***Lazy***

lazy  
  
Lazy initialization is useed to prevent unnecessary initialization of the code.  
It will be initialized only when the variable is used.

val *pi*: Double by *lazy* **{** 3.14  
**}**fun main(){  
 *println*("The value of Area= ${*pi*\*9\*9}")  
 *println*("The Rectangle Area of Area= ${*pi*\*9\*9}")  
}

Additional Information.  
  
A variable with lazy initialization will not be initialized until it’s called or used. This way, the variable is initialized only once and then its value is cached for further use in the program.

Since a property initialized with lazy delegation is supposed to use the same value throughout, it is immutable in nature and is generally used for read-only properties. You must mark it with a val declaration

It is thread-safe, i.e. computed only once and shared by all threads by default. Once initialized, it remembers or caches the initialized value throughout the program  
  
<https://blog.logrocket.com/initializing-lazy-lateinit-variables-kotlin/#:~:text=A%20variable%20with%20lazy%20initialization,further%20use%20in%20the%20program>.