val

***val (Immutable)***  
1. We cannot change the value of variable which is declared using val keyword

fun main(args: Array<String>) {  
 val name:String="Vinay T Shetty"  
 */\*\*  
 \* Changing the value of the variable  
 \*/* name="Lakshmi"  
 *println*("Name=${name}")  
}  
*/\*\*  
 \* Val cannot be reassigned  
 \*/*

2. Assignment of the variable later.

fun main(args: Array<String>) {  
 val name:String  
 */\*\*  
 \* Assigining Later is valid, only if the type of the variable is mentioned while assigining.Its applicable for local variable only.  
 \*/* name="Vinay T Shetty"  
 *println*("Name=${name}")  
}

Error Case if the data type is not mentioned

fun main(args: Array<String>) {  
 val name  
 */\*\*  
 \* Assigining Later is Invalid,if the type is data type is not mentioned  
 \*/* name="Vinay T Shetty"  
 *println*("Name=${name}")  
}  
  
*/\*\*  
 \* This variable must either have a type annotation or be initialized  
 \*/*

3. Val cannot be re-assigned.

fun main(args: Array<String>) {  
 val name="Vinay T Shetty"  
 name="Vinay"  
}  
*/\*\*\*  
 \* Error :- Val cannot be reassigned  
 \*/*

Instance Variables Test Case  
  
1. Val variables Re-Assignement is not valid.

class Test{  
 val userName:String="Vinay T Shetty"  
 fun demo(){  
 userName="Vinay"  
 *println*("${userName}")  
 }  
}  
fun main(args: Array<String>) {  
 val test=Test()  
 test.demo()  
}  
*/\*\*  
 \* Val cannot be reassigned  
 \*/*

2. Assignment of the variable Later is invalid.Intialization is compulsary for instance variables in case.

class Test{  
 val userName:String  
 fun demo(){  
 userName="Vinay"  
 *println*("${userName}")  
 }  
}  
fun main(args: Array<String>) {  
 val test=Test()  
 test.demo()  
}  
*/\*\*  
 \* Property must be initialized or be abstract  
 \*/*

Demo