

# ▼ C# Fundamentals

## Nullable Types

# Nullable Types:

- Um tipo de valor anulável  $T?$  representa todos os valores de seu tipo de valor subjacente  $T$  e um valor nulo adicional. Por exemplo, você pode atribuir qualquer um dos três valores a seguir a uma variável  $bool?$ : `true` , `false` ou `null` .
- Qualquer tipo de valor anulável é uma instância da `System.Nullable<T>` estrutura genérica. Você pode se referir a um tipo de valor anulável com um tipo subjacente  $T$  em qualquer um dos seguintes formulários intercambiáveis: `Nullable<T>` ou `T?`

# Nullable Types:

- Normalmente, você usa um tipo de valor anulável quando precisa representar o valor indefinido de um tipo de valor subjacente. Por exemplo, um booleano ou bool variável pode ser apenas true ou false . No entanto, em alguns aplicativos, um valor de variável pode ser indefinido ou estar ausente. Por exemplo, um campo de banco de dados pode conter true ou false não pode conter nenhum valor, ou seja, NULL . Você pode usar o bool? tipo nesse cenário.



# Nullable Types:

- **Declaração e atribuição:**
- **Como um tipo de valor é implicitamente conversível para o tipo de valor anulável correspondente, você pode atribuir um valor a uma variável de um tipo de valor anulável, como faria com o tipo de valor subjacente. Você também pode atribuir o null valor. Por exemplo:**

# Nullable Types:

- Declaração e atribuição:

C#

```
double? pi = 3.14;  
char? letter = 'a';  
  
int m2 = 10;  
int? m = m2;  
  
bool? flag = null;  
  
// An array of a nullable value type:  
int?[] arr = new int?[10];
```

# Nullable Types:

- Links de documentação oficial:
- **Documentação:**
- <https://docs.microsoft.com/pt-br/dotnet/csharp/language-reference/builtin-types/nullable-value-types>





# Dúvidas?