



Academy .NET

Console Input/Output

Sommario

- Stampa sulla console
Stampa di stringhe e numeri
- Lettura dalla console
Lettura di caratteri
Lettura di stringhe
Conversione di stringhe in tipi numerici
Lettura di tipi numerici
- Vari esempi

La Console

- La console viene utilizzata per visualizzare le informazioni in una finestra di testo
- Può visualizzare diversi valori:
 - Stringhe**
 - Tipi numerici**
 - Tutti i tipi di dati primitivi**
- Per stampare sulla console si utilizza la classe Console (**System.Console**)

La classe Console

- Fornisce metodi per l'input e l'output dalla Console
- Input
 - Read (...):** legge un singolo carattere
 - ReadKey (...):** legge una combinazione di tasti
 - ReadLine (...):** legge una singola riga di caratteri
- Output
 - Write (...):** stampa l'argomento specificato sulla console
 - WriteLine (...):** stampa i dati specificati nella console e passa alla riga successiva

La classe Console

Stampa di una variabile di tipo stringa:

```
string str = "Hello C#!";  
...  
Console.WriteLine(str);
```

Stampa di più variabili utilizzando string formatting:

```
string name = "Marry";  
int year = 1987;  
...  
Console.WriteLine("{0} was born in {1}.", name, year);  
// Marry was born in 1987.
```

La stampa successiva inizierà dalla nuova riga.

La classe Console - Esempio

```
static void Main()
{
    string name = "Peter";
    int age = 18;
    string town = "Sofia";

    Console.Write("{0} is {1} years old from {2}.",
        name, age, town);
    // Result: Peter is 18 years old from Sofia.
    Console.Write("This is on the same line!");
    Console.WriteLine("Next sentence will be" +
        " on a new line.");

    Console.WriteLine("Bye, bye, {0} from {1}.",
        name, town);
}
```


La classe Console

- **{index [, alignment] [: formatString]}**

- **index**

L'indice **in base zero** dell'argomento la cui rappresentazione di stringa deve essere inclusa in questa posizione nella stringa

- **Allineamento**

Un numero intero con segno che indica la lunghezza totale del campo in cui è inserito l'argomento

un numero intero positivo - allineato a destra

un numero intero negativo - allineato a sinistra

La classe Console

- **{index [, alignment] [: formatString]}**
- **formatString**
Specifica il formato dell'argomento corrispondente del risultato della stringa , ad es. "F", "D", "0,00"
- Esempio:

```
static void Main()
{
    double pi = 1.234;
    Console.WriteLine("{0:0.000000}", pi);
    // 1.234000
}
```

La classe Console: Esempio

```
static void Main()
{
    int a=2, b=3;
    Console.Write("{0} + {1} =", a, b);
    Console.WriteLine(" {0}", a+b);
    // 2 + 3 = 5

    Console.WriteLine("{0} * {1} = {2}",
        a, b, a*b);
    // 2 * 3 = 6

    float pi = 3.14159206;
    Console.WriteLine("{0:F2}", pi); // 3,14

    Console.WriteLine("Bye - Bye!");
}
```

Leggere dalla Console



- Si utilizza la console per leggere le informazioni dalla riga di comando
- É possibile leggere:
 - Caratteri**
 - Stringhe**
 - Tipi numerici** (dopo la conversione)
- Per leggere dalla console si utilizzano i metodi **Console.Read()** e **Console.ReadLine()**

Leggere dalla Console

- Ottiene un singolo carattere dalla console (dopo aver premuto **[Enter]**)
Restituisce un risultato di tipo **int**
Restituisce **-1** se non ci sono più simboli
- Per ottenere il carattere effettivamente letto, si utilizza il **cast** su **char**:

```
int i = Console.Read ();  
char ch = (char) i; // Trasmette int a char  
// Ottiene il codice del simbolo inserito  
Console.WriteLine ("Il codice di" {0} "è {1}.", Ch, i);  
}
```

Console.ReadKey()

- Attende finché non viene premuta una combinazione di tasti
Legge un singolo carattere dalla console o una combinazione di tasti
- Restituisce un risultato di tipo **ConsoleKeyInfo**
KeyChar: contiene il carattere inserito
Modifiers: mantiene lo stato di [Ctrl], [Alt], ...

```
ConsoleKeyInfo key = Console.ReadKey();  
Console.WriteLine();  
Console.WriteLine("Character entered: " + key.KeyChar);  
Console.WriteLine("Special keys: " + key.Modifiers);
```

Console.ReadLine()

- Prende in input una riga di caratteri
- Restituisce un **valore stringa**
- Se viene premuto **Ctrl + Z** durante la lettura dell'input dalla console, il metodo restituisce **null**.

```
Console.Write("Please enter your first name: ");  
string firstName = Console.ReadLine();  
  
Console.Write("Please enter your last name: ");  
string lastName = Console.ReadLine();  
  
Console.WriteLine("Hello, {0} {1}!",  
    firstName, lastName);
```

Lettura di stringhe dalla console



- I tipi numerici non possono essere letti direttamente dalla console
- Per leggere un tipo di numero, procede come segue:
 - Leggere un valore di stringa
 - Convertirlo (**parse**) nel tipo numerico richiesto
- **int.Parse (stringa) int**
 - Converte una stringa in **int**

```
string str = Console.ReadLine()  
int number = int.Parse(str);  
  
Console.WriteLine("You entered: {0}", number);
```


Conversione di stringhe in numeri

I tipi numerici hanno un metodo **Parse(...)** per estrarre il valore numerico da una stringa:

int.Parse (stringa) - string -> int

long.Parse (string) - string -> long

float.Parse (string) - string -> float

Genera **FormatException** in caso di errore

```
string s = "123";  
int i = int.Parse(s); // i = 123  
long l = long.Parse(s); // l = 123L  
string invalid = "xxx1845";  
int value = int.Parse(invalid); // FormatException
```

Conversione di stringhe in numeri

```
static void Main()
{
    int a = int.Parse(Console.ReadLine());
    int b = int.Parse(Console.ReadLine());

    Console.WriteLine("{0} + {1} = {2}",
        a, b, a+b);
    Console.WriteLine("{0} * {1} = {2}",
        a, b, a*b);

    float f = float.Parse(Console.ReadLine());
    Console.WriteLine("{0} * {1} / {2} = {3}",
        a, b, f, a*b/f);
}
```

Conversione di stringhe in numeri

La conversione può essere eseguita anche utilizzando i metodi della classe **Convert**:

- **Convert.ToInt32 (stringa)** - string -> int
- **Convert.ToSingle (string)** - string -> float
- **Convert.ToInt64 (stringa)** - string -> long

```
string s = "123";  
int i = Convert.ToInt32(s); // i = 123  
long l = Convert.ToInt64(s); // l = 123L  
string invalid = "xxx1845";  
int value = Convert.ToInt32(invalid); // FormatException
```

Gestione degli errori

- A volte è necessario gestire gli errori durante la conversione di un numero:

Due opzioni: usare il blocco **try-catch** o **TryParse()**

- Gestione errori con TryParse ():

```
string str = Console.ReadLine();
int number;
if (int.TryParse(str, out number))
{
    Console.WriteLine("Valid number: {0}", number);
}
else
{
    Console.WriteLine("Invalid number: {0}", str);
}
```

Leggere e Scrivere in Console - Esempio

```
Console.Write("Enter person name: ");  
string person = Console.ReadLine();  
  
Console.Write("Enter company name: ");  
string company = Console.ReadLine();  
  
Console.WriteLine("  Dear {0},", person);  
Console.WriteLine("We are pleased to tell you " +  
    "that {1} has chosen you to take part " +  
    "in the \"Introduction To Programming\" " +  
    "course. {1} wishes you good luck!",  
    person, company);  
  
Console.WriteLine("  Yours,");  
Console.WriteLine("  {0}", company);
```

Calcolo Area - Esempio

```
Console.WriteLine("This program calculates " +  
    "the area of a rectangle or a triangle");  
  
Console.Write("Enter a and b (for rectangle) " +  
    " or a and h (for triangle): ");  
int a = int.Parse(Console.ReadLine());  
int b = int.Parse(Console.ReadLine());  
  
Console.Write("Enter 1 for a rectangle or 2 " +  
    "for a triangle: ");  
  
int choice = int.Parse(Console.ReadLine());  
double area = (double) (a*b) / choice;  
Console.WriteLine("The area of your figure " +  
    " is {0}", area);
```

Domande & approfondimenti

Academy .NET