Imports

By default imported packages

A. java.lang._
B. scala._
C. scala.Predef

1. import <package_name>._
   import evety thing from that package
   java, <package_name>.*

```
C:\Users\Vinit>scala
Welcome to Scala version 2.10.5 (Java HotSpot(TM) 64-Bit Server VM, Java
1.8.0_151).
Type in expressions to have them evaluated.
Type :help for more information.

scala>  val hashSet1 = new HashSet()
<console>:7: error: not found: type HashSet
        val hashSet1 = new HashSet()
                           ^

scala> import java.util._
import java.util._

scala>  val hashSet1 = new HashSet()
hashSet1: java.util.HashSet[Nothing] = []

scala> val dat1 = new Date()
dat1: java.util.Date = Sun Apr 05 10:49:02 SGT 2020
```

2. import <package_name>.<Given Class/trait/interface>
   imports only given class/trait/interface from that package

   Eg: import java.util.HashSet

```
C:\Users\Vinit>scala
Welcome to Scala version 2.10.5 (Java HotSpot(TM) 64-Bit Server VM, Java
1.8.0_151).
Type in expressions to have them evaluated.
Type :help for more information.

scala>  val hashSet1 = new HashSet()
<console>:7: error: not found: type HashSet
        val hashSet1 = new HashSet()
                           ^
```

```
scala> import java.util.HashSet     // Imported HashSet
import java.util.HashSet

scala>  val hashSet1 = new HashSet()
hashSet1: java.util.HashSet[Nothing] = []

scala> val dat1 = new Date()        // Because we only imported HashSet
<console>:8: error: not found: type Date
       val dat1 = new Date()
                      ^
```

3. Import multiple classes
   import <package_name>.{<Given Class/trait/interface> , .. ,.. ,.. }
   imports multiple classes/trait/interface from that package

   Eg: import java.util.{HashSet, Date}

```
C:\Users\Vinit>scala
Welcome to Scala version 2.10.5 (Java HotSpot(TM) 64-Bit Server VM,
Java 1.8.0_151).
Type in expressions to have them evaluated.
Type :help for more information.

scala>  val hashSet1 = new HashSet()
<console>:7: error: not found: type HashSet
        val hashSet1 = new HashSet()
                           ^

scala> val dat1 = new Date()
<console>:7: error: not found: type Date
       val dat1 = new Date()
                      ^

scala> import java.util.{HashSet, Date}
import java.util.{HashSet, Date}

scala>  val hashSet1 = new HashSet()
hashSet1: java.util.HashSet[Nothing] = []

scala> val dat1 = new Date()
dat1: java.util.Date = Sun Apr 05 11:08:06 SGT 2020

scala> val cur = new Currency()
<console>:8: error: not found: type Currency
       val cur = new Currency()
                     ^
```

4. Rename/alias an imported Class
   import <packagename>.{<class Name> => alias}

import java.util.{Date=>UtilDate}
import java.sql.{Date=>SqlDate}

Note: If the same class Date is present in both the packages java.util and java.sql, scala will pick the Date class from the package that was imported at the end.

```
scala> import java.util.Date
import java.util.Date

scala> import java.sql.Date     //java.sql.Date was imported at
end
import java.sql.Date

scala> val date1 = new Date()    // if you just say Date, it will
refer to package that was imported at the end
<console>:13: error: overloaded method constructor Date with
alternatives:
  (x$1: Long)java.sql.Date <and>
  (x$1: Int,x$2: Int,x$3: Int)java.sql.Date
 cannot be applied to ()
       val date1 = new Date()
                   ^

scala> val date1 = new Date(20,1,1)    // refers to java.sql.Date,
bcoz java.sql.Date is the package that was imported at the end
warning: there were 1 deprecation warning(s); re-run with -
deprecation for details
date1: java.sql.Date = 1920-02-01


scala> val date1 = new java.util.Date()  //however you can also
specify the entire path/ i.e. with package name.
date1: java.util.Date = Sun Apr 05 11:35:32 SGT 2020
```

Rename or Alias – Practical

```
C:\Users\Vinit>scala
Welcome to Scala version 2.10.5 (Java HotSpot(TM) 64-Bit Server
VM, Java 1.8.0_151).
Type in expressions to have them evaluated.
Type :help for more information.

scala> import java.sql.{Date => SqlDate}  // renamed Date to
SqlDate for java.sql package
```

```
import java.sql.{Date=>SqlDate}

scala> import java.util.{Date => UtilDate}// renamed Date to
UtilDate for java.util package

import java.util.{Date=>UtilDate}

scala> val date = new Date()     //Date is not found because we
renamed it above
<console>:9: error: not found: type Date
       val date = new Date()
                      ^

scala> val date = new UtilDate()
date: java.util.Date = Sun Apr 05 11:41:26 SGT 2020

scala> val date = new SqlDate(20,1,1)
warning: there were 1 deprecation warning(s); re-run with -
deprecation for details
date: java.sql.Date = 1920-02-01
```

Note: in Java, you need to write import statemet at the start of program, however in scala you can write import statement any where in middle of program.

```
// program 1
// In java, you write only on top
// In Scala, mostly you will see import written on top.

import java.util.Date

object DemoAIfElse {
  // main  then press ctrl > space > enter

  def hello() = {
    var date2 = new Date()
    println(date2)
  }

  def main(args: Array[String]): Unit = {
    var date1 = new Date()
    println(date1)
      }
}
// program 2
// In Scala, you can also write import in middle of program. However scope of
import statement gets reduced/limited
```

```
object DemoAIfElse {
  // main  then press ctrl > space > enter

  def hello() = {
    import java.util.Date
    var date2 = new Date()
    println(date2)
  }

  def main(args: Array[String]): Unit = {
    import java.util.Date
    var date1 = new Date()
    println(date1)
      }
}
```