Below we look at Major operators used in kotlin

1. Comparison and Equality Operators

Here's a table of equality and comparison operators, their meaning, and corresponding functions:

Operator	Meaning	Expression	Translates to
>	greater than	a > b	a.compareTo(b) > 0
<	less than	a < b	a.compareTo(b) < 0
>=	greater than or equals to	a >= b	a.compareTo(b) >= 0
<=	less than or equals to	a < = b	a.compareTo(b) <= 0
==	is equal to	a == b	a?.equals(b) ?: (b === null)
!=	not equal to	a != b	!(a?.equals(b) ?: (b === null))

Comparison and equality operators are used in control flow such as *if expression*, *when expression*, and *loops*.

Example: Comparison and Equality Operators

```
fun main(args: Array<String>) {

    val a:Int = -12
    val b:Int = 12

    // use of greater than operator
    if (a > b) {
        println("a is larger than b.")

    } else {
        println("b is larger than a.")

    }
}
```

2. Assignment Operators

Assignment operators are used to assign value to a variable. We have already used simple assignment operator **=** before.

```
val age:Int = 5
```

Here, 5 is assigned to variable *age* using **=** operator.

Here's a list of all assignment operators and their corresponding functions:

Expression	Equivalent to	Translates to
a +=b	a = a + b	a.plusAssign(b)
a -= b	a = a - b	a.minusAssign(b)

Expression	Equivalent to	Translates to
a *= b	a = a * b	a.timesAssign(b)
a /= b	a = a / b	a.divAssign(b)
a %= b	a = a % b	a.modAssign(b)

Example: Assignment Operators

```
fun main(args: Array<String>) {
    var number:Int = 12

    number *= 5   // number = number*5
    println("number  = $number")
}
```

When you run the program, the output will be:

```
number = 60
```

3. Logical Operators

There are two logical operators in Kotlin: `||` and &&

Here's a table of logical operators, their meaning, and corresponding functions.

Operator	Description	Expression	Corresponding Function
||	`true` if either of the Boolean expression is `true`	`(a>b)||(a<c)`	`(a>b)or(a<c)`
&&	true if all Boolean expressions are `true`	`(a>b)&&(a<c)`	`(a>b)and(a<c)`

Logical operators are used in control flow such as *if expression*, *when expression*, and *loops*.

Example: Logical Operators

```
fun main(args: Array<String>) {
    val a:Int = 10
    val b:Int = 9
    val c:Int = -1

    // && is a Logical operator, both conditions must be true
    if(a>b && a>c) {
         println("a is large")
    }//end if
}//end main
```

When you run the program, the output will be:

```
a is large
```

4. Arithmetic Operators

Here's a list of arithmetic operators in Kotlin:

Kotlin Arithmetic Operators

Operator	Meaning
+	Addition (also used for string concatenation)
-	Subtraction Operator
*	Multiplication Operator
/	Division Operator
%	Modulus Operator

Example: Arithmetic Operators

```
fun main(args: Array<String>) {

    val number1: Double = 12.5
    val number2: Double  = 3.5
    var result: Double    //this will hold our results later

    result = number1 + number2
    println("Addition is  = $result")

    result = number1 - number2
    println("Subtraction is = $result")

    result = number1 * number2
    println("Product is = $result")

    result = number1 / number2
    println("Division = $result")

    result = number1 % number2
    println("Modulus is = $result")
}
```

When you run the program, the output will be:

```
Addition is = 16.0
Subtraction is  = 9.0
Product is  = 43.75
Division = 3.5714285714285716
Modulus is = 2.0
```

NB: The + operator is also used for the concatenation of `String` values.