



دانشگاه جامع علمی کاربردی

به نام خدا

دانشگاه جامع علمی کاربردی

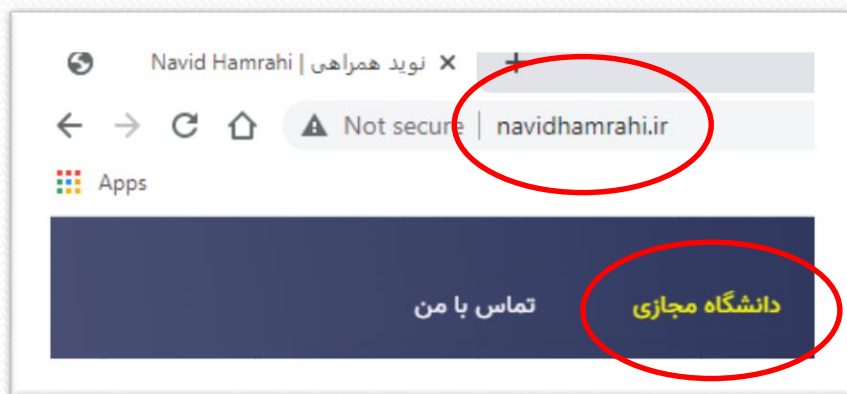
# راهنمای مقدماتی برنامه نویسی در C#

مدرس : نوید همراهی

[www.navidhamrahi.ir](http://www.navidhamrahi.ir)

[n.hamrahi@gmail.com](mailto:n.hamrahi@gmail.com)

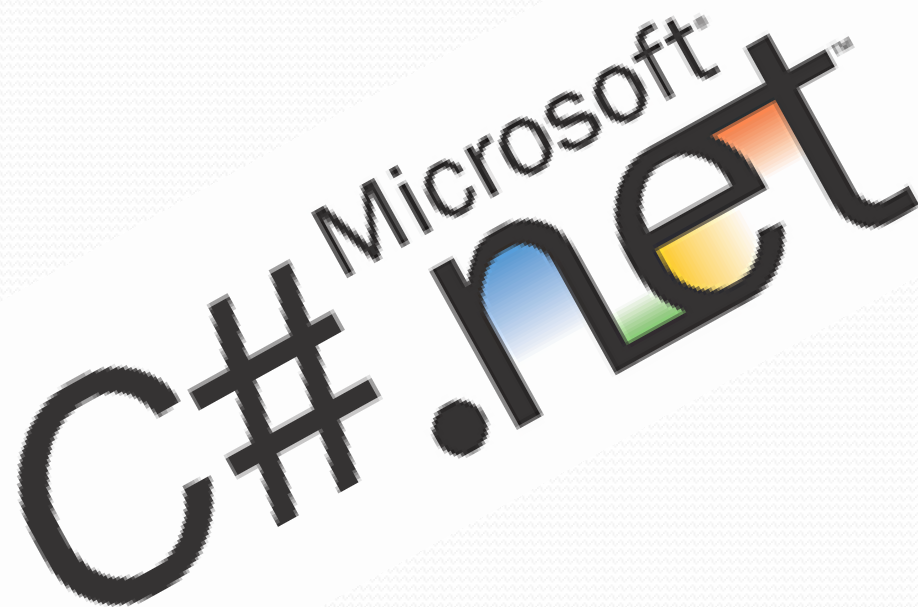
# نحوه دریافت جزوه و آگاهی از عملکرد و روند کلاس



۱- ورود به وب سایت : [www.navidhamrahi.ir](http://www.navidhamrahi.ir)

۲- ورود به بخش : دانشگاه مجازی

# زبان برنامه نویسی سی شارپ :

The logo for Microsoft C#.net is displayed on a white rectangular background that is tilted. The word "Microsoft" is written in a black, sans-serif font at the top. Below it, "C#" is written in a large, bold, black font. To the right of "C#", ".net" is written in a large, bold, black font. The ".net" part features a colorful gradient: the dot is blue, the "n" is green, the "e" is yellow, and the "t" is orange. A small trademark symbol (TM) is located at the top right of the "t".

Microsoft  
C#.net™

# انواع کامپایلر :

## Command Line Compiler

- Microsoft C# Command Line Compiler

## Visual Compiler

- Microsoft Visual C#



## چند نکته راجع به C# :

### ❑ Case Sensitive ?

❑ **ReadLine** , **readline** , **Readline** , **ReAdliNe** , ...

### ❑ Description : // OR /\* \*/

❑ // توضیحات

❑ /\* توضیحات \*/

## مثال ۱ – بررسی یک برنامه ساده و تعریف چند اصطلاح :

```
// Namespace اعلان  
using System;  
  
// کلاس آغازین برنامه  
class WelcomeCSS  
{  
    // آغاز کار اجرای برنامه  
    public static void Main()  
    {  
        // نوشتن متن در خروجی  
        Console.WriteLine("Welcome to the C# Persian Tutorial!");  
    }  
}
```

## مثال ۱ – بررسی یک برنامه ساده و تعریف چند اصطلاح :

```
// Namespace اعلان  
using System;
```

```
// کلاس آغازین برنامه
```

```
class WelcomeCSS
```

```
{
```

```
// آغاز کار اجرای برنامه
```

```
public static void Main()
```

```
{
```

```
// نوشتن متن در خروجی
```

```
Console.WriteLine("Welcome to the C# Persian Tutorial!");
```

```
}
```

```
}
```

## مثال ۱ – بررسی یک برنامه ساده و تعریف چند اصطلاح :

```
// Namespace اعلان
```

```
using System;
```

```
// کلاس آغازین برنامه
```

```
class WelcomeCSS
```

```
{
```

```
// آغاز کار اجرای برنامه
```

```
public static void Main()
```

```
{
```

```
// نوشتن متن در خروجی
```

```
Console.WriteLine("Welcome to the C# Persian Tutorial!");
```

```
}
```

```
}
```

## مثال ۱ – بررسی یک برنامه ساده و تعریف چند اصطلاح :

```
// Namespace اعلان  
using System;  
  
// کلاس آغازین برنامه  
class WelcomeCSS  
{  
    // آغاز کار اجرای برنامه  
    public static void Main()  
    {  
        // نوشتن متن در خروجی  
        Console.WriteLine("Welcome to the C# Persian Tutorial!");  
    }  
}
```

## مثال ۱ – بررسی یک برنامه ساده و تعریف چند اصطلاح :

```
// Namespace اعلان  
using System;  
  
// کلاس آغازین برنامه  
class WelcomeCSS  
{  
    // آغاز کار اجرای برنامه  
    public static void Main()  
    {  
        // نوشتن متن در خروجی  
        Console.WriteLine("Welcome to the C# Persian Tutorial!");  
    }  
}
```

## یادآوری // و /\* \*/ :

```
// Namespace اعلان
```

```
using System;
```

```
// کلاس آغازین برنامه
```

```
class WelcomeCSS
```

```
{
```

```
// آغاز کار اجرای برنامه
```

```
public static void Main()
```

```
{
```

```
// نوشتن متن در خروجی
```

```
Console.WriteLine("Welcome to the C# Persian Tutorial!");
```

```
}
```

```
}
```

## یادآوری بعضی از دستورات :

- ❑ **Console.Write (...)**
- ❑ **Console.WriteLine (...)**
- ❑ **Console.ReadLine (...)**



## مثال ۲ – بررسی یک برنامه کاربردی :

```
// Namespace اعلان
using System;
// کلاس آغازین برنامه
class NamedWelcome
{
    // آغاز اجرا برنامه
    public static void Main(string[] args)
    {
        // نمایش بر روی صفحه
        Console.WriteLine("Hello, {0}!", args[0]);
        Console.WriteLine("Welcome to the C# Persian Tutorial!");
    }
}
```

## مثال ۲ – بررسی یک برنامه کاربردی :

```
// Namespace اعلان
```

```
using System;
```

```
// کلاس آغازین برنامه
```

```
class NamedWelcome
```

```
{
```

```
// آغاز اجرا برنامه
```

```
public static void Main(string[] args)
```

```
{
```

```
// نمایش بر روی صفحه
```

```
Console.WriteLine("Hello, {0}!", args[0]);
```

```
Console.WriteLine("Welcome to the C# Persian Tutorial!");
```

```
}
```

```
}
```

## مثال ۲ – بررسی یک برنامه کاربردی :

```
// Namespace اعلان
```

```
using System;
```

```
// کلاس، آغازین برنامه
```

```
class NamedWelcome
```

```
{
```

```
// آغاز اجرا برنامه
```

```
public static void Main(string[] args)
```

```
{
```

```
// نمایش بر روی صفحه
```

```
Console.WriteLine("Hello, {0}!", args[0]);
```

```
Console.WriteLine("Welcome to the C# Persian Tutorial!");
```

```
}
```

```
}
```

## مثال ۲ – بررسی یک برنامه کاربردی :

```
// Namespace اعلان  
using System;  
// کلاس آغازین برنامه  
class NamedWelcome  
{  
    // آغاز اجرا برنامه  
    public static void Main(string[] args)  
    {  
        // نمایش بر روی صفحه  
        Console.WriteLine("Hello, {0}!", args[0]);  
        Console.WriteLine("Welcome to the C# Persian Tutorial!");  
    }  
}
```

## مثال ۲ – بررسی یک برنامه کاربردی :

```
// Namespace اعلان
using System;
// کلاس آغازین برنامه
class NamedWelcome
{
    // آغاز اجرا برنامه
    public static void Main(string[] args)
    {
        // نمایش بر روی صفحه
        Console.WriteLine("Hello, {0}!", args[0]);
        Console.WriteLine(" welcome to the C# Persian Tutorial! ");
    }
}
```

## مثال ۲ – بررسی یک برنامه کاربردی :

```
// Namespace اعلان
using System;
// کلاس آغازین برنامه
class NamedWelcome
{
    // آغاز اجرا برنامه
    public static void Main(string[] args)
    {
        // نمایش بر روی صفحه
        Console.WriteLine("Hello, {0}!", args[0]);
        Console.WriteLine("Welcome to the C# Persian Tutorial!");
    }
}
```

## مثال ۳ – بررسی یک برنامه که قابلیت محاوره با کاربر را دارد :

```
// Namespace اعلان
using System;

// کلاس آغازین برنامه
class InteractiveWelcome
{
    // آغاز اجرای برنامه
    public static void Main()
    {
        // متنی بر روی صفحه نمایش داده می‌شود
        Console.Write("What is your name?: ");
        // متنی نمایش داده شده و برنامه منتظر دریافت ورودی می‌ماند
        Console.Write("Hello, {0}! ", Console.ReadLine());
        Console.WriteLine("Welcome to the C# Persian Tutorial!");
    }
}
```

## مثال ۳ - خروجی برنامه :

**What Is Your Name ? :**

**Meysam / Enter**

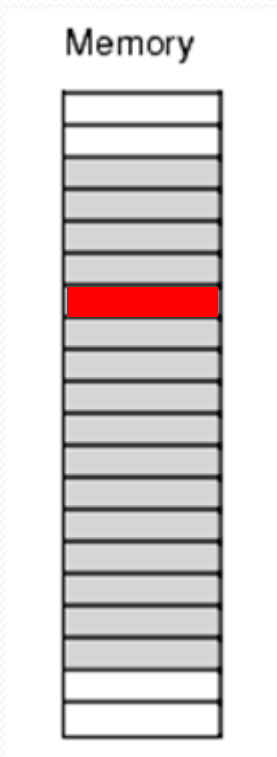
**Hello, Meysam**

**Welcome To The C# Persian Tutorial !**



## متغیر چیست ؟

مکانی از حافظه است که می تواند مقداری را در خود نگه داری نماید



### انواع متغیر :

- متغیر عددی صحیح
- متغیر عددی اعشاری
- متغیر کارکتری و رشته ای
- متغیر منطقی

## انواع متغیر عددی صحیح :

نوع	اندازه به بیت	رنج قابل قبول
<b>sbyte</b>	۸	۱۲۷ تا -۱۲۸
<b>byte</b>	۸	۲۵۵ تا ۰
<b>short</b>	۱۶	۳۲۷۶۷ تا -۳۲۷۶۸
<b>ushort</b>	۱۶	۶۵۵۳۵ تا ۰
<b>int</b>	۳۲	۲۱۴۷۴۸۳۶۴۷ تا -۲۱۴۷۴۸۳۶۴۸
<b>uint</b>	۳۲	۴۲۹۴۹۶۷۲۹۵ تا ۰
<b>long</b>	۶۴	۹۲۲۳۳۷۲۰۳۶۸۵۴۷۷۵۸۰۷ تا -۹۲۲۳۳۷۲۰۳۶۸۵۴۷۷۵۸۰۸
<b>ulong</b>	۶۴	۱۸۴۴۶۷۴۴۰۷۳۷۰۹۵۵۱۶۱۵ تا ۰

## انواع متغیر عددی اعشاری :

نوع	اندازه به بیت	دقت	رنج قابل قبول
<b>float</b>	۳۲	۷ رقم	$1.5 \times 10^{-45}$ تا $3.4 \times 10^{38}$
<b>double</b>	۶۴	۱۵-۱۶ رقم	$5.0 \times 10^{-324}$ تا $1.7 \times 10^{308}$
<b>decimal</b>	۱۲۸	۲۸-۲۹ رقم دسیمال	$1.0 \times 10^{-28}$ تا $7.9 \times 10^{28}$

# انواع متغیر عددی کارکتری و رشته ای :

**Char** کارکتری عبارت است از :

**String** رشته ای عبارت است از :

**متغیر منطقی :**

**Boolean** منطقی عبارت است از :

## انواع عملگر ها :

نوع عمل	عملگر(ها)	شرکت پذیری
عملیات ابتدایی	(x) x.y f(x) a[x] x++ x-- new typeof sizeof checked unchecked	از چپ
عملیات یکانی	+ - ! ~ ++x --x (T)x	از چپ
عملیات ضربی	* / %	از چپ
عملیات جمعی	+ -	از چپ
عمل شیفت	<< >>	از چپ
عملیات رابطه‌ای	< > <= >= is	از چپ
عملیات تساوی	== !=	از راست
عمل AND منطقی	&	از چپ
عمل OR منطقی		از چپ
عمل XOR منطقی	^	از چپ
عمل AND شرطی	&&	از چپ
عمل OR شرطی		از چپ
عمل شرطی	?:	از چپ
عمل انتساب	= *= /= %= += -= <<= >>= &= ^=  =	از راست

## عملگرهای یکانی (Unary) :

نوع عمل	عملگر(ها)	شرکت پذیری
عملیات ابتدایی	(x) x.y f(x) a[x] x++ x-- new typeof sizeof checked unchecked	از چپ
عملیات یکانی	+ - ! ~ ++x --x (T)x	از چپ
عملیات ضربی	* / %	از چپ
عملیات جمعی	+ -	از چپ
عمل شیفت	<< >>	از چپ
عملیات رابطه‌ای	< > <= >= is	از چپ
عملیات تساوی	== !=	از راست
عمل AND منطقی	&	از چپ
عمل OR منطقی		از چپ
عمل XOR منطقی	^	از چپ
عمل AND شرطی	&&	از چپ
عمل OR شرطی		از چپ
عمل شرطی	?:	از چپ
عمل انتساب	= *= /= %= += -= <<= >>= &= ^=  =	از راست

## عملگر های دو تایی :

نوع عمل	عملگر(ها)	شرکت پذیری
عملیات ابتدایی	(x) x.y f(x) a[x] x++ x-- new typeof sizeof checked unchecked	از چپ
عملیات یکانی	+ - ! ~ ++x --x (T)x	از چپ
عملیات ضربی	* / %	از چپ
عملیات جمعی	+ -	از چپ
عمل شیفت	<< >>	از چپ
عملیات رابطه‌ای	< > <= >= is	از چپ
عملیات تساوی	== !=	از راست
عمل AND منطقی	&	از چپ
عمل OR منطقی		از چپ
عمل XOR منطقی	^	از چپ
عمل AND شرطی	&&	از چپ
عمل OR شرطی		از چپ
عمل شرطی	?:	از چپ
عمل انتساب	= *= /= %= += -= <<= >>= &= ^=  =	از راست

## تبدیل نوع های داده ای :

❖ (int)

❖ (long)

❖ (float)

❖ ( و ... )

مثال :

```
Int A=2;
```

```
Int B=10;
```

```
Float C=0;
```

```
C = (float)A * (float)B ;
```

خروجی :

```
C = 2.00 * 10.00 = 20.00
```



## مثال ۴ :

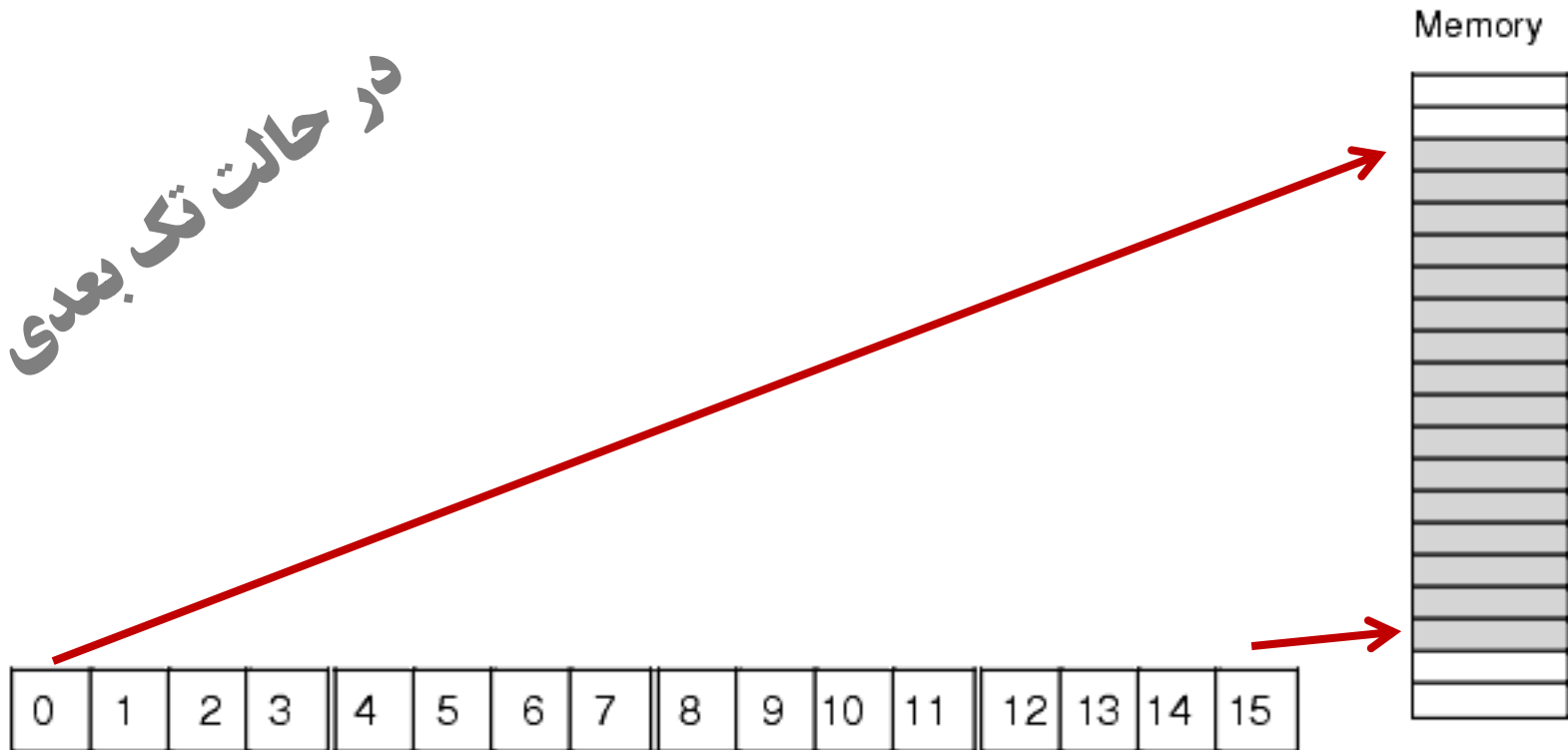
```
using System;
class Binary
{
    public static void Main()
    {
        int x, y, result;
        float floatResult;
        x = 7;
        y = 5;
        result = x+y;
        Console.WriteLine("x+y: {0}", result);
        result = x-y;
        Console.WriteLine("x-y: {0}", result);
        result = x*y;
        Console.WriteLine("x*y: {0}", result);
        result = x/y;
        Console.WriteLine("x/y: {0}", result);
        floatResult = (float)x/(float)y;
        Console.WriteLine("x/y: {0}", floatResult);
        result = x%y;
        Console.WriteLine("x%y: {0}", result);
        result += x;
        Console.WriteLine("result+=x: {0}", result);
    }
}
```

x+y: 12  
x-y: 2  
x\*y: 35  
x/y: 1  
x/y: 1.4  
x%y: 2  
result+=x: 9

خروجی برنامه :

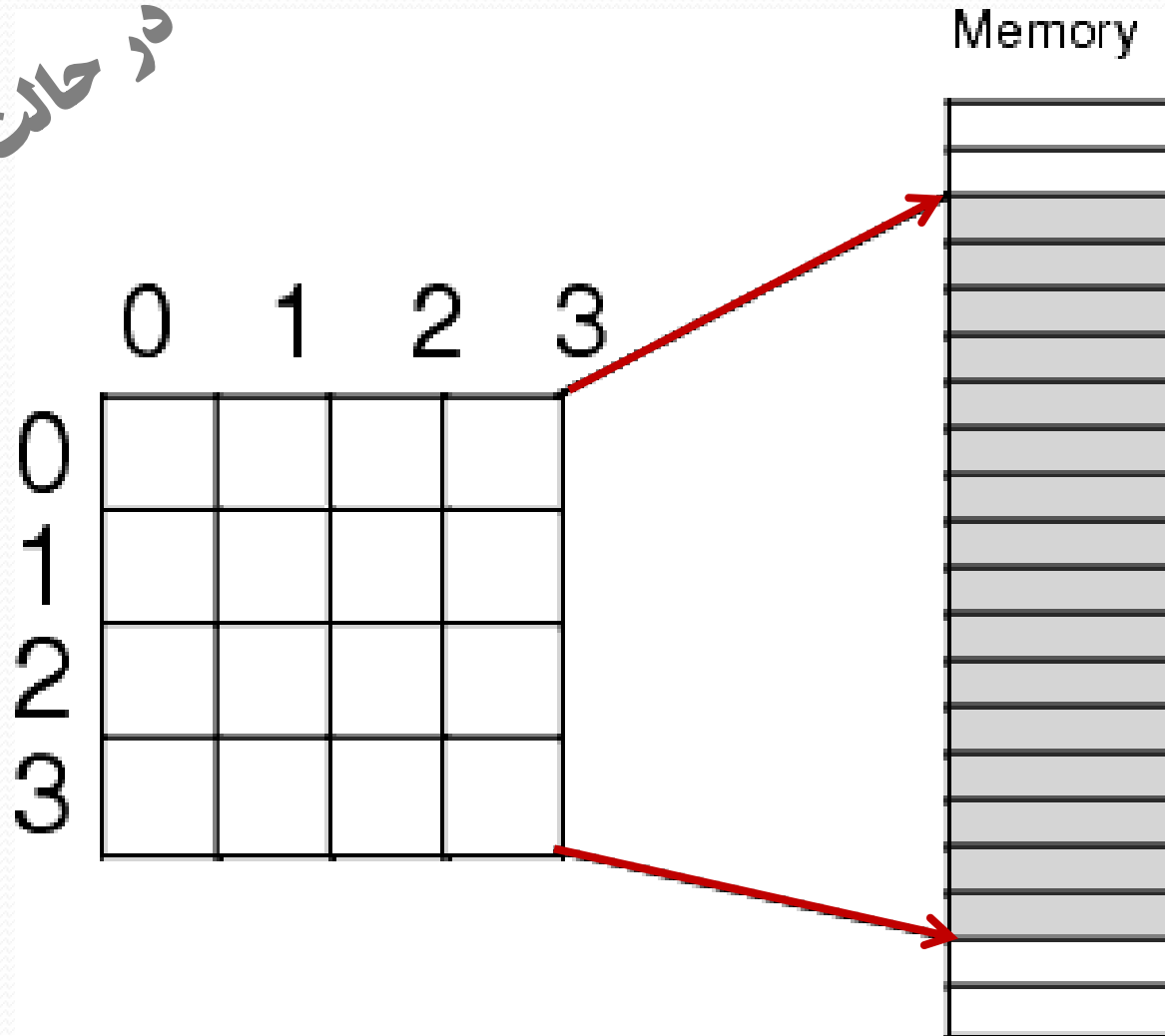
# آرایه چیست؟

در حالت تک بعدی



# آرایه چیست؟

در حالت دو بعدی



# آرایه تک بعدی چیست؟

♦ اندیس



$X[4] = 10 ;$



اندیس  $n$

# نحوه تعریف آرایه یک بعدی :

## 1 - Type [ ] ArrayName ;

Sample -> int[10] temp;

Sample -> int[]temp = {11,13,15,1,2};

## 2 - Type [ ] ArrayName = new Type [ ]

Sample -> int[]temp = new int[10]

# آرایه دو بعدی چیست ؟

اندریس  $j$  و  $i$

	0	1	2
0			
1			
2	10		

$X[2,0] = 10 ;$

اندریس  $j$  و  $i$

## نحوه تعریف آرایه دو بعدی :

### 1 - Type [ , ] ArrayName ;

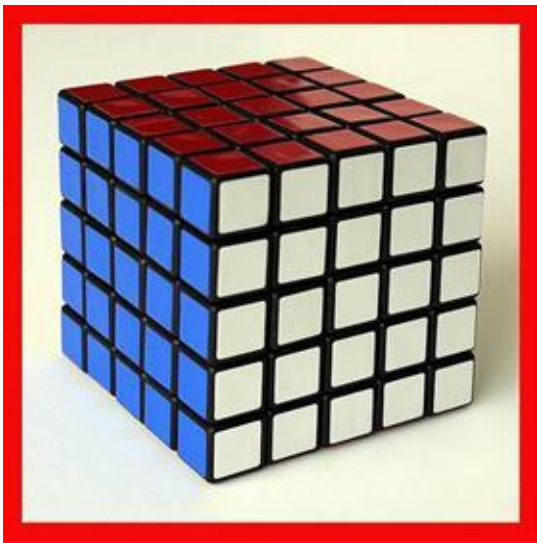
Sample -> int[10,20] temp;

Sample -> int[ 2,1 ]temp = {45,17};

### 2 - Type [ , ] ArrayName = new Type [ , ]

Sample -> int[ , ]temp = new int[65,98]

## آرایه n بعدی :



سه بعدی

Sample -> `int[19,19,19] temp;`

**Type [a,b,c,...] Name;**



```

using System;
class Array
{
    public static void Main()
    {
        int[] myInts = { 5, 10, 15 };
        bool[][] myBools = new bool[2][];
        myBools[0] = new bool[2];
        myBools[1] = new bool[1];
        double[,] myDoubles = new double[2, 2];
        string[] myStrings = new string[3];
        Console.WriteLine("myInts[0]: {0}, myInts[1]: {1}, myInts[2]: {2}", myInts[0],
            myInts[1], myInts[2]);
        myBools[0][0] = true;
        myBools[0][1] = false;
        myBools[1][0] = true;
        Console.WriteLine("myBools[0][0]: {0}, myBools[1][0]: {1}", myBools[0][0],
            myBools[1][0]);
        myDoubles[0, 0] = 3.147;
        myDoubles[0, 1] = 7.157;
        myDoubles[1, 1] = 2.117;
        myDoubles[1, 0] = 56.00138917;
        Console.WriteLine("myDoubles[0, 0]: {0}, myDoubles[1, 0]: {1}", myDoubles[0,
            0], myDoubles[1, 0]);
        myStrings[0] = "Joe";
        myStrings[1] = "Matt";
        myStrings[2] = "Robert";
        Console.WriteLine("myStrings[0]: {0}, myStrings[1]: {1}, myStrings[2]: {2}",
            myStrings[0], myStrings[1], myStrings[2]);
    }
}

```

myInts[0]: 5, myInts[1]: 10, myInts[2]: 15  
myBools[0][0]: True, myBools[1][0]: True  
myDoubles[0, 0]: 3.147, myDoubles[1, 0]: 56.00138917  
myStrings[0]: Joe, myStrings[1]: Matt, myStrings[2]: Robert

## مثال ۵ :

## خروجی برنامه :

## بررسی دستور if :

جهت قسمت های تصمیم گیری در برنامه می بایست از این دستور استفاده نمود.

## نحوه نوشتن دستور :

```
if ( شرط )  
{  
    دستور یا دستورات  
}
```

## بررسی دستور if - else :

If ( شرط )

```
{  
    دستور یا دستورات;  
}
```

else

```
{  
    دستور یا دستورات;  
}
```

## بررسی دستور if های تو در تو :

If ( شرط ۱ )

{ دستور یا دستورات }

else If ( شرط ۲ )

{ دستور یا دستورات }

else If ( شرط ۳ )

{ دستور یا دستورات }

...

else

{ دستور یا دستورات }

```
using System;
```

```
class IfSelect
```

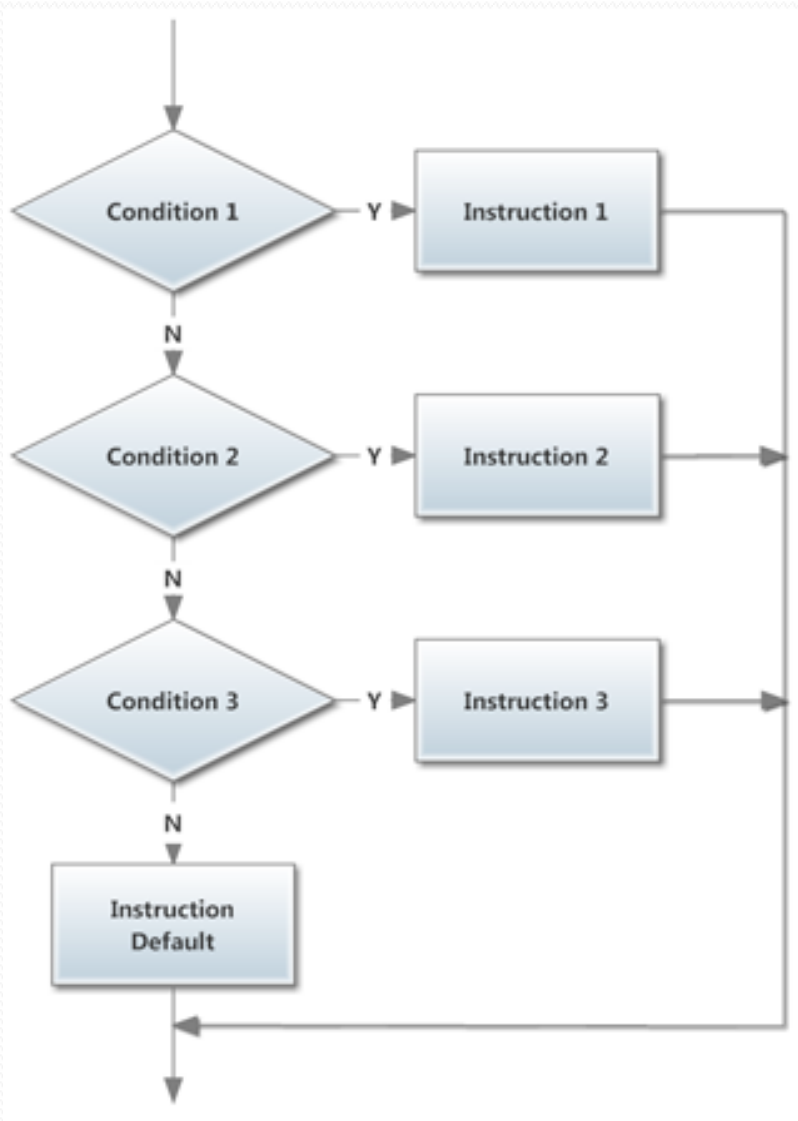
```
{  
    public static void Main()  
    {  
        string myInput;  
        int myInt;  
        Console.WriteLine("Please enter a number: ");  
        myInput = Console.ReadLine();  
        myInt = Int32.Parse(myInput);  
        // تصمیم‌گیری ساده و اجرای عمل داخل دو گروه  
        if (myInt > 0)  
        {  
            Console.WriteLine("Your number {0} is greater than zero.", myInt);  
        }  
        // تصمیم‌گیری ساده و اجرای عمل بدون استفاده از دو گروه  
        if (myInt < 0)  
            Console.WriteLine("Your number {0} is less than zero.", myInt);  
        // تصمیم‌گیری دوگانه  
        if (myInt != 0)  
        {  
            Console.WriteLine("Your number {0} is not equal to zero.", myInt);  
        }  
        else  
        {  
            Console.WriteLine("Your number {0} is equal to zero.", myInt);  
        }  
        // تصمیم‌گیری چندگانه  
        if (myInt < 0 || myInt == 0)  
        {  
            Console.WriteLine("Your number {0} is less than or equal to zero.", myInt);  
        }  
        else if (myInt > 0 && myInt <= 10)  
        {  
            Console.WriteLine("Your number {0} is between 1 and 10.", myInt);  
        }  
        else if (myInt > 10 && myInt <= 20)  
        {  
            Console.WriteLine("Your number {0} is between 11 and 20.", myInt);  
        }  
        else if (myInt > 20 && myInt <= 30)  
        {  
            Console.WriteLine("Your number {0} is between 21 and 30.", myInt);  
        }  
        else  
        {  
            Console.WriteLine("Your number {0} is greater than 30.", myInt);  
        }  
    }  
} //Main() پایان متد  
} //IfSelect پایان کلاس
```

## مثال ۶ دستور if :

## بررسی دستور switch :

جهت تصمیم گیری چندین شرط در یک برنامه می بایست از این دستور استفاده نمود .

**نکته :** از if های تو در تو نیز می توان استفاده کرد ولی به نظر شما چرا این کار را انجام نمی دهند ؟



**switch** ( متغیری که قرار است بررسی شود )

{

**case** مقدار مورد نظر :

دستور یا دستورات

break;

**case** مقدار مورد نظر :

دستور یا دستورات

break;

**default:**

دستور یا دستورات

}

نحوه نوشتن دستور :

```
using System;
```

```
class SwitchSelect
```

```
{  
    public static void Main()  
    {  
        string myInput;  
        int myInt;  
        begin:  
        Console.WriteLine("Please enter a number between 1 and 3: ");  
        myInput = Console.ReadLine();  
        myInt = Int32.Parse(myInput);  
        // دستور switch به همراه متغیری از نوع صحیح  
        switch (myInt)  
        {  
            case 1:  
                Console.WriteLine("Your number is {0}.", myInt);  
                break;  
            case 2:  
                Console.WriteLine("Your number is {0}.", myInt);  
                break;  
            case 3:  
                Console.WriteLine("Your number is {0}.", myInt);  
                break;  
            default:  
                Console.WriteLine("Your number {0} is not between 1 and 3.", myInt);  
                break;  
        } //switch پایان بلوک  
        decide:  
        Console.WriteLine("Type \"continue\" to go on or \"quit\" to stop: ");  
        myInput = Console.ReadLine();  
        // دستور switch به همراه متغیری از نوع رشته‌ای  
        switch (myInput)  
        {  
            case "continue":  
                goto begin;  
            case "quit":  
                Console.WriteLine("Bye.");  
                break;  
            default:  
                Console.WriteLine("Your input {0} is incorrect.", myInput);  
                goto decide;  
        } //switch پایان بلوک  
    } //Main() پایان متد  
} //SwitchSelect پایان کلاس
```

## مثال ۷ دستور switch :



## بررسی حلقه های تکرار :

جهت انجام دادن کارهای تکراری از حلقه ها استفاده می نمایم .

### انواع حلقه ها :

حلقه while

حلقه do-while

حلقه for

حلقه foreach

## مثال ۸ حلقه while :

```
using System;

class WhileLoop
{
    public static void Main()
    {
        int myInt = 0;
        while (myInt < 10)
        {
            Console.WriteLine("{0} ", myInt);
            myInt++;
        }
        Console.WriteLine();
    }
}
```

using System;

class DoLoop

```
{  
    public static void Main()  
    {  
        string myChoice;  
        do  
        {  
            // منوی نمایش داده می شود  
            Console.WriteLine("My Address Book\n");  
            Console.WriteLine("A - Add New Address");  
            Console.WriteLine("D - Delete Address");  
            Console.WriteLine("M - Modify Address");  
            Console.WriteLine("V - View Addresses");  
            Console.WriteLine("Q - Quit\n");  
            Console.WriteLine("Choice (A,D,M,V,or Q): ");  
            // ورودی کاربر بررسی می شود  
            myChoice = Console.ReadLine();  
            // تصمیمی بر اساس ورودی کاربر گرفته می شود  
            switch(myChoice)  
            {  
                case "A":  
                case "a":  
                    Console.WriteLine("You wish to add an address.");  
                    break;  
                case "D":  
                case "d":  
                    Console.WriteLine("You wish to delete an address.");  
                    break;  
                case "M":  
                case "m":  
                    Console.WriteLine("You wish to modify an address.");  
                    break;  
                case "V":  
                case "v":  
                    Console.WriteLine("You wish to view the address list.");  
                    break;  
                case "Q":  
                case "q":  
                    Console.WriteLine("Bye.");  
                    break;  
                default:  
                    Console.WriteLine("{0} is not a valid choice", myChoice);  
                    break;  
            }  
            Console.WriteLine("Press Enter key to continue...");  
            Console.ReadLine();  
            Console.WriteLine();  
        } while (myChoice != "Q" && myChoice != "q");  
    }  
}
```

## مثال ۹ حلقه do-while:

## مثال ۱۰ حلقه for :

```
using System;

class ForLoop
{
    public static void Main()
    {
        for (int i=0; i < 20; i++)
        {
            if (i == 10)
                break;
            if (i % 2 == 0)
                continue;
            Console.Write("{0} ", i);
        }
        Console.WriteLine();
    }
}
```

## مثال ۱۱ حلقه foreach :

```
using System;

class ForEachLoop
{
    public static void Main()
    {
        string[] names = {"Meysam", "Ghazvini", "C#", "Persian"};
        foreach (string person in names)
        {
            Console.WriteLine("{0} ", person);
        }
    }
}
```

## بررسی دو نکته :

۱ - دستور **continue** : دستوری می باشد جهت انجام ادامه حلقه .

۲ - دستور **break** : دستوری می باشد جهت ادامه ندادن حلقه و خارج شدن از آن .

## نکته :

در این درس با نحوه کار با دستورات عمل‌های ***for***، ***while***، ***do-while*** و ***foreach*** آشنا شدید. حلقه ***while*** تا زمانی که شرطش صحیح باشد انجام می‌شود بدین معنی که تا زمانی که عبارت منطقی آن دارای مقدار ***true*** باشد، دستورات درون بلوک آن اجرا می‌شوند. حلقه ***do***، دستورات بلوک خود را حداقل یکبار اجرا می‌کند و سپس شرط مورد نظر را بررسی می‌نماید و در صورتیکه عبارت منطقی آن مقدار ***true*** بازگرداند، دستورات بلوک خود را تکرار می‌نماید. حلقه ***for*** دستورات بلوک خود را به تعداد دفعات مشخص اجرا می‌نماید و حلقه ***foreach*** عناصر یک مجموعه را مورد پیمایش قرار می‌دهد. در نهایت نیز اشاره می‌شود که روند اجرای حلقه‌ها با استفاده از دستورات ***break*** و ***continue*** تغییر می‌نماید.

**کلیه حقوق این جزوه متعلق به**

**مهندس نوید همراهی می باشد**

**و استفاده از مطالب آن با ذکر منبع بلامانع است.**

**[www.navidhamrahi.ir](http://www.navidhamrahi.ir)**

**پایان**