Scala Singleton and Companion Object

Scala Singleton Object

Singleton object is an object which is declared by using object keyword instead by class. No object is required to call methods declared inside singleton object.

In scala, there is no static concept. So scala creates a singleton object to provide entry point for your program execution.

If you don't create singleton object, your code will compile successfully but will not produce any output. Methods declared inside Singleton Object are accessible globally. A singleton object can extend classes and traits.

Scala Singleton Object Example

```
object Singleton{

   def main(args:Array[String]){

      SingletonObject.hello()       // No need to create object.

   }

}


object SingletonObject{

   def hello(){

      println("Hello, This is Singleton Object")

   }

}
```

Output:

```
Hello, This is Singleton Object
```

Scala Companion Object

In scala, when you have a class with same name as singleton object, it is called companion class and the singleton object is called companion object.

The companion class and its companion object both must be defined in the same source file.

Scala Companion Object Example

```
class ComapanionClass{
   def hello(){
      println("Hello, this is Companion Class.")
   }
}
object CompanoinObject{
   def main(args:Array[String]){
      new ComapanionClass().hello()
      println("And this is Companion Object.")
   }
}
```

Output:

```
Hello, this is Companion Class.
And this is Companion Object.
```