KOTLIN

Open source

Most of errors are found in compile time
Which is better than other languages which find errors in run time.

Output - -> JVM (java virtual machine)

Print - -> print a statement ;
Println - -> print a statement and move the cursor in next line;

Variables :
var - -> can be reassigned
val - -> can't be reassigned

Data ki type define krone ki zarurat nhi h
Wo apne aap decide kalega
But if u want to tell type of data you can do :

```
Var score : int = 3
Var temp : float = 33.5
Val  number : boolean = true
```

RANGE IN KOTLIN

- -> upper bound is included

```
fun main (){
val num=5
val result = num in 1..5
print (result)
}
```

- -> upper bound is not included

```
fun main (){
val num=5
val result = num in 1 until 5
print (result)
}
```

WHEN STATEMENT
- -> when statement always ends with 'ELSE'

```
fun main (){
```

```
val num=5
val result =when (num){
    1->"one"
    2->"two"
    3->"three"
    4->"four"
    5->"five"
    else->"not in range"
}
println (result)
}
```

FOR LOOP SYNTAX
- -> by default step is 1 .

```
fun main (){
for (I in 1..5 step 2){
println (I)
} }
```

OR

```
fun main (){
for (I in 5 downTo 1 step 2){
println (I)
} }
```

FUNCTIONS SYNTAX

```
fun function_name ( data1 : datatype, data2 : datatype ) : return_datatype
{

// statement

return
}
```

STORING FUNCTION IN VARIABLE

Syntax :

```
var variable_name = ::function_name
```

ARRAYS

```
fun main()
{
    // declaring an array using arrayOf()
    val arrayname = arrayOf(1, 2, 3, 4, 5)
    for (i in 0..arrayname.size-1)
    {
        print(" "+arrayname[i])
    }
    println()
    // declaring an array using arrayOf<Int>
    val arrayname2 = arrayOf<Int>(10, 20, 30, 40, 50)
    for (i in 0..arrayname2.size-1)
    {
        print(" "+arrayname2[i])
    }
}
```

CLASSES AND OBJECTS

```
class employee {
    // properties
    var name: String = ""
    var age: Int = 0
    var gender: Char = 'M'
    var salary: Double = 0.toDouble()

      // member functions
       fun name(){

    }
    fun age() {

    }
    fun salary(){

    }
}
```