***Null Safe Operators***

***Assigining null values in Kotlin.***1.Assign null values in kotlin for Type inference is allowed.

***var username=null  
var password=null  
fun main() {  
 var email=null  
 var address=null  
 println("username = ${username}")  
 println("Passowrd = ${password}")  
 println("Email = ${email}")  
 println("Address = ${address}")  
}***

2. Assgin null values in kotlin for datatypes mentioned is Invalid.

***var username:String=null  
var password:String=null  
fun main() {  
 var email:String=null  
 var address:String=null  
 println("username = ${username}")  
 println("Passowrd = ${password}")  
 println("Email = ${email}")  
 println("Address = ${address}")  
}  
/\*\*  
e: D:\Delete\KotlinPractice\src\main\kotlin\main.kt: (1, 21): Null can not be a value of a non-null type String  
e: D:\Delete\KotlinPractice\src\main\kotlin\main.kt: (2, 21): Null can not be a value of a non-null type String  
e: D:\Delete\KotlinPractice\src\main\kotlin\main.kt: (4, 22): Null can not be a value of a non-null type String  
e: D:\Delete\KotlinPractice\src\main\kotlin\main.kt: (5, 24): Null can not be a value of a non-null type String  
 \*/***

***Nullable Operators(?)***

In Kotiln we cannot assign null values to variables with DataType.

using nullable property for assigining null values in Kotiln

***var username:String?=null  
var password:String?=null  
fun main() {  
 var email:String?=null  
 var address:String?=null  
 println("username = ${username}")  
 println("Passowrd = ${password}")  
 println("Email = ${email}")  
 println("Address = ${address}")  
}  
/\*\*  
Output  
username = null  
Passowrd = null  
Email = null  
Address = null  
 \*/***

***Error Case***

***var username:String=null  
var password:String=null  
fun main() {  
 var email:String=null  
 var address:String=null  
 println("username = ${username}")  
 println("Passowrd = ${password}")  
 println("Email = ${email}")  
 println("Address = ${address}")  
}  
/\*  
e: D:\Delete\KotlinPractice\src\main\kotlin\main.kt: (1, 21): Null can not be a value of a non-null type String  
e: D:\Delete\KotlinPractice\src\main\kotlin\main.kt: (2, 21): Null can not be a value of a non-null type String  
e: D:\Delete\KotlinPractice\src\main\kotlin\main.kt: (4, 22): Null can not be a value of a non-null type String  
e: D:\Delete\KotlinPractice\src\main\kotlin\main.kt: (5, 24): Null can not be a value of a non-null type String  
\*/***

***SafeCall(?.)***

1. Return the value if its not null else return null value.
2. Should use it, when the user is OK to show null as output.

var *username*:String?=null  
var *password*:Int?=123455  
fun main() {  
 var email:String?=null  
 var address:String?="ABC State India"  
 *println*("UserName= ${*username*}")  
 *println*("Email= ${email}")  
  
 *println*("Password= ${*password*}")  
 *println*("Address= ${address}")  
}  
  
/\*  
UserName= null  
Email= null  
Password= 123455  
Address= ABC State India  
\*/

***SafeCall with let (?.let)***

1. It Executes the blocks if the value is not null.
2. If the value is null, It wont execute the block

var *username*: String? = null  
var *password*: Int? = 123455  
fun main() {  
 var email: String? = null  
 var address: String? = "ABC State India"  
  
 *username*?.*let* **{** *println*("Username= ${*username*}")  
 **}** email?.*let* **{** *println*("Email = ${email}")  
 **}** *password*?.*let* **{** *println*("pasword= ${*password*}")  
 **}** address?.*let* **{** *println*("Address= ${address}")  
 **}**}  
/\*  
OutPut  
pasword= 123455  
Address= ABC State India  
\*/

Without Safe Call Test case where null values are printed.

var *username*: String? = null  
var *password*: Int? = 123455  
fun main() {  
 var email: String? = null  
 var address: String? = "ABC State India"  
  
 *username*.*let* **{** *println*("Username= ${*username*}")  
 **}** email.*let* **{** *println*("Email = ${email}")  
 **}** *password*.*let* **{** *println*("pasword= ${*password*}")  
 **}** address.*let* **{** *println*("Address= ${address}")  
 **}**}  
  
/\*  
Output  
  
Username= null  
Email = null  
pasword= 123455  
Address= ABC State India   
\*/

***Elvis Operator(?:)***

Returns its first operand if that operand evaluates to a true value, and otherwise evaluates and returns its second operand

var *username*: String? = null  
var *password*: String? = "123455"  
fun main() {  
 var email: String? = null  
 var address: String? = "ABC State India"  
  
 val usname=*username*?.length?:-1  
 val pswrd=*password*?.length?:-1  
 val emil=email?.length?:-1  
 val addrs=address?.length?:-1  
 *println*("Username= ${usname}")  
 *println*("Password= ${pswrd}")  
 *println*("Email= ${emil}")  
 *println*("Address= ${addrs}")  
}  
  
/\*  
Output  
Username= -1  
Password= 6  
Email= -1  
Address= 15   
\*/

***Non-null Assertion Operator (!!)***

Should use when the value is compulsary is not null. Else will get a exception at runtime

var *password*: String? = "123455"  
fun main() {  
 var address: String? = "ABC State India"  
 *println*("Password= ${*password*!!.length}")  
 *println*("Address= ${address!!.length}")  
}

Error Test Case

var *username*: String? = null  
var *password*: String? = "123455"  
fun main() {  
 var email: String? = null  
 var address: String? = "ABC State India"  
 *println*("Username= ${*username*!!.length}")  
 *println*("Password= ${*password*!!.length}")  
 *println*("Email= ${email!!.length}")  
 *println*("Address= ${address!!.length}")  
}  
/\*  
OutPut  
Exception in thread "main" java.lang.NullPointerException  
at MainKt.main(main.kt:7)  
at MainKt.main(main.kt)  
\*/