Scala Final

Final is a keyword, which is used to prevent inheritance of super class members into derived class. You can declare final variables, methods and classes also.

Scala Final Variable Example

You can't override final variables in subclass. Let's see an example.

```
class Vehicle{
    final val speed:Int = 60
}
class Bike extends Vehicle{
  override val speed:Int = 100
   def show(){
      println(speed)
   }
}
object MainObject{
   def main(args:Array[String]){
      var b = new Bike()
      b.show()
   }
}
```

Output:

Error - value speed cannot override final member

Scala Final Method

Final method declare in the parent class can't be override. You can make any method to final if you don't want to get it overridden. Attempt to override final method will cause to a

compile time error.

Scala Final Method Example

```
class Vehicle{
    final def show(){
        println("vehicle is running")
    }
}
class Bike extends Vehicle{
  //override val speed:Int = 100
   override def show(){
       println("bike is running")
   }
}
object MainObject{
   def main(args:Array[String]){
       var b = new Bike()
       b.show()
   }
}
```

Output:

```
method show cannot override final member
   override def show(){
              ^
one error found
```

Scala Final Class Example

You can also make final class. Final class can't be inherited. If you make a class final, it can't be extended further.

```
final class Vehicle{
    def show(){
        println("vehicle is running")
    }

}

class Bike extends Vehicle{
      override def show(){
       println("bike is running")
   }
}

object MainObject{
   def main(args:Array[String]){
       var b = new Bike()
       b.show()
   }
}
```

Output:

```
error: illegal inheritance from final class Vehicle
class Bike extends Vehicle{
```

^

one error found