of 1-D array

Grammar

```
type[] arrayname;
```



Ex: declare myAry as an 1-D array

```
int[] myAry;
```

2. Create an 1-D array object

Grammar
`arrayname = new type[size];` // size is array's length

Ex: create an array object called myAry, this array object contains myAry[0] ~ myAry[4] five array elements:

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

3. C# allows combining declaration and realization into one statement, usage:

Grammar
type[] arrayname = new type[n];

Ex: combine declaration and realization of myAry into a single statement:

int[] myAry;

myAry = new int[5];

} Combine: **int[] myAry = new int[5];**

4-1.2. Initial Values of Array

- Every array element is a variable
- A declared array can use assign operator (=) to assign initial values to array elements
- If no initial value during declaration:
  ① number types: initial value is 0
  ② string types: initial value is empty string

2 ways to set initial values during declaration:

1. Use equal sign to assign initial values of array elements

```
int[] tAry = new int[5];
tAry[0] = 10;   tAry[1] = 20;   tAry[2] =30;   tAry[3] = 40;   tAry[4] = 50;
```

2. Use brackets to assign initial values, separated by comma

```
int []tAry = new int[ ] {10, 20, 30, 40, 50};
```

Ignore array length when assigning initial values

Array Declaration, Realization and Initial Value Assignment

Grammar

1. Declaration

   ```
   type[] arrayname;
   ```

2. Realization

   ```
   arrayname = new type[size];
   ```

3. Combine declaration and realization in one statement

   ```
   type[] arrayname = new type[n];
   ```

4. Combine declaration, realization and initial value in one statement

   ```
   type[] arrayname = new type[] { initiationlist };
   ```

4-1.3. Length Property of Array

Grammar
int varName = arrayName.Length;

【例】① long[] score = new long[4]; ⇦ Declare and realize score long array

int size = score.Length; ⇦ size =4

② double[] price=new double[] {1.2, 3.0, -0.0000005, 55, 1.6E14 };

int size = price.Length; ⇦ size = 5

③ string[] wk = {“Sun”, “Mon”, “Tue”, “Wed” , “Thu ”, “Fri” , “Sat”};

int size = wk.Length; ⇦ size = 7

4-2. Use for Loop

- **Read contents of tAry**

```
// FileName : ary1.sln
    int [] tAry = new int[] { 10, 20, 30, 40, 50 };
    for (int k = 0; k <= tAry.Length-1; k++)
            Console.Write (" tAry[{0}]={1}    ", k , tAry[k]);
    Console.Read();
```

Result:

tAry[0]=10 tAry[1]=20 tAry[2]=30
tAry[3]=40   tAry[4]=50

- **Exchange contents of tAry[1]=20 and tAry[3]=40**
- **Usage:**

```
temp= tAry[1]  ;
 tAry[1]= tAry[3];
 tAry[3] =temp;
```

4-2.2. Use foreach Loop

- **Combined with array name, array data type and the same variable name:**
  ⇨ read array elements one by one
  ⇨ avoid the consideration of index

Grammar

```
foreach (type element in group) {
    [statements]
    [break;]
    [statements]
};
```

Ex: display the values of array elements in the string array continuously on screen

```
string[] msg = new string[4]{"Time", "is", "money",  "."};
foreach (string word in msg)
    Console.Write("Δ{0} ", word);   // Δ : a space
Console.Read();
```

Result: "Time is money ."

Example (readAry1):

Write a program to store names and serial numbers, as shown in the list. As the serial number is inputted, the related name is shown. If the inputted number is out of range, show "… This serial number is unavailable …".

S/N	Name	Array element
1	AA	name[0]
2	BB	name[1]
3	CC	name[2]
4	DD	name[3]
5	EE	name[4]

Hint: the subscript's difference between serial number and array element is 1

Result:

```
1. Input seat number (1-5) : 3
2. Student name : CC
```

```
1. Input seat number (1-5) : 8
... This is an empty number ...
```

FileName : readAry1.sln

```
05  static void Main(string[] args)
06  {
07          string[] name = new string[5];
09           name[0]=“AA";  name[1]=“BB"; name[2]=“CC";
10           name[3]=“DD";  name[4]=“EE";
11           int no;
12           Console.Write(" 1. Input seat number (1-5)： ");
13           no = int.Parse(Console.ReadLine());
14           if(no>=1 && no<=5)
15               Console.WriteLine(" 2. Student name : {0} ", name[no-1]);
16           else
17               Console.WriteLine(" ... This is an empty number ... ");
18           Console.Read();
19   }
```

The End