

```
In [1]: "hello world"

res0: String = "hello world"
```

```
In [2]: println("hello, world")

hello, world
```

```
In [3]: println("hi")

hi
```

```
In [4]: val name= "Buddy"

name: String = "Buddy"
```

```
In [4]: name="hey"

Main.scala:25: reassignment to val
name="hey"
  ^
```

```
In [5]: var vname:String="Buddy"

vname: String = "Buddy"
```

```
In [6]: var vin:Int=1

vin: Int = 1
```

```
In [7]: println(name)

Buddy
```

```
In [8]: println(s"Hello, $name") //String interpolation-mixing variable name with

Hello, Buddy
```

```
In [9]: 3+2

res8: Int = 5
```

```
In [10]: 3.+(2)
```

```
res9: Int = 5
```

```
In [13]: 3.2+2
```

```
res12: Double = 5.2
```

```
In [16]: 3.0+(2)
```

```
res15: Double = 5.0
```

```
3.2+(2)
```

```
In [15]: 3.2+(2)
```

```
res14: Double = 5.2
```

```
In [11]: object MyWorld{  
        def main(args:Array[String]){  
            println("Hello,from main!")  
        }  
    }
```

```
defined object MyWorld
```

```
In [12]: MyWorld.main(Array())
```

```
Hello,from main!
```

```
In [17]: MyWorld.main(Array("Great"))
```

```
Hello,from main!
```

```
In [18]: object MySimplerWorld extends App{ //defined in jupyter envt, so have to  
        println("Hello from MySimplerWorld...")  
    }
```

```
defined object MySimplerWorld
```

```
In [19]: MySimplerWorld.main(Array())
```

```
Hello from MySimplerWorld...
```

```
In [20]: object MySimplerWorld extends App{  
    println(args(0))  
}
```

defined object MySimplerWorld

```
In [21]: MySimplerWorld.main(Array("Great"))
```

Great

```
In [22]: object MySimplerWorld extends App{  
    println("Hello "+args(0))  
}
```

defined object MySimplerWorld

```
In [23]: MySimplerWorld.main(Array("Buddy"))
```

Hello Buddy

```
In [24]: class Person(name:String)//equivalent to an empty class
```

defined class Person

```
In [25]: val p=new Person("Appu")
```

p: Person = cmd23\$\$user\$Person@4d5aca0b

```
In [25]: p.name
```

Main.scala:25: value name is not a member of cmd24.INSTANCE.\$ref\$cmd23
.Person
p.name
 ^

```
In [26]: class Person(name:String){//name is part of primary constructor for the c  
    println(name)  
}
```

defined class Person

In [26]: `p.name`

```
Main.scala:25: value name is not a member of cmd24.INSTANCE.$ref$cmd23
.Person
p.name
  ^
```

In [27]: `val p=new Person("Appu")`

```
Appu
p: Person = cmd25$$user$Person@3fd537d
```

In [27]: `p.name`

```
Main.scala:25: value name is not a member of cmd26.INSTANCE.$ref$cmd25
.Person
p.name
  ^
```

In [28]: `class Person(){
 println("Hello")
}`

```
defined class Person
```

In [29]: `val p = new Person()`

```
Hello
p: Person = cmd27$$user$Person@63bea32d
```

In [30]: `class Person(name:String){
 val nameVal=name
}`

```
defined class Person
```

In [31]: `val p=new Person("Buddy")
p.nameVal`

```
p: Person = cmd29$$user$Person@1780df98
res30_1: String = "Buddy"
```

```
In [31]: p.nameVal="hey"
```

```
Main.scala:25: reassignment to val
p.nameVal="hey"
  ^
```

```
In [32]: class Person(name:String){
  var nameVar=name
}
val p=new Person("Buddy")
p.nameVar="Best Buddy"
```

```
defined class Person
p: $user.Person = cmd31$$user$Person@6076dcd4
```

```
In [33]: p.nameVar
```

```
res32: String = "Best Buddy"
```

```
In [34]: p.nameVar="changed"
```

```
In [35]: p.nameVar
```

```
res34: String = "changed"
```

```
In [2]: class GreatNumber(n:Int){
  //require (n>0) //fail to compile
  private var _n:Int = n
  def ^(p:Int) = scala.math.pow(_n,p) // ^ is function name
  def squared() = _n * _n //same as def squared():Int{ return _n * _n }
}
```

```
val c = new GreatNumber(2)
c ^ 3 //same as c.^(3)
c squared //c.squared()
```

```
defined class GreatNumber
c: $user.GreatNumber = cmd1$$user$GreatNumber@6bb936e4
res1_2: Double = 8.0
res1_3: Int = 4
```

```
In [4]: val func = (a:Int, b:Int) => {
        a+b
      }:Int //return type
        func(2,3)

func: (Int, Int) => Int = <function2>
res3_1: Int = 5
```

```
In [8]: def method(a:Int, b:Int):Int ={
        a*b
      }
        method(2,3)

defined function method
res7_1: Int = 6
```

```
In [9]: scala.util.Properties.versionString

res8: String = "version 2.11.7"
```

```
In [8]: val convertedFun : (Int, Int) => Int = method //elaborated syntax //will

Main.scala:25: missing arguments for method method in class $user;
follow this method with `_' if you want to treat it as a partially app
plied function
method //elaborated syntax
^
```

```
In [ ]: val x :Int = 10
        convertedFun(5, 9)
```

```
In [6]: val convFun= method _ //it will fail when a method is returning another m

convFun: (Int, Int) => Int = <function2>
```

```
In [ ]: convFun(5,6)
```

```
In [7]:

res6: Int = 30
```

//ENUMERATIONS

```
In [25]: //Enumerations
object MyEnum extends Enumeration{
  type EnumType=Value
  val MIN,MIN_1 =Value
  val ONE= Value(100)
  val TWO, THREE = Value
  val FOUR = Value(14, "Fourteen")
  val MAX = Value(12)
}

println(MyEnum.TWO)
val x:MyEnum.EnumType = MyEnum(14)
x
```

TWO

```
defined object MyEnum
x: $user.MyEnum.EnumType = Fourteen
res24_3: $user.MyEnum.EnumType = Fourteen
```

```
In [19]: val y:MyEnum.EnumType = MyEnum(5) //error as values starts from 0,1,2....
//starts with 0 because passed an empty constructor
//starts with 100 now as first value is 100
```

```
java.util.NoSuchElementException: key not found: 5 (key not found: 5)
scala.collection.MapLike$class.default(MapLike.scala:228)
scala.collection.AbstractMap.default(Map.scala:59)
scala.collection.mutable.HashMap.apply(HashMap.scala:65)
scala Enumeration.apply(Enumeration.scala:114)
cmd18$$user$$anonfun$1.apply(Main.scala:25)
cmd18$$user$$anonfun$1.apply(Main.scala:24)
```

```
In [15]: println(MyEnum.ONE)
```

ONE

```
In [20]: println(MyEnum.ONE.id)
```

100

```
In [21]: println(MyEnum.MAX.id)
```

12

```
In [22]: println(MyEnum.TWO.id)
```

```
101
```

```
In [26]: println(MyEnum.MIN.id)
println(MyEnum.MIN_1.id)
```

```
0
```

```
1
```

```
In [26]: var greet="hey"
println(greet)
greet="hi"
println(greet)
greet=1
println(greet)
```

```
Main.scala:36: type mismatch;
   found   : Int(1)
   required: String
greet=1
      ^
```

```
In [ ]:
```