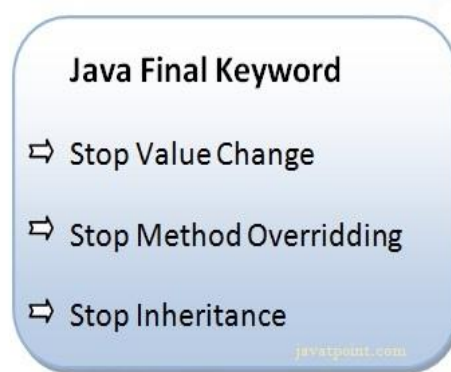


FINAL KEYWORD IN JAVA

The **final keyword** in Java is used to restrict the user. The final keyword can be used in many contexts. The final can be:

1. variable
2. method
3. class

The final keyword can be applied to the variables, a final variable that has no value is called a blank final variable or uninitialized final variable. It can be initialized in the constructor only. The blank final variable can be static also which will be initialized in the static block only.



1) FINAL VARIABLE

If you make any variable as final, you cannot change the value of final variable (It will be constant).

EXAMPLE OF FINAL VARIABLE

There is a final variable speed limit, we are going to change the value of this variable, but It can't be changed because final variable once assigned a value can never be changed.

```
class FinalVar{  
  
    final int speedlimit=90;//final variable  
    void run(){  
        speedlimit=400;  
    }  
    public static void main(String args[]){  
        FinalVar obj=new FinalVar();  
        obj.run();  
    }  
}
```

```
}

C:\Windows\System32\cmd.exe
F:\Java Code 2020>javac FinalVar.java
FinalVar.java:5: error: cannot assign a value
to final variable speedlimit
    speedlimit=400;
    ^
1 error
```

2) FINAL METHOD

If you make any method as final, you cannot override it.

EXAMPLE OF THE FINAL METHOD

```
class FinalMethod
{
    final void run()
    {
        System.out.println("running");
    }
}
class FinalTest extends FinalMethod {
    void run()
    {
        System.out.println("running safely with 100kmph");
    }

    public static void main(String args[]){
        FinalTest f= new FinalTest();
        f.run();
    }
}
```

```
F:\Java Code 2020>javac FinalTest.java
FinalTest.java:10: error: run() in FinalTest
cannot override run() in FinalMethod
        void run()
            ^
    overridden method is final
1 error
```

3) FINAL CLASS

If you make any class as final, you cannot extend it.

EXAMPLE OF FINAL CLASS

```
1. final class Bike{
2.
3. } //base
4.
5. class Honda extends Bike{
6.     void run(){System.out.println("running safely with 100kmph");}
7.
8.     public static void main(String args[]){
9.         Honda honda= new Honda();
10.        honda.run();
11.    }
12. }
```

Output:Compile Time Error

Ques: Can we inherit the final method???

Ans) Yes, final method is inherited but you cannot override it. For Example:

```
1. class Bike{
2.     final void run(){System.out.println("running...");}
3. }
4. class Honda extends Bike{
5.     public static void main(String args[]){
6.         new Honda().run();
7.     }
8. }
```

Output:running...