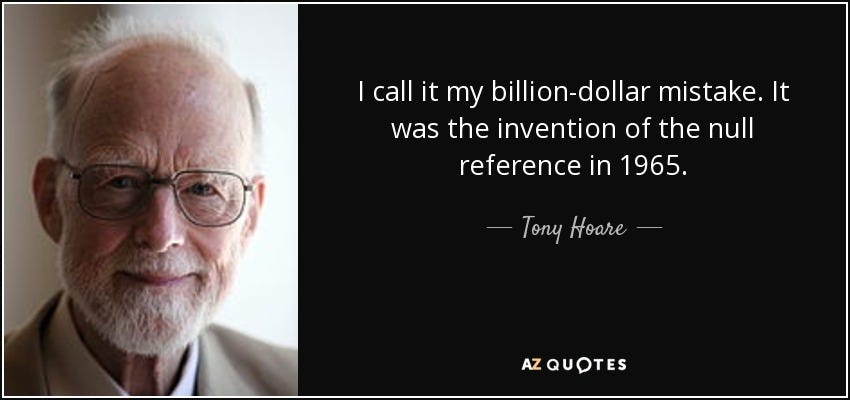
# Annexe 4B – Nullabilité de Kotlin



Tony Hoare est un programmeur des années ’60, il a développé le langage ALGOL W, langage où on a pour la première fois retrouvé le mot-clé while. Ce langage est devenu une base pour le langage Pascal.

Il devait aussi inventer les types des références de ce langage orienté-objet; il voulait au départ que l’usage soit 100% sécure mais il a finalement décidé de permettre que des références soient nulles car c’était simple à mettre en œuvre.

Cette décision, prise par de nombreux autres langages par la suite a mené à des milliards d’erreurs, de krashs, de vulnérabilités et de souffrance depuis 60 ans…

Kotlin veut éviter à tout prix qu’on fasse des NullPointerExceptions qui ferait planter le programme. Essayons néanmoins de le faire planter !

Partons du code suivant :

fun main(args: Array<String>)  
{  
var mot:String  
*println* ( mot.length) 🡪 en Java, ça ferait un NullPointerException   
}

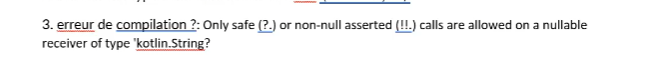
1.erreur de compilation  ?: la varriable n’est pas initialiser

2. initialisons la variable par null

2. erreur de compilation ? : null n’est pas accepté comme valeur de type string (non null)

3. Utilisons un type admettant les valeurs null ( String? )

3. erreur de compilation ?: Cliquez ou appuyez ici pour entrer du texte.



4. Utilisons le « call safe »

4. erreur de compilation ? : non mais retourne null pas reussi a fair plante le code

5. Nous voudrions que le length retourne 0 plutôt que null. Essayons l’opérateur elvis

5. erreur de compilation ? non on y est arrivé

fun main(args: Array<String>)

{

var mot:String? = null

println ( mot?.length?:0)→ affiche 0 si mot vaut nul

}

6. Essayons de lancer une exception NullPointerException avec l’opérateur Elvis

Erreur de compilation ? non

7. Ouin mais on a triché un peu…essayons plutôt l’opérateur d’assertion non-null !!(on assume que la variable n’est pas null)

7. réussi à planter ?

8. Des ClassCastExceptions peuvent être lancées si on essaie de transtyper une String en un Int. Tentons de faire planter ce code :

var mot:String = "abc"  
var nombre:Int? = mot as? Int //opperateur de transtypage securitair  
*println*(nombre)

8. réussi à faire planter ? non

9. enlevons le transtypage sécuritaire ( as )

9. Réussi à le faire planter ?

On peut faire ça : val listeAvecDesNull: List<Int?> = listOf(1, 2, null, 4)

On peut utiliser une fonction let :

val listeAvecDesNull: List<String?> = *listOf*("hugo", "loic", null, "eddy")  
for (item in listeAvecDesNull) {  
 item?.*let* **{** *println*(item.length) **}**}

Une image contenant texte, capture d’écran, logiciel

Le contenu généré par l’IA peut être incorrect.