

```
In [ ]: //Type Inference
```

```
In [1]: val func = (a:Int, b:Int) => a+b  
  
func: (Int, Int) => Int = <function2>
```

```
In [ ]: //EXPRESSIONS
```

```
In [4]: var x=10  
if(x==15) println("yes") else println("no")  
  
no  
  
x: Int = 10
```

```
In [2]: var x=10  
if(x=15) println("yes") else println("no")  
//Unit means no return value for this  
//doesnt return anything//like void  
  
Main.scala:28: type mismatch;  
found   : Unit  
required: Boolean  
if(x=15) println("yes") else println("no")  
  ^
```

```
In [3]: val x={  
    val y=5  
    y + 20  
    y  
}  
println(x)  
  
5  
  
x: Int = 5
```

```
In [5]: val x={  
    val y=5  
    y + 20  
  
}  
println(x)  
  
25  
  
x: Int = 25
```

```
In [6]: //CONTROLS
```

```
In [7]: if(x>10) x else 10
```

```
res6: Int = 25
```

```
In [8]: if(x>10) {  
    x  
  }  
  else{  
    10  
  }
```

```
res7: Int = 25
```

```
In [9]: if(x>10){  
    x match{  
      case 25 => 1  
      case _ => 0 // _ is wild card. for default  
    }  
  }
```

```
res8: AnyVal = 1
```

```
In [10]: //FOR LOOP
```

```
In [11]: for(i <- 1 to 3)  
println(i)
```

```
1  
2  
3
```

```
In [12]: for(i <- 1 until 5) println(i)
```

```
1  
2  
3  
4
```

```
In [13]: for(i <- 1 to 10
         if i%2 == 0
         if i%3 == 0)
         println(i)
```

6

```
In [14]: for(i <- 1 to 10
         if i%2 == 0)
         println(i)
```

2
4
6
8
10

```
In [15]: for(i <- 1 to 10
         if i%2 == 0)
         yield i
```

res14: collection.immutable.IndexedSeq[Int] = Vector(2, 4, 6, 8, 10)

```
In [16]: for{i <- 1 to 10
         if i%2 == 0}
         yield i           //for apache spark // lazily evaluated
```

res15: collection.immutable.IndexedSeq[Int] = Vector(2, 4, 6, 8, 10)

```
In [18]: for{i <- 1 until 5} println(i)
```

1
2
3
4

```
In [21]: val zy= for(i <- 1 to 10
         if i%2 == 0)
         yield i+1
```

zy: collection.immutable.IndexedSeq[Int] = Vector(3, 5, 7, 9, 11)

```
In [22]: for(i <- 1 to 10  
           if (i%2 == 0 || i%3 ==0)  
           if i%3 == 0)  
           println(i)
```

3

6

9