

```

string a = "abc";
Int32.TryParse(a, out int c);
int b = Int32.Parse(a);
Console.WriteLine(c);
Console.WriteLine(b);

```

Exception Unhandled

System.FormatException: 'Input string was not in a correct format.'

[View Details](#) | [Copy Details](#) | [Start Live Share session...](#)

▸ [Exception Settings](#)

```

// See https://aka.ms/new-console-template
Console.WriteLine("Hello, World!");

```

```

string a = "abc";
Int32.TryParse(a, out int c);
int b = Int32.Parse(a);
Console.WriteLine(c);
Console.WriteLine(b);

```

```

[2] int32 b
)
// [2 1 - 2 36]
IL_0000: ldstr      "Hello, World!"
IL_0005: call       void [System.Console]System.Console::WriteLine(string)
IL_000a: nop

// [4 1 - 4 18]
IL_000b: ldstr      "abc"
IL_0010: stloc.0     // a

// [5 1 - 5 30]
IL_0011: ldloc.0     // a
IL_0012: ldloc.s   c
IL_0014: call      bool [System.Runtime]System.Int32::TryParse(string, int32)
IL_0019: pop

// [6 1 - 6 24]
IL_001a: ldloc.0     // a
IL_001b: call      int32 [System.Runtime]System.Int32::Parse(string)
IL_0020: stloc.2     // b

// [7 1 - 7 22]
IL_0021: ldloc.1     // c
IL_0022: call      void [System.Console]System.Console::WriteLine(int32)
IL_0027: nop

// [8 1 - 8 22]
IL_0028: ldloc.2     // b
IL_0029: call      void [System.Console]System.Console::WriteLine(int32)
IL_002e: nop
IL_002f: ret

} // end of method Program::'<Main>$'

```

Метод Parse сразу пробует получить число из его строки. Если там число, то он его возвращает, если нет, то выводит `FormatException`.

Метод `TryParse` сначала проверяет, можно ли получить число из строки. Если это возможно - возвращает `true` и полученное число, иначе возвращает `false`, но не выводит `Exception`.

Скорее всего `Parse` лучше использовать, если точно уверен, что результат не выдаст `Exception`. `TryParse` удобно использовать, когда нужно понять можно ли сконвертировать выражение в нужный формат, и в зависимости от результата, сделать определенные действия (так как возвращает `bool` значение).