



آموزش برنامه نویسی سی شارپ تحت کنسول

www.dazhyarco.net

آموزش سی شارپ تحت کنسول

• درباره C#

زبان سی شارپ یک زبان شی گرا و سطح بالا هستش

زبان سی شارپ یک زبان چند الگویی منظم شده مدل های تابعی، امری، عمومی، شی گرا و جز گرا در بستر دات نت می باشد.

تصویری از تیم طراحی سی شارپ:



• ساختار کلی برنامه درسی شارپ

USING نام توابع

CLASS نام فایل

```
{
    MAIN ()
    {
        ; تعریف متغیرها
        ; دستورات
    }
}
```

• انواع داده درسی شارپ:

نوع داده	توضیحات
String	شامل دنباله ای از کاراکترها می شود
Bool	داده ای منطقی که می تواند مقدار True یا False داشته باشد
Char	کاراکتر که شامل اعداد ، حروف
INT	شامل کلیه اعداد ۳۲ بیتی
Long	شامل کلیه اعداد ۶۴ بیتی
Float	اعداد صحیح و اعشاری
Double	اعداد صحیح و اعشاری که طولش از Float بیشتر است

متغیر: مکانی از حافظه که داده ها در آن قرار می گیرد.

نام هر خانه حافظه متغیر می تواند تغییر کند ولی آدرس آن همواره ثابت است.

• نحوه تعریف متغیر

نام متغیر نوع متغیر

Int x;

Double y

نکته: در نام گذاری متغیر ها به موارد زیر باید دقت کرد:

۱- متغیر هم نام نداریم

۲- در انتخاب نام متغیر نمی شود از اسامی معنی دار استفاده کرد

۳- در برنامه نویسی به زبان سی شارپ باید توجه کرد که این زبان به حروف حساس است کلمات کلیدی و دستورات بایستی با حروف کوچک نوشته شود

۴ - نام متغیر باید با یکی از حروف A تا Z و _ (آندرلاین) شروع شود.

۵- نکته: برای تعریف متغیرهایی که از یک نوع می باشند می توان از (کاما) استفاده کرد

• نوشتن دستور

برای نوشتن دستورها در محیط کنسول تمام دستورها باید داخل {} باشد
برای اینکه یک کارکتر عینا چاپ شود باید داخل (" ") باشد

• خروج داده

برای چاپ در خروجی از دستور Console.WriteLine() یا Console.Write() استفاده می کنیم.

Console.Write(): یک رشته را تایپ کن

Console.WriteLine(): یک رشته را تایپ کن و برو خط بعدی

• نمایش داده در خروجی

وقتی برنامه را اجرا می کنیم خروجی به حدی سریع می آید که کاربر قادر به دیدن آن نیست کنیم. برای دیدن خروجی از دستور روبه رو استفاده می کنیم: System.Console.ReadKey();

این دستور به این معناست که تا کاربر کلیدی را نزده از پنجره داس خارج نشود.

مثال: برنامه ای بنویسید که رشته سلام برنامه نویس را توی خروجی نمایش دهد:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace csprogrammer
{
    class Program
    {
        static void Main(string[] args)
        {
            System.Console.WriteLine("Hello Programmer");
            System.Console.ReadKey();
        }
    }
}
```

آیا می توانیم همین رشته رو طور دیگری رایت کنیم؟

بله ما می توانیم یک متغیر از نوع رشته تعریف کنیم و رشته یا حرف و یا کلمه مورد نظر خودمون رو داخل این ظرف بریزیم و محتویات این ظرف رو چاپ کنیم :

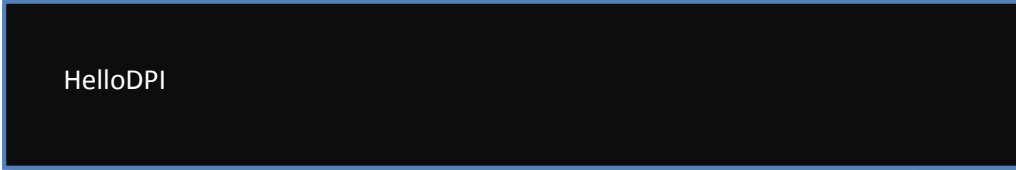
```
static void Main(string[] args)
{
    string s = "Hello Programmer";
    System.Console.WriteLine(s);
    System.Console.ReadKey();
}
```

مثال: برنامه ای بنویسید که دو تا رشته را کنار هم نمایش بده؟

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

```
System.Console.WriteLine("Hello" + "DPI");  
System.Console.ReadKey();  
}
```

خروجی برنامه:



HelloDPI

می بینیم که دو تا رشته به هم چسبیده ما برای جدا کردن رشته ها از (+) می توانیم استفاده کنیم:

```
static void Main(string[] args)  
{  
    System.Console.WriteLine("Hello" + " " + " DPI ");  
    System.Console.ReadKey();  
}
```

خروجی برنامه:



Hello DPI

• نحوه گرفتن داده از کاربر

تا اینجا تمام مقادیر رو خودمان وارد می کردیم و در خروجی نمایش داده می شد. ولی اگر بخواهیم وقتی برنامه را اجرا کردیم داده از کاربر گرفته شود. چیکار باید بکنیم؟؟؟
مثال: برنامه ای بنویسید که کاربر نامش رو وارد کند؟

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace csprogrammer
{
    class Program
    {
        static void Main(string[] args)
        {
            String Name;
            System.Console.Write ("Your Name=");
            Name= System.Console.ReadLine();
            System.Console.WriteLine(Name); // اگر مقدار داخل پرانتز رو داخل قرار بدیم
            عبارت داخل پرانتز عینا نمایش داده می شود
            System.Console.ReadKey();
        }
    }
}
```

خروجی برنامه:

Your Name=DPI

DPI

• نحوه گذاشتن توضیحات در برنامه

برای نوشتن توضیحات در یک خط از // استفاده می کنیم
برای نوشتن در چند خط از /* */ استفاده می کنیم

• نحوه گرفتن یک عدد از کاربر

داده های که در پنجره خروجی سی شارپ تایپ می شود از نوع رشته می باشد بنابراین اگر متغیر تعریف شده از نوع رشته باشد نیاز به تبدیل ندارد. در غیر این صورت بایستی ورودی به داده مورد نظر تبدیل شود.

فرض می کنیم می خواهیم یک مقدار صحیح از ورودی بخوانیم. روش کار به صورت زیر است

```
int a;
a= Convert.ToInt32(System.Console. ReadLine ());
Char c;
c = Convert.ToChar(System.Console. ReadLine ());
String s;
s= System. Console. ReadLine ();
```

نکته: برای تعریف متغیرهایی که از یک نوع می باشند می توان از (کاما) استفاده کرد
مثال: برنامه ای بنویسید که دو عدد صحیح را با هم جمع کند؟

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace csprogrammer
{
    class Program
    {
        static void Main(string[] args)
        {
            int a,b,sum;
            System.Console.WriteLine("2 add sahih ra vardkonid: ");
```

```

a= Convert.ToInt32(System.Console. ReadLine ());
b= Convert.ToInt32(System.Console. ReadLine ());
sum= a+ b;
System.Console.Write(" hasel jam = " );
System.Console.WriteLine(sum);
System.Console.ReadKey();
}
}
}

```

خروجی برنامه:

```
2 add sahih ra vardkonid:
```

```
۵
```

```
۶
```

```
hasel jam = ۱۱
```

مثال: می خواهیم برنامه ای بنویسیم که این بار میانگین دو عدد را نمایش دهد؟

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace csprogrammer
{
    class Program
    {
        static void Main(string[] args)
        {

```

```

int a,b,sum;
double Avg;
double _Avg;

System.Console.WriteLine("2 add sahih ra vardkonid: ");
a= Convert.ToInt32(System.Console. ReadLine ());
b= Convert.ToInt32(System.Console. ReadLine ());
sum= a+ b;
Avg=sum / 2;
_Avg=sum / 2.0;
System.Console.Write(" avag Avg = " + Avg );
System.Console.Write(" avag _Avg = " + _Avg );
System.Console.ReadKey();
}
}
}

```

خروجی برنامه:

2 add sahih ra vardkonid:

8

9

avag Avg = ۸٫۵

avag _Avg = ۸٫۵

نکته: با وجود این که متغیر **Avg** را از نوع اعشاری تعریف کردیم فقط قسمت صحیح آن را نمایش می دهد .

چون (/) تقسیمی که ما انجام می دهیم دوکار انجام می دهد یکی برای اعداد صحیح و یکی برای اعداد اعشاری اگر یک طرف قضیه اعشار باشد (چه در قسمت عدد و چه در قسمت متغیر) تقسیمش اعشاری می شود و اگر دو طرف قضیه صحیح باشد پس در خروجی قسمت صحیح را نمایش می دهد.

مثال: برنامه ای بنویسید که مربع یک عدد را نمایش دهد؟

مربع یک عدد: یعنی عدد در خودش ضرب شود مثلاً مربع عدد ۵ برابر است با ۲۵

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace csprogrammer
{
    class Program
    {
        static void Main(string[] args)
        {
            int a,b;
            System.Console.Write("\ add sahih ra vardkonid: ");
            a= Convert.ToInt32(System.Console.ReadLine());
            b=a*a;
            System.Console.Write( b );
            System.Console.ReadKey();
        }
    }
}
```

خروجی برنامه:

```
1 add sahih ra vardkonid: 40
1600
```

مثال: برنامه ای بنویسید که دو عدد از ورودی خوانده و بدون متغیر کمکی X و Y را جابجا کند؟

```
int x, y;
x = Convert.ToInt32(System.Console.ReadLine());
y = Convert.ToInt32(System.Console.ReadLine());
به جای Convert.ToInt32 میتوانیم از int.parse استفاده کنیم
x = x + y;
y = x - y;
x = x - y;
System.Console.WriteLine(x);
System.Console.WriteLine(y);
System.Console.ReadKey();
```

• دستورات شرطی

با یک مثال شروع می کنیم : مثلاً من به شما می گویم برو نانوايي اگر خلوت بود ۲۰ تا نان بگیر ما در اینجا یک شرط گذاشتیم که اگر شرط درست بود این کار رو انجام بده.

نکته: دستورات شرطی به دو شکل **If** و **switch** به کار می رود .

• دستور if

شکل کلی دستور if به صورت زیر است

```
If ( شرط یا شروط )
{
    دستورات ۱
}
Else
{
    دستورات ۲
}
```

در صورتی که شرط یا شروط برقرار باشد دستورات ۱ انجام می شود و در غیر اینصورت دستورات ۲ انجام می شود. یعنی دستورات یک و دو همزمان اجرا نمی شوند.

نکته: این ساختار می تواند **else** نداشته باشد.

دستور if را می توان به شکل زیر تعمیم داد:

```
If ( شرط 1 )
{
    دستورات ۱
}
Else if ( شرط 2 )
{
    دستورات ۲
}
.
.
Else if ( شرط n-1 )
{
    دستورات n - ۱
}
Else
{
```

```

    دستور n
}

```

مثال: برنامه ای بنویسید که دو عدد را بگیرد و بزرگترین آن ها را پیدا کند.

```

Double x, y, max;
System.Console.WriteLine("Enter ۲ number:");
x = Convert.ToDouble(System.Console.ReadLine());
y = Convert.ToDouble(System.Console.ReadLine());
if (x > y)
{
    max = x;
}
else
    max = y;
System.Console.WriteLine("max = ", max);
System.Console.ReadKey();

```

خروجی برنامه:

```

Enter ۲ number:

10

20

max =20

```

مثال: برنامه ای بنویسید عددی را از ورودی خوانده و تشخیص دهد زوج است یا فرد ؟

```

int a,x,r;
System.Console.Write("Enter One Number: ");
a=Convert.ToInt32(System.Console.ReadLine());
x= a / 2;

```

```

r=a - (x*2);
// این دو دستور معادل r= a % 2 می باشد
نکته: علامت % باقیمانده را بدست می آورد
if (r==0)
System.Console.WriteLine("Number Is Zoje ");
else
System.Console.WriteLine("Number Is Fard");
System.Console.ReadKey();

```

خروجی
برنامه:

```

Enter One Number:10
Number Is Zoje

```

• گذاشتن توضیحات در برنامه

برای نوشتن توضیحات در یک خط از // استفاده می کنیم.
برای نوشتن در چند خط از /* */ استفاده می کنیم.

علامت مقایسه برابر است با ==

علامت مخالف برابر است با !=

مثال: برنامه ای بنویسید ماه تولد شخص را گرفته و فصل تولد او را تشخیص و چاپ نماید.

```

int a;
System.Console.Write("Enter Month Number : ");
a=Convert.ToInt32(System.Console.ReadLine());
if (a ==1 || a==2|| a==3)
System.Console.WriteLine("bahar");
else if (a ==4 || a==5|| a==6)
System.Console.WriteLine("tabestan");
else if (a>=7 && a<=9)

```

```

System.Console.WriteLine("paiiz");
else if (a>=10 && a<=12)
System.Console.WriteLine("zemstan");
else
System.Console.WriteLine("Adad eshtebah ast");
System.Console.ReadKey();

```

با استفاده از یک if تودر تو نیز می توان ساده تر نوشت .

```

int a;
System.Console.Write("Enter Number Month : ");
a=Convert.ToInt32(System.Console.ReadLine());
if (a >= 1 && a <= 12)
{
    if (a <= 3)
        System.Console.WriteLine("Bahar");
    else if (a <= 6)
        System.Console.WriteLine("Tabestan");
    else if (a <= 9)
        System.Console.WriteLine("Paiiz");
    else
        System.Console.WriteLine("Zemstan");
}
else
    System.Console.WriteLine("Add Eshtebah ast");
System.Console.ReadKey();

```

مثال: برنامه ای بنویسید که یک عدد را از ورودی دریافت و قدر مطلق آن را نمایش دهد؟

قدر مطلق دو حالت دارد :

۱- عدد بزرگتر از ۰ بود عینا عدد را چاپ می کند

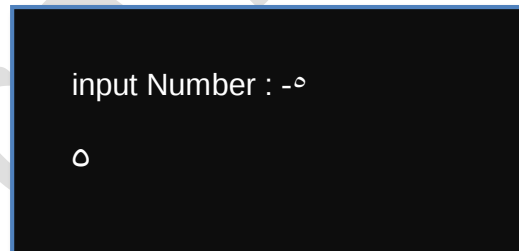
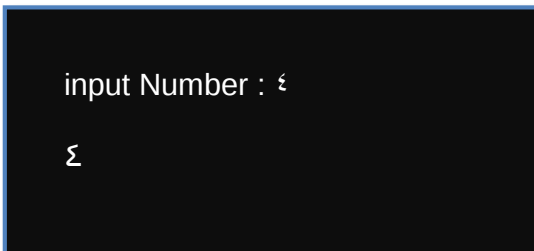
۲- عدد کوچکتر از ۰ بود آن عدد را در یک منفی ضرب می کند

```

int a;
System.Console.WriteLine("input Number : ");
a=Convert.ToInt32(System.Console.ReadLine());
if (a < 0)
{
    a = -a;
}
System.Console.WriteLine (a);
System.Console.ReadKey();

```

خروجی برنامه:



• دستور Switch

یکی دیگر از ساختارهای تصمیم دستور سوئیچ می باشد که نحوه عملکرد آن همانند دستور if است و تنها کدنویسی آن تفاوت دارد. شکل کلی این دستور به صورت زیر است:

Switch (شرط)

```

{
    Case مقدار ۱ : { دستورات ۱ ; }
    break
    Case مقدار ۲ : { دستورات ۲ ; }
    break
    default { دستورات ۲ ; }
    break;
}

```

مثال: برنامه ای بنویسید ماه تولد شخص را گرفته و فصل تولد او را تشخیص و چاپ نماید(با استفاده از دستورات سوئیچ)

```
int a;
a= Convert.ToInt32(System.Console.ReadLine());
switch (a)
{
case 1:
case 2:
case 3:
System.Console.WriteLine("bahar");
break;
case 4:
case 5:
case 6:
System.Console.WriteLine("tabestan");
break;
case 7:
case 8:
case 9:
System.Console.WriteLine("paiiz");
break;
case 10:
case 11:
case 12:

System.Console.WriteLine("zemestan");
break;
default:
System.Console.WriteLine("mah eshtebah vared shode ast!");
break;
}
Console.ReadKey();
```

اگر تعداد شرط ها بیشتر از ۳ تا بود به جای **if** بهتر است از دستور **Switch** استفاده کرد

مثال:

```
static void Main(string[] args)
{
    String strinput;

    Strinput= Console.ReadLine();

    if (Strinput == "Dazhyar" )
    {
        Console.WriteLine(10);
    }

    else if (Strinput == "Pardazesh")
    {
        Console.WriteLine(20);
    }

    else if (Strinput == "Isatis")
    {
        Console.WriteLine(30);
    }

    else
    {
        Console.WriteLine(1000);
    }

    Console.ReadKey();
}
```

```
static void Main(string[] args)
{
    String strinput=" ";

    Strinput= Console.ReadLine();

    switch (Strinput)
    {
        case "Dazhyar" :
            Console.WriteLine(10);
            break;

        case "Pardazesh": Console.WriteLine(20);
            break;

        case " " Isatis" :
            Console.WriteLine(30);
            break;

        default:
            Console.WriteLine(1000);
    }

    Console.ReadKey();
}
```

مثال: برنامه ای بنویسید دو عدد را به همراه یک عملگر محاسباتی از ورودی دریافت و نتیجه را در خروجی چاپ کند

(عملگرهای محاسباتی % / * - + است)

```
int a,b;
double ja;
Char ch;
Console.Write("Enter Number one: ");
a=Convert.ToInt32(Console.ReadLine());
Console.WriteLine("-----");
Console.Write("Enter Number tow: ");
b=Convert.ToInt32(Console.ReadLine());
Console.WriteLine("-----");
Console.Write("Select Steatment * or / or + or - or % ? ");
ch=Convert.ToChar(Console.ReadLine());
Console.Clear();
switch(ch)
{
case '*':
ja =a*b;
break;
case '+':
ja =a+b;
break;
case '-':
ja =a-b;
break;
case '/':
ja =a/b;
break;
case '%':
ja =a%b;
break;
default:
Console.Clear();
Console.WriteLine("Error For Statement !!");
ja =0;
break;
```

```

}
Console.Write(a);
Console.Write(ch);
Console.Write(b);
Console.Write(" = ");
Console.Write(ja); Console.ReadKey();

```

• حلقه های تکرار

اگر بخواهیم کدی را یک بار نوشته ولی چند بار تکرار کنیم از حلقه های تکرار استفاده می کنیم. هر حلقه تکرار شامل بخش های زیر است:

- 1- مقدار آغازین: مقداری که حلقه با آن شروع می شود.
- 2- مقدار پایانی (شرط خاتمه): مقداری که حلقه با آن پایان می پذیرد.
- 3- شمارنده حلقه: متغیری که از مقدار آغازین شروع و به مقدار پایانی ختم می شود.
- 4- گام حلقه: مقداری است که در هر بار اجرای حلقه به شمارنده اضافه شده یا از آن کاسته می شود تا از مقدار آغازین به مقدار پایانی برسیم.

حلقه های تکرار با سه دستور **for** و **while** و **do while** پیاده سازی می شوند

• ساختار حلقه for

```

For( مقدار آغازین; مقدار پایانی; مقدار حلقه )
{
    دستورات ;
}

```

• ساختار حلقه while

```

مقدار آغازین;
While( مقدار پایانی )
{
    دستورات ;
    گام حلقه ;
}

```

• ساختار حلقه do while

```
مقدار آغازین;
Do
{
    دستورات;
    گام حلقه;
}
While (مقدار پایانی)
```

• تفاوت حلقه while و do while

در حلقه for و while قبل از اجرای حلقه شمارنده با مقدار پایانی مقایسه می شود اگر یکی بود حلقه به هیچ عنوان اجرا نمی شود یعنی در این دو حلقه ممکن است حلقه اصلا اجرا نشود. ولی در حلقه do while حلقه حداقل یکبار اجرا شده سپس شرط خاتمه حلقه چک می شود.

نکته: در گام حلقه برای اضافه کردن و کم کردن گام از این سه روش می توان استفاده کرد:

```
i = i + 1;
```

```
i += 1;
```

```
i ++;
```

```
i = i - 1;
```

```
i -= 1;
```

```
i --;
```

میخواهیم درمورد دستور پلاس پلاس و ماینس ماینس حرف بزنیم
در قالب یه مثال این موضوع رو بررسی میکنیم
توی این مثال من ابتدا اومدم یه متغیر از نوع اینتیجر تعریف کردم بنام آی و مقدارش برابر صفر قرار دادم
و توخروجی رایتش کردم دقت کنید:

```
static void Main(string[] args)
{
    int i = 0;
    Console.WriteLine(i);
```

```
Console.ReadKey();
}
```

خروجی برنامه:

0

در خروجی باید صفر را به ما نشان دهد درسته؟؟؟

میبینیم که این اتفاق افتاده حالا بیاییم دستور پلاس پلاس رو به `i` بدیم ببینیم چه اتفاقی براش میفته؟
یعنی به اینصورت:

```
static void Main(string[] args)
{
    int i = 0;
    Console.WriteLine(i++);
    Console.ReadKey();
}
```

الان انتظار داریم یک رو برامون رایت کنه دیگه درسته ؟؟؟؟
اما این اتفاق نمیفته:

خروجی برنامه:

0

همونطور که میبینید دوباره صفر رو برامون رایت کرد.

خیلی خوب بیاییم علامت پلاس پلاس رو قبلش بزاریم ببینیم چه اتفاقی میفته ؟؟؟؟

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

```
{
    int i = 0;
    Console.WriteLine(++i);
    Console.ReadKey();
}
```

خروجی برنامه:

1

همونطور که نتیجه رو میبینید عدد یک رو تو خروجی نمایش داد خوب براستی چرا همچین اتفاقی افتاد؟؟؟؟

بینید تو قسمت اول یعنی ++i ابتدا اومد مقدار i رو رایت کرد و فرصت پیدا نکرد پلاس پلاس رو تو برنامه شرکت بده اما تو برنامه دوم ما اومدیم ابتدا پلاس پلاس رو شرکت دادیم و در واقع یه مقداری به متغیر خودمون اضافه کردیم و مقدارشو رایت کردیم:
همین دستور برای ماینس ماینس یا همون منها منها هم صادقه یعنی:

```
int i = 10;
Console.WriteLine(--i);
```

خروجی برنامه:

9

تو خروجی انتظار داریم عدد 9 رو ببینیم

مثال: برنامه ای بنویسید اعداد ۲ تا ۱۰۰۰ را چاپ کند؟ (با حلقه for)

```
int i ;
for (i = 2; i <= 1000; i++)
{
    System. Console.WriteLine(i);
}
System. Console.ReadKey();
```

همین برنامه با حلقه **while**

```
int i ;
i=2;
while(i<1000);
{
    System. Console.WriteLine(i);
    i ++;
}
System. Console.ReadKey();
```

همین برنامه با حلقه **do while**

```
int i ;
i=2;
do
{
    System. Console.WriteLine(i);
    i ++;
} while(i<1000);

System. Console.ReadKey();
```

مثال : برنامه ای بنویسید مضرب ۵ تا ۵۰ را چاپ کند؟ ← ۵۰ -۲۰ -۱۵ -۱۰ -۵

```
for ( int i = 5; i <= 50; i+=5)
{
    System. Console. WriteLine (i);
}
System. Console.ReadKey();
```

مثال : برنامه ای بنویسید مضرب ۳ تا n را چاپ کند؟

```
int n ;
n= Convert.ToInt32(System.Console.ReadLine());
for ( int i = 3; i <= n; i+=3)
{
    System. Console. WriteLine (i);
}
System. Console.ReadKey();
```

مثال : برنامه ای بنویسید که اعداد n تا m را در خروجی چاپ کند؟

```
int n,m;
n= Convert.ToInt32(System.Console.ReadLine());
m= Convert.ToInt32(System.Console.ReadLine());

for ( int i = n; i <= m; i++)
{
    System. Console. WriteLine (i);
}
System. Console.ReadKey();
```

مثال : برنامه ای بنویسید که اعداد بین n تا m را در خروجی چاپ کند؟

```
int n,m;

n= Convert.ToInt32(System.Console.ReadLine());
m= Convert.ToInt32(System.Console.ReadLine());

for ( int i = n+1; i < m; i++)
{
    System. Console. WriteLine (i);
}
System. Console.ReadKey();
```

مثال : برنامه ای بنویسید که مقسوم علیه های یک عدد را در خروجی چاپ کند؟

مقسوم علیه: یعنی اعدادی که بر آن عدد بخش پذیر باشد

```
int n;

System.Console.Write("input One Number : ");
n= Convert.ToInt32(System.Console.ReadLine());

for ( int i = 1; i < n; i++)
{
    if (n % i ==0)
    {
        System. Console. WriteLine (i);
    }
}
System. Console.ReadKey()
```

خروجی برنامه:

```

input One Number : 12

1
2
3
4
6
12

```

مثال : برنامه ای بنویسید که اعداد یک رقمی را در خروجی نمایش دهد؟

```

for ( int i = 1; i <=9; i+=2)
{
    System. Console. WriteLine (i);
}
System. Console.ReadKey();

```

مثال : برنامه ای بنویسید که مجموع اعداد ۱ تا n را در خروجی نمایش دهد؟

```

int n;
int Sum =0;
n= int .parse(System.Console.ReadLine());

for ( int i = 1; i <=n; i++)
{
    Sum+=i; // معادل این دستور Sum= Sum+i;
}
System. Console. WriteLine (Sum);

```

System. Console.ReadKey();

مثال: برنامه ای بنویسید که حاصلضرب اعداد ۱ تا ۵ را در خروجی نمایش

$$1*2*3*4*5=120$$



دهد؟

```
int D=1;
for ( int i = 1; i <=5; i++)
{
    D *=i;
}
System. Console. WriteLine (D);
System. Console.ReadKey();
```

مثال : برنامه ای بنویسید اعداد 10 تا ۱ را چاپ کند؟

```
int i ;
for (i = 10; i >= 1; i--)
{
    System. Console.WriteLine(i);
}
System. Console.ReadKey();
```

• ماتریس یا حلقه های تو در تو

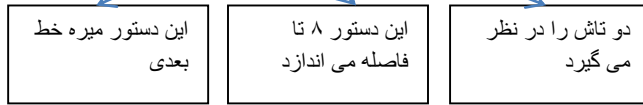
یعنی یک حلقه تو دل حلقه دیگر است

← \n یعنی یک خط بیا پایین تر

← \t مثل کلید Tab هست ۸ تا فاصله می اندازد

← \\ اگر خواستیم دو تای آنها (\t و \n) را در نظر بگیرد باید آخر دستور \\ قرار دهیم

("my name \n is parsa \t Rabiey \\")



مثال : برنامه ای بنویسید که شکل زیر را در خروجی نمایش دهد؟

```
for ( int i = 1; i <=3; i++)
{
```

```
    for ( int j = 1; j <=4; j++)
    {
```

```
        System. Console. WriteLine ( " \t * " );
```

```
    }
```

```
    System. Console. WriteLine (); // یک سطر بیا پایین تر
```

```
}
```

```
System. Console.ReadKey();
```

```
* * * *
* * * *
* * * *
```

مثال : برنامه ای بنویسید که شکل زیر را در خروجی نمایش دهد؟

```
for ( int i = 1; i <=3; i++)
{
```

```
    for ( int j = 1; j <=4; j++)
    {
```

```
        System. Console. WriteLine ( " \t ( " + i + "," +
```

```
    }
```

```
    System. Console. WriteLine (); // یک سطر بیا پایین تر
```

```
}
```

```
(1,1) (1,2) (1,3) (1,4)
(2,1) (2,2) (2,3) (2,4)
(3,1) (3,2) (3,3) (3,4)
```

```
System. Console.ReadKey();
```

مثال: حل جدول ضرب با استفاده از حلقه های تو در تو

```
for ( int i = 1; i <= ۱۰; i++)
{
    for ( int j = 1; j <= ۱۰; j++)
    {
        System. Console. WriteLine (i * j + " \ t" );
    }
    System. Console. WriteLine (); // یک سطر بیا پایین تر
}

System. Console.ReadKey();
```

مثال : برنامه ای بنویسید که شکل زیر را در خروجی نمایش دهد؟

```
for ( int i = 1; i <= 4; i++)
{
    for ( int j = 1; j <= i; j++)
    {
        System. Console. WriteLine (" * ");
    }

    System. Console. WriteLine (); // یک سطر بیا پایین تر
}

System. Console.ReadKey();
```

```
*
*  *
*  *  *
*  *  *  *
```

مثال : برنامه ای بنویسید که شکل زیر را در خروجی نمایش دهد؟

```
for ( int i = 4; i >=1; i--)
{
    for ( int j = 1; j <=i; j++)
    {
        System. Console. WriteLine (" * ");
    }

    System. Console. WriteLine (); // یک سطر بیا پایین تر
}

System. Console.ReadKey();
```

```
* * * *
* * *
* *
*
```

مثال : برنامه ای بنویسید که شکل زیر را در خروجی نمایش دهد؟

```
int Count=10 ;

for ( int i = 1; i <=8; i+=2)
{
    for ( int c = 1; c <= Count; c++)
    {
        System. Console. WriteLine (" ");
    }
    for ( int j = 1; j <=i; j++)
    {

        System. Console. WriteLine (" * ");
    }
}
```

```
*
* * *
* * * * *
* * * * *
```

```

        System. Console. WriteLine (); // یک سطر بیا پایین تر
    }

    System. Console.ReadKey();

```

مثال : برنامه ای بنویسید که شکل زیر را در خروجی نمایش دهد؟

```

for ( int i = 1; i <=10; i++)
{
    for ( int j = 1; j <=10; j++)
    {
        if ( i = j || i + j =11 )
        {
            System. Console. WriteLine ( " x " );
        }
        else
        {
            System. Console. WriteLine ( "   " );
        }
    }

    System. Console. WriteLine (); // یک سطر بیا پایین تر
}

System. Console.ReadKey()

```

```

      x           x
      x           x
      x           x
      x       x
      x       x
      x       x
      x       x
      x       x
      x       x
      x       x

```

برنامه ای بنویسید که شکل زیر را در خروجی نمایش دهد؟ (فقط با این تفاوت که تعداد سطر و ستون را از کاربر بگیرد)

```

int n ;

int Countspace=n ;

for ( int i = 1; i <=n; i++)

```

```

{
//-----
    for ( int c; c <= Countspace; c++)
    {
        System. Console. WriteLine ( " ");
    }
    Countspace --;
//-----
    for ( int j = 1; j <=i; j++)
    {
        System. Console. WriteLine ( " x ");
    }
    System. Console. WriteLine (); //یک سطر بیا پایین تر
}
System. Console.ReadKey();

```

آرایه ها

• آرایه های یک بعدی

هنگامی که بخواهیم تعدادی متغیر یکسان و از یک نوع ذخیره کنیم به جای تعریف چندین متغیر مختلف یک متغیر تعریف کرده و با اندیس، به تعداد دلخواه خانه های حافظه را اختصاص می دهیم. به عنوان مثال اگر بخواهیم حقوق ۲۰ کارمند یک اداره را ذخیره کنیم، به جای تعریف ۲۰ متغیر مختلف، از آنجا که کلیه داده ها برای نگهداری حقوق هستند، یک متغیر تعریف کرده و به تعداد ۲۰ عدد اندیس گذاری می کنیم. نحوه تعریف آرایه ها در C# به صورت زیر می باشد:

[طول آرایه] نوع متغیر = new نام آرایه [نوع متغیر];

Int [] a = new int [10];

Double [] b = new double [25];

Char [] c = new char [30];

نکته: اولین عنصر آرایه صفر است.

• تعریف متغیر ها و آرایه ها:

ببینید متغیر ها فقط به یک نقطه از حافظه اشاره می کنند اما آرایه ها به چند نقطه از حافظه میتونن اشاره کنند.

مثلاً اگر یه متغیر ی داشته باشیم بنام **i** و اون رو برابر ۲ قرار بدیم از نوع اینتیجر باشه، این متغیر **i** ما فقط به یک نقطه از حافظه اشاره میکنه که مقدار ۲ در اون قرار داده شده. اما حالا بیاییم در قالب یه آرایه در نظر بگیریم:

Int [] d;

۲	۸	۴۴	۱۷	۶۶	۶	۱	۸۷	۶۵	۲۱
---	---	----	----	----	---	---	----	----	----

همونطور که میبینید ما یه آرایه تعریف کردیم از نوع عددی اینتیجر و تونستیم مقادیر زیادی رو بهش اختصاص بدیم که هر کدوم از این خونه ها به یه مکانی از حافظه اشاره میکنند که مقداری در اونها ذخیره شده است

در اینجا یه آرایه از نوع رشته ای تعریف میکنیم بنام **S** که دو تا خونه داره و:

String [] s=new String [2];

Dazhyar	Pardazesh
---------	-----------

شماره خونه ها هم از صفر شروع میشه یعنی تو قسمت بالا ما دو تا خونه داریم با اندیس صفر و اندیس یک که اندیس صفر اشاره به خونه دازیار میکنه و اندیس یک اشاره به خونه پردازش میکنه یعنی

S[0]= " Dazhyar";
S[1]= " Pardazesh ";

نکته: آرایه با حلقه **for** مچ است.

مثال : برنامه ای بنویسید که ۵ عدد را از کاربر بگیره و دوباره آن را چاپ کند؟

Int [] mynumber = new int [5];

```
for ( int i = 0; i <5; i++)
{
```

```
    System. Console. WriteLine (" number "+ i + " = ");
```

```
    mynumber [ i ]=int .parse(System. Console.ReadKey());
```

}

System. Console.ReadKey();

مثال : برنامه ای بنویسید 10 عدد صحیح را از ورودی خوانده و درون آرایه ذخیره کند . سپس مجموع اعداد خوانده شده را محاسبه و چاپ کند.

```
int [ ] a= new int [10];
```

```
int sum=0 , i;
```

```
Console.WriteLine("Enter 10 int Number= ");
```

```
for(i=0;i<10;i++)
```

```
    a[i]= Convert.ToInt32(Console.ReadLine());
```

```
for(i=0;i<10;i++)
```

```
    sum += a[i];
```

```
Console.WriteLine(sum);
```

```
Console.ReadKey();
```

• آرایه های دو بعدی یا ماتریس

آرایه هایی که تا الان بررسی کردیم آرایه های یک بعدی یا بردار بودند . اکنون می خواهیم به بررسی آرایه هایی دو بعدی بپردازیم که m ستون و n سطر دارد. شکل کلی تعریف یک آرایه دو بعدی به صورت زیر است:

[بعد دوم , بعد اول] نوع آرایه NEW = نام آرایه [,] نوع آرایه

```
INT [ , ] A= NEW INT [3,2]
```

مثال : برنامه ای بنویسید که یک آرایه 5×6 را از ورودی خوانده سپس آن را چاپ کند؟

```
int [ , ] a= new int [5,6];
int i,j;
Console.WriteLine(" Enter Data For Matris: ");
for (i = 0; i < 5; i++) // سطر
for (j = 0; j < 6; j++) // ستون
a[i, j] = Convert.ToInt32(Console.ReadLine());
Console.WriteLine(" Matris Is ");
for (i = 0; i < 5; i++) // سطر
{
for (j = 0; j < 6; j++) // ستون
{
Console.Write(a[i, j]);
Console.Write(" ");
}
Console.WriteLine();
}
Console.ReadKey();
```

نکته: در این نوع ماتریس به صورت سطری می باشد. برای پیمایش ستونی باید جای i و j را عوض کنیم

مثال: برنامه ای بنویسید که یک ماتریس 10×20 را از ورودی گرفته ترانهاده آن را در خروجی چاپ کند)

ترانهاده یعنی تعویض سطر و ستون)

```
int[,] a = new int[10, 20];
int[,] t = new int[20, 10];
int i, j;
Console.WriteLine(" Enter Data For Matris: ");
Console.WriteLine("Enter numbers of 10*20 array:");
for (i = 0; i < 10; i++)
for (j = 0; j < 20; j++)
a[i, j] = Convert.ToInt16(Console.ReadLine());
for (i = 0; i < 10; i++)
for (j = 0; j < 20; j++)
```

```
t[j, i] = a[i, j]; }
for (i = 0; i < 20; i++)
for (j = 0; j < 10; j++)
Console.WriteLine(t[i, j]);
Console.ReadKey();
```

مفهوم زیر بنا یا تابع

تابع چیست ببینید شما اگر قصد دارید یک عملیات ریاضی در برنامه انجام بدید برای کوتاه تر شدن دستورات یک بخش و راحت تر نوشتن کد باید از تابع استفاده کنید که مقدار برگردونه در این زبان برنامه نویسی تابع به صورت زیر تعریف میشود :

(نوع و اسامی پارامترها) نام تابع نوع خروجی تابع سطح دسترسی به تابع

```
{
    بدنه ی تابع
}
```

هر تابعی می تواند صفر یا تعداد بیشماری آرگومان ورودی و صفر یا تعداد بیشماری خروجی داشته باشد . بوسیله یک تابع می توان پیچیدگی کار را مخفی کرد و صرفا با صدا زدن نام آن ، یک سری از عملیات را انجام داد. تابع حتما باید داخل **class Program** و داخل **static void Main(string[] args)** فراخوانی شود .

نکات:

۱- برای اینکه بتوانیم تابع را فراخوانی کنیم باید کلمه **Static** را بین سطوح دسترسی به تابع و خروجی تابع بنویسیم .

۲- اگر خروجی تابع از نوع **void** بود یعنی خروجی ندارد که در واقع بهش می گویند پروسیجر

۳- اگر خروجی تابع از نوع `int` یا `string` یا هر چیز دیگری بود بود یعنی خروجی دارد و باید از کلمه `return` برای برگرداندن خروجی استفاده کنیم. که به این در واقع تابع می گویند.

مثالی از پروسیجر یعنی با نوع خروجی `void`

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace csprogrammer
{
    class Program
    {
        static void Main(string[] args)
        {
            sum();
            Console.ReadKey();
        }
        public static void Sum()
        {
            int a , b , c;
            a= Convert.ToInt32(Console.ReadLine());

            b= Convert.ToInt32(Console.ReadLine());

            c= a + b;

            Console.WriteLine(c);
        }
    }
}
```

```

    }
}
}

```

همان مثال با نوع خروجی `int`

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace csprogrammer
{
    class Program
    {
        static void Main(string[] args)
        {
            int p;
            p= sum();
            Console.WriteLine( p );
            Console.ReadKey();
        }
        public static int Sum()
        {
            int a , b , c;
            a= Convert.ToInt32(Console.ReadLine());
            b= Convert.ToInt32(Console.ReadLine());

            c= a + b;

            return c ;
        }
    }
}

```

}

متغیر سراسری: متغیری که بعد از `class Program` تعریف می شود متغیر سراسری گفته می شود.

برای فراخوانی متغیر سراسری باید از کلمه کلیدی `static` استفاده کنیم.

متغیر محلی: متغیری که داخل تابع تعریف می شود.

```
class Program
```

```
{
    static int count;      → متغیر سراسری
    static void Main(string[] args)
    {
        int p;
        p= sum();
        Console.WriteLine( p );
        Console.ReadKey();
    }
    public static int Sum()
    {
        int a , b , c;      → متغیر محلی

        a= Convert.ToInt32(Console.ReadLine());
        b= Convert.ToInt32(Console.ReadLine());

        c= a + b;

        return c ;
    }
}
```

گفتیم که هر تابع می تواند صفر یا تعداد بیشماری آرگومان ورودی و صفر یا تعداد بیشماری آرگومان خروجی داشته باشد

یک مثال:

```
class Program
```

```
{
    static void Main(string[] args)
    {
        int p;
        p= sum( 80 , 60 );
        Console.WriteLine( p );
        Console.ReadKey();
    }
    public static int Sum( int a, int b)
    {
        int c;
        c= a + b;
        return c;
    }
}
```

برای فراخوانی این تابع چون ورودی قرار دادیم حتما باید مقداری مرتبط با (80 , 60)
نوع ورودی بگذاریم

توابع همنام (over load)

گاهی اوقات لازم می شود که دو یا چند تابع با یک نام داشته باشیم این در صورتی امکان پذیر است که نوع یا تعداد آرگومان های ورودی با هم فرق داشته باشد

```
class Program
{
    static void Main(string[] args)
    {
        int p= sum( 80 , 60 );
        int d= sum( "DPI" );

        Console.WriteLine( p );
        Console.WriteLine( d );
        Console.ReadKey();
    }
    public static int Sum( int a, int b)
    {
        return a + b;
    }
    public static int Sum( string a)
    {
        return a ;
    }
}
```

پروژه ماشین حساب با استفاده از توابع

```
class Program
{
    static void Main(string[] args)
    {
        int n,m;
        Console.Write ( "    number a =" );
        n= int.parse(Console.ReadLine());
        Console.Write ( "    number b =" );
        m= int.parse(Console.ReadLine());
        Console.WriteLine( " sum=" + Sum( n,m) );
        Console.WriteLine( " Mul =" + Mul ( n,m) );
        Console.WriteLine( " Min =" + Min ( n,m) );
        Console.WriteLine( " Div =" + Div ( n,m) );
        Console.ReadKey();
    }
    public static int Sum ( int a, int b)
    {
        return a + b;
    }
    public static int Mul ( int a, int b)
    {
        return a * b;
    }
    public static int Min ( int a, int b)
    {
        return a - b;
    }
    public static double Div ( int a, int b)
```

```

{
    return a / ( double ) b;
}
}

```

پروژه ماشین حساب با استفاده از توابع و دستور switch

```

class Program
{
    static void Main(string[] args)
    {
        int a,b;
        Console.Write ( "number a =" );
        a= int.parse(Console.ReadLine());
        Console.Write ( "number b =" );
        b= int.parse(Console.ReadLine());
        String key;
        while ( true )
        {
            Console.WriteLine( " \n\t\t + - * / 1 = Exit=");
            switch ( key)
            {
                case "+": Console.WriteLine(Sum( n,m));
                    break;

                case "*": Console.WriteLine(Mul ( n,m));

                    break;

                case "-": Console.WriteLine(Min ( n,m));

```

```

        break;

        case "/": Console.WriteLine(Div ( n,m));

        break;

        case "1":return

    }

    Console.ReadKey();
}

public static int Sum ( int a, int b)
{
    return  a + b;
}

public static int Mul ( int a, int b)
{
    return  a * b;
}

public static int Min ( int a, int b)
{
    return  a - b;
}

public static double Div ( int a, int b)
{
    return  a / ( double ) b;
}
}

```

لطفا سوالات و انتقادات خود را از طریق سایت ، با ما در میان بگذارید.

آرزوی موفقیت برای شما | شرکت فنی مهندسی دازیار پردازش ایستایس