

# Working with Nulls in C#

---

WORKING WITH NULLABLE VALUE TYPES AND STRINGS



**Jason Roberts**

.NET MVP

@robertsjason    dontcodetired.com



# Overview



Reference and value types overview

Using magic numbers for null value types

Nullable value types with `Nullable<T>`

C# shorthand for nullable value types

Nullable Boolean values

Null, empty, and whitespace strings



# Course Outline

**Working with  
Nullable Value  
Types and Strings**

**Accessing and  
Checking for Null  
Values**

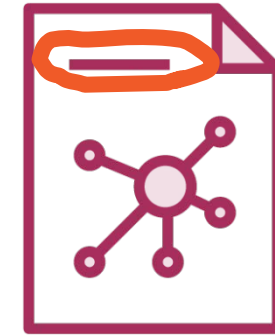
**Eliminating Null  
Reference  
Exceptions**



# Reference and Value Types Overview



Value



<http://dontcodetired.com/ebook.html>

Reference



# Reference and Value Types Overview

## Value

C# struct

Independent instances/copies

Value change doesn't affect other  
copies

The value **is** the information

No reference, cannot be null

No need to check for nulls

## Reference

C# class

Single shared instance

Value change affects people  
(references) pointing to it

The reference **points to** the information

Reference may point to “nothing” (null)

Null checking code may be required



A **value type** may  
sometimes need to  
additionally represent  
a **null value**.

“Magic numbers”

`Nullable<T>`



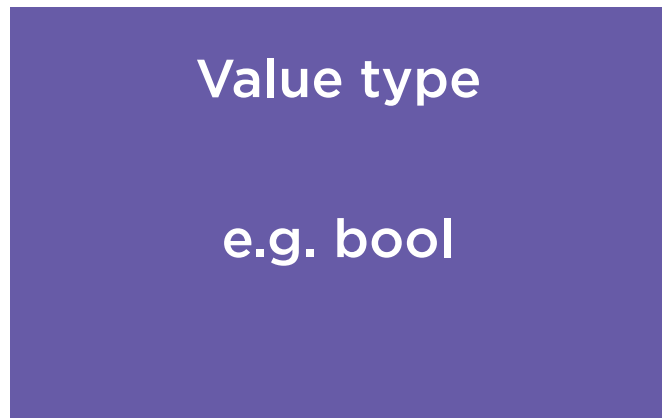
# Nullable Value Types

Nullable types are instances of the **System.Nullable<T>** struct. A nullable type can represent the correct range of values for its underlying value type, **plus an additional null** value.

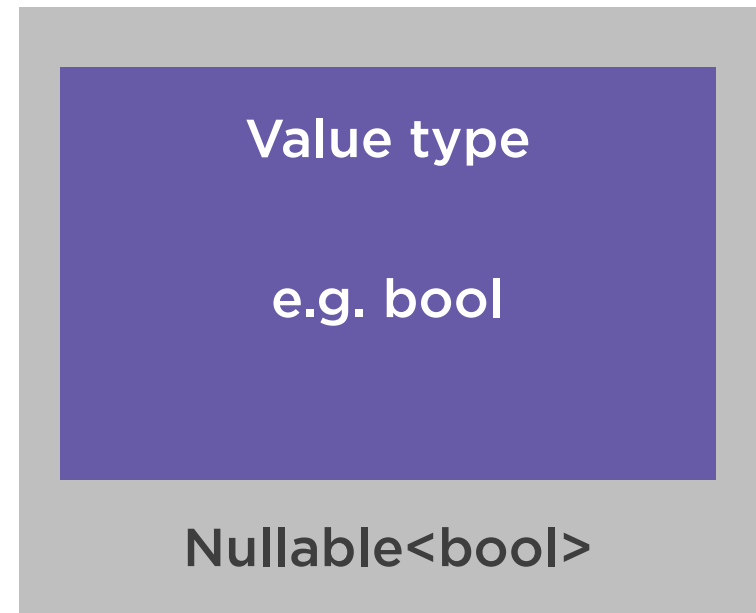
- MSDN



# Introduction to Nullable Value Types with `Nullable<T>`



True  
False



True  
False  
null





# Null and Empty Strings

```
string name = "Sarah"; // reference type  
string name = null;  
string name = ""; // empty string  
string name = "   "; // whitespace string  
  
if (name == null) {...}  
if (string.IsNullOrEmpty(name)) {...}  
if (string.IsNullOrWhiteSpace(name)) {...}
```



# Summary



Reference and value types overview

Value types independent copies

References point to instance

Using magic numbers for null value types

-1 & `DateTime.MinValue`

Nullable value types with `Nullable<T>`

`Nullable<int>` & `Nullable<DateTime>`

C# shorthand: `int?` & `DateTime?`

Nullable Boolean values

`string.IsNullOrEmpty(player.Name)`



Next:

Accessing and Checking for Null Values

