Variable

Variable declaration in Kotlin can be done in 2 ways

1. Val
2. Var

We don't require specifying the type of variable explicitly  
Kotlin complier knows this by initilizer expression ("Vinay T Shetty" is a String and 28 is an Int value).

This is called type inference in programming

fun main(args: Array<String>) {  
 val age=28  
 var name="Vinay T Shetty"  
 *println*("Name=${name}\nage=${age}")  
}

fun main(args: Array<String>) {  
 val age:Int=28  
 var name:String="Vinay T Shetty"  
 *println*("Name=${name}\nage=${age}")  
}

General Cases  
Property Initialization.  
  
Local Variable.  
Property initialization can be done in later before using Local Variable.

fun main() {  
 var username:String  
 username="Vinay T Shetty"  
}

Type Declaration is Mandatory

fun main() {  
 var username  
 username="Vinay T Shetty"  
}  
*/\*\*  
 \* This variable must either have a type annotation or be initialized  
 \*/*

Top Level Variable.  
  
Property initialization is compulsary.

var username:String  
fun main() {  
 *username*="Vinay T Shetty"  
}  
*/\*\*  
Output  
Property must be initialized  
 \*/*

Property must be initialized even if the data type is not mentioned.

var username  
fun main() {  
 *username*="Vinay T Shetty"  
}  
/\*  
Output  
Property must be initialized  
\*/

Reason for Declaring the variable type

fun main() {  
 var username  
 username="vinay"  
 username=5  
}  
/\*  
Data Type must have a type if not ,It will declared in different data types.  
\*/  
  
/\*  
Error  
This variable must either have a type annotation or be initialized  
\*/

Variable Assignment Logic in Kotlin.  
  
1.Variable property initialization is compulsary in Kotlin.If not it will be initialized into different datatypes.

var username  
fun main(){  
 *username*="Vinay"  
 *username*=100  
 *println*()  
}  
/\*  
 Property must be initialized  
\*/

fun main(){  
 var username  
 username="Vinay"  
 username=100  
 *println*()  
}  
/\*  
 This variable must either have a type annotation or be initialized  
\*/