

Nulls

Every type has an equivalent nullable type, denoted by a trailing `?`

```
val ns: String? = null
val possiblyNullStrings: List<String?> = listOf("apple", "pear",
null, "orange")
```

We can process potentially null objects using a safe-call, but this gives a potentially null result.

```
fun processPotentiallyNull(s: String?): String? = s?.uppercase()
```

If we want to make sure we have a non-null result, we can combine safe-call with the Elvis operator to give a default (non-null) result.

```
fun processPotentiallyNullWithElvis(s: String?): String =
s?.uppercase() ?: ""
```