

SmartCast

available to non available

```
fun smartCast(n: Number?) {  
    requireNotNull(n)  
    println(n.toDouble())  
}
```



after the check n has the
type Number



after the check n has the
type Number

Smart Cast

nullable to non nullable

after the check n has the
type Number

```
fun smartCast(n: Number?) {  
    requireNotNull(n)  
    println(n.toDouble())  
}
```

after the check n has the
type Number

Smart Cast

properties

```
data class Star(val age: Number)

fun smartCast(star: Star) {
    when (star.age) {
        is Int -> println(1..star.age)
        is Double -> print(star.age.isFinite())
    }
}
```