

▼ C# Fundamentals

Datas

Datas:

DateTime Estrutura

Namespace: [System](#)

Assemblies: [mscorlib.dll](#), [System.Runtime.dll](#)

Representa um momento no tempo, geralmente expresso como uma data e hora do dia.

C#

 Copiar

```
public struct DateTime : IComparable, IComparable<DateTime>, IConvertible, IEquatable<DateTime>, IFormattable,  
System.Runtime.Serialization.ISerializable
```

Herança [Object](#) → [ValueType](#) → [DateTime](#)

Implementações [IComparable](#) , [IComparable<DateTime>](#) , [IConvertible](#) , [IEquatable<DateTime>](#) , [IFormattable](#) , [ISerializable](#)

Datas:

Inicializando um objeto DateTime

Você pode atribuir um valor inicial a um novo `DateTime` valor de várias maneiras diferentes:

- Chamar um construtor, qualquer um em que você especifique argumentos para valores ou use o Construtor implícito sem parâmetros.
- Atribuindo um `DateTime` para o valor de retorno de uma propriedade ou método.
- Analisar um `DateTime` valor de sua representação de cadeia de caracteres.
- Usando recursos de linguagem específica de Visual Basic para instanciar um `DateTime` .

Os trechos de código a seguir mostram exemplos de cada um:


Datas:

Invocar construtores

Você chama qualquer uma das sobrecargas do `DateTime` Construtor que especificam elementos do valor de data e hora (como ano, mês e dia, ou o número de tiques). O código a seguir cria uma data específica usando o `DateTime` Construtor que especifica o ano, o mês, o dia, a hora, o minuto e o segundo.

[!code-vbSystem.DateTime.Instantiation#1]


C#


 Copiar

```
var date1 = new DateTime(2008, 5, 1, 8, 30, 52);  
Console.WriteLine(date1);
```

Você invoca o `DateTime` Construtor implícito sem parâmetros da estrutura quando deseja que um seja `DateTime` inicializado para seu valor padrão. (Para obter detalhes sobre o Construtor implícito sem parâmetros de um tipo de valor, consulte [tipos de valor](#).) Alguns compiladores também dão suporte à declaração de um `DateTime` valor sem atribuir explicitamente um valor a ele. A criação de um valor sem uma inicialização explícita também resulta no valor padrão. O exemplo a seguir ilustra o `DateTime` Construtor implícito sem parâmetros em C# e Visual Basic, bem como uma declaração que não é uma `DateTime` atribuição no Visual Basic.

C#

 Copiar

 Executar

```
var dat1 = new DateTime();  
// The following method call displays 1/1/0001 12:00:00 AM.  
Console.WriteLine(dat1.ToString(System.Globalization.CultureInfo.InvariantCulture));  
// The following method call displays True.  
Console.WriteLine(dat1.Equals(DateTime.MinValue));
```

Datas:

Atribuindo um valor calculado

Você pode atribuir ao `DateTime` objeto um valor de data e hora retornado por uma propriedade ou método. O exemplo a seguir atribui a data e hora atuais, a data e hora UTC (hora universal coordenada) atual e a data atual para três novas `DateTime` variáveis.

[!code-vbSystem.DateTime.Instantiation#3]

C#

 Copiar

```
DateTime date1 = DateTime.Now;  
DateTime date2 = DateTime.UtcNow;  
DateTime date3 = DateTime.Today;
```



Datas:

Analizando uma cadeia de caracteres que representa um DateTime

Os `Parse` `ParseExact` métodos,, e `TryParse` `TryParseExact` All convertem uma cadeia de caracteres em seu valor de data e hora equivalente. Os exemplos a seguir usam `Parse` os `ParseExact` métodos e para analisar uma cadeia de caracteres e convertê-la em um `DateTime` valor. O segundo formato usa um formulário suportado pelo padrão [ISO 8601](#) para representar uma data e hora no formato de cadeia de caracteres. Essa representação padrão é geralmente usada para transferir informações de data nos serviços Web.

[!code-vbSystem.DateTime.Instantiation#4]

C#

 Copiar

```
var dateString = "5/1/2008 8:30:52 AM";
DateTime date1 = DateTime.Parse(dateString,
                                System.Globalization.CultureInfo.InvariantCulture);

var iso8601String = "20080501T08:30:52Z";
DateTime dateISO8602 = DateTime.ParseExact(iso8601String, "yyyyMMddTHH:mm:ssZ",
                                             System.Globalization.CultureInfo.InvariantCulture);
```

Os `TryParse` `TryParseExact` métodos e indicam se uma cadeia de caracteres é uma representação válida de um `DateTime` valor e, se for, executa a conversão.

Datas:

- **Links de documentação oficial:**
- **Documentação:**
- <https://docs.microsoft.com/pt-br/dotnet/api/system.datetime?view=net-5.0>



Dúvidas?