

# Using Kotlin's Nullability Constraints to Improve Code

---



**Kevin Jones**

@kevinrjones [www.rocksolidknowledge.com](http://www.rocksolidknowledge.com)



# Nullability

Java often throws `NullPointerException`

Need lots of defensive code

Kotlin supports 'nullable' types

Means only explicit variables can be null



```
boolean closeMeeting(Meeting meeting) {  
    if (meeting.canClose) return meeting.close();  
    return false;  
}
```

Is this function safe?

**What happens if 'null' is passed?**

**Can null be passed?**



```
fun closeMeeting(m: Meeting): Boolean {  
    return if (m.canClose) m.close()  
        else false  
}
```

Is this function safe?

**Can null be passed?**

**Do we need null checks**



```
closeMeeting(null)
```

```
Error:(7, 18) Kotlin: Null can not be a value of a non-null  
type Meeting
```

Try Calling With Null

**Code does not compile**



What if you  
want to allow  
null?

**Variable can be declared to accept null**

**Put '?' after type name**

**val m: Meeting?**



```
val m:Meeting? = null  
m.close()
```

Error:(6, 6) Kotlin: Only safe (?.) or non-null asserted (!!.) calls are allowed on a nullable receiver of type Meeting?

Must recognise the nullability

Use '?' when using the variable



```
val m:Meeting? = null  
val newMeeting = Meeting()  
newMeeting = m;
```

Error:(7, 18) Kotlin: Type mismatch: inferred type is Meeting? but Meeting was expected

# Assignment

**Cannot assign to non-null type**





## Safe Call (?.)

`m?.method()`

Checks for null

**Calls function only if non-null reference**



Elvis (?:)

`newMeeting = m ?: Meeting()`

Also known as the null-coalescing operator

Returns value or null



Safe cast (as?)

**val saveable = o as? ISaveable**  
**Casts to type or returns null**



## Not-Null assertions (!!)

`m!!.close()`

Very blunt instrument

Asserts that something is not null

Throws NPE if it is

Throws at line operator is used (not later)

It's meant to stand out!!



# Demo



## Null checks



'let' function

Used to avoid explicit null checks

Useful when passing Nullable values ...

... to functions expecting non-null values



# Demo



## 'let' function



## Late initialized properties

**Sometimes values cannot be initialized  
when declared**

**Do not want to mark them as Nullable**

**Use 'lateinit' instead**





# Demo



Trying to pass null



# Summary



**Kotlin provides excellent support for handling nulls**

**Better and safer than Java**

