Console.ReadLine(); > To take input from user

It returns input in string form

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FirstProject

{

class Second

{

static void Main()

{// To take input from user

string name;

Console.WriteLine("Enter Name");

name = Console.ReadLine();

Console.WriteLine("Your Name is " + name);

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FirstProject

{

class Second

{

static void Main()

{// To take input from user

// To take integers

int num;

Console.WriteLine("Enter Number");

num = Convert.ToInt16(Console.ReadLine());

Console.WriteLine("Entered number is " + num);

}

}

}

<https://www.w3schools.com/cs/cs_data_types.asp>

<https://www.tutorialsteacher.com/csharp/csharp-data-types>

|  |
| --- |
|  |
| Type | Description | Range | Suffix |
| byte | 8-bit unsigned integer | 0 to 255 |  |
| sbyte | 8-bit signed integer | -128 to 127 |  |
| short | 16-bit signed integer | -32,768 to 32,767 |  |
| ushort | 16-bit unsigned integer | 0 to 65,535 |  |
| int | 32-bit signed integer | -2,147,483,648 to 2,147,483,647 |  |
| uint | 32-bit unsigned integer | 0 to 4,294,967,295 | u |
| long | 64-bit signed integer | -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 | l |
| ulong | 64-bit unsigned integer | 0 to 18,446,744,073,709,551,615 | ul |
| float | 32-bit Single-precision floating point type | -3.402823e38 to 3.402823e38 | f |
| double | 64-bit double-precision floating point type | -1.79769313486232e308 to 1.79769313486232e308 | d |
| decimal | 128-bit decimal type for financial and monetary calculations | (+ or -)1.0 x 10e-28 to 7.9 x 10e28 | m |
| char | 16-bit single Unicode character | Any valid character, e.g. a,\*, \x0058 (hex), or\u0058 (Unicode) |  |
| bool | 8-bit logical true/false value | True or False |  |
| object | Base type of all other types. |  |  |
| string | A sequence of Unicode characters |  |  |
| DateTime | Represents date and time | 0:00:00am 1/1/01 to 11:59:59pm 12/31/9999 |  |

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FirstProject

{

class Second

{

static void Main()

{// To take input from user

// To take integers

int num;

Console.WriteLine("Enter Number");

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

char c;

Console.WriteLine("Enter character");

c = Convert.ToChar(Console.ReadLine());

Console.WriteLine("Entered number is " + num);

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FirstProject

{

class Third

{

static void Main()

{

int x, y;

Console.WriteLine("Enter Num 1");

x = Convert.ToByte(Console.ReadLine());

Console.WriteLine("Enter Num 2");

y = Convert.ToByte(Console.ReadLine());

Console.WriteLine("Sum of {0} and {1} is {2}", x, y, (x + y));

Console.WriteLine("Difference of {0} and {1} is {2}", x, y, (x - y));

Console.WriteLine("Product of {0} and {1} is {2}", x, y, (x \* y));

Console.WriteLine("Quotient of {0} and {1} is {2}", x, y, (x / y));

Console.WriteLine("Remainder of {0} and {1} is {2}", x, y, (x % y));

}

}

}

In any programming language , statements could be of how many types

1. Sequential ( Program statements executes in a sequence)
2. Conditional (( Program statements executes depending upon some condition)

If

If else

If elseif elseif else

switch

1. Repetitive (( Program statements executes multiple times)

Loops

1. Do while
2. While
3. For
4. Foreach

Conditional statements

**Syntax of if**

If(condition)

{

Statements;

}

**2. If else**

**If(condition)**

**{**

**Statements;**

**}**

**Else**

**{**

**Statements;**

**}**

**3.Multiple conditions**

**If else if**

If(condition1)

{

Statements;

}

Else If(condition2)

{

Statements;

}

…..

Else If(condition n)

{

Statements;

}

Else

{

statements;

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FirstProject

{

class Third

{

static void Main()

{

// Enter two numbers and choice, if choice is 1 , do addition

// if it is 2 , do subtraction

int x, y;

int choice;

Console.WriteLine("Enter Num 1");

x = Convert.ToByte(Console.ReadLine());

Console.WriteLine("Enter Num 2");

y = Convert.ToByte(Console.ReadLine());

Console.WriteLine("ENter Choice");

choice = Convert.ToByte(Console.ReadLine());

if(choice==1)

Console.WriteLine("Sum of {0} and {1} is {2}", x, y, (x + y));

else if(choice==2)

Console.WriteLine("Difference of {0} and {1} is {2}", x, y, (x - y));

else if(choice==3)

Console.WriteLine("Product of {0} and {1} is {2}", x, y, (x \* y));

else if(choice==4)

Console.WriteLine("Quotient of {0} and {1} is {2}", x, y, (x / y));

else if(choice==5)

Console.WriteLine("Remainder of {0} and {1} is {2}", x, y, (x % y));

else

Console.WriteLine("Invalid choice");

}

}

}

**Switch syntax**

Switch(variable/ expression)

Case value1 : statement; break;

Case value 2: statemenets; braeak;

Default : statements ; break;

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FirstProject

{

class swicthdemo

{

static void Main()

{

// Enter two numbers and choice, if choice is 1 , do addition

// if it is 2 , do subtraction

int x, y;

int choice;

Console.WriteLine("Enter Num 1");

x = Convert.ToByte(Console.ReadLine());

Console.WriteLine("Enter Num 2");

y = Convert.ToByte(Console.ReadLine());

Console.WriteLine("ENter Choice");

choice = Convert.ToByte(Console.ReadLine());

switch (choice)

{

case 1: Console.WriteLine("Sum of {0} and {1} is {2}", x, y, (x + y));

break;

case 2: Console.WriteLine("Difference of {0} and {1} is {2}", x, y, (x - y));

break;

case 3: Console.WriteLine("Product of {0} and {1} is {2}", x, y, (x \* y));

break;

case 4: Console.WriteLine("Quotient of {0} and {1} is {2}", x, y, (x / y));

break;

case 5: Console.WriteLine("Remainder of {0} and {1} is {2}", x, y, (x % y));

break;

default:

Console.WriteLine("Invalid choice");

break;

}

}

}

}

Loops

Statements in a loop

3 statements

1. Initialization part
2. Termination point / Condition
3. Increment / Decrement statement

**Do while**

Initialization part;

Do

{

Statements;

Inc / dec;

} while(condition)

**While loop**

Initialization part;

While(condition)

{

Statements;

ince / decre;

}

**For Loop**

**For(**Initialization part;condtion; incr/decr)

{

Statements;

}

**Do while & while**

**Do while always executes one time even if condition is false**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FirstProject

{

class LoopsDemo

{

static void Main()

{

int n = 1;

do

{

Console.WriteLine(n);

n++;

} while (n <= 10);

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FirstProject

{

class LoopsDemo

{

static void Main()

{

int n = 11;

while(n<=10)

{

Console.WriteLine(n);

n++;

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FirstProject

{

class LoopsDemo

{

static void Main()

{

int n = 1;

for (; n <= 10;)

{

Console.WriteLine(n);

n++;

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FirstProject

{

class LoopsDemo

{

static void Main()

{

for (int n = 2; n <= 50;n+=2)

{

Console.WriteLine(n);

}

}

}

}

Multiples of 5

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FirstProject

{

class LoopsDemo

{

static void Main()

{

for (int n = 100; n >= 50;n-=5)

{

Console.WriteLine(n);

}

}

}

}