

```

import kotlin.random.Random

//funcion normal
fun numeros (numero:Int =0, numero2:Int = 0 ): Int{

    return numero * numero2
}

//compacta
fun cuboNumero(num:Int) = num * num * num

//aninima
val cuadrado = fun(num:Int) = num * num

//lamda
val cubo = {num:Int ->  num * num * num}

//objeto
/* clase normal
class persona{
    var nombre:String = "Aaron"
    var apellido:String = "Gonzalez"
    var edad:Int = 25

    fun decir(){
        println("Mi nombre es $nombre $apellido y mi edad es $edad")
    }
}*/
//constructor primario
//asi ahorramos mas codigo

abstract class localidad(){
    abstract val vivienda: String
    abstract val calle: String
}

class vivo(): localidad(){
    override val calle = "Calle Blas Infante"
    override val vivienda = "Tocina"
}

interface datosVivienda {
    fun metrosCuadrados (id: Int) : Int
    fun colorCasa (id: Int) : String
    fun numeroCasa (id: Int) : Int
}
/*
class casa : datosVivienda{
    override fun colorCasa(id: Int): String {
        TODO("Not yet implemented")
    }
}*/

```

```

abstract class aspectoCoche(){
    abstract val color:Int
    abstract val marca:String
}

interface moverse {
    fun correr()
}

class coche: aspectoCoche(),moverse{
    override val marca = "verde"
    override val color = 5
    override fun correr(){
        println("el coche se mueve a")
    }
}

class persona( var nombre:String = "Aaron", var apellido: String = "Gonzalez", var edad:Int = 25){

    var altura : Int = 0
    set(nuevaAltura){
        altura = nuevaAltura
    }

    val annosPorDos: Int
    get() = edad * 2

    init{
        println("Encantado de hablar contigo")
    }

    fun decir(){
        println("Mi nombre es $nombre $apellido y mi edad es $edad")
    }

    constructor(segundoApellido: String):this(){
        println("mi nombre completo es: $nombre $apellido $segundoApellido")
    }
}

fun main() {
    /* val num1 = 1..10
    val num2 = 5
    var prueba = 0
    var rando = num1.random()
    when(rando){
        in 1..4 -> println("suspenso")
        in 5..6 -> println("ta bien")
        in 7..8 -> println("flama")
        else -> println("perfect")
    }*/

    //var nulo:Int? = null
    /*var nulo:Int = 8
    var resultado = nulo?.toInt() ?: 2
    println(resultado)*/

    /*val datos = arrayOf("pepe","juabn")

    for(x in 5 downTo 1 step 2){

```

```

    println(x)
}*/
//println(numeros(numero2 = 5))

//val lista = listOf("aaron","ivan","daniel","perro","loro")
//val lista = listOf(1,2,5,9,3,5)
//val resultado = lista.filter { it.length >5 || it.length == 5}
//val resultado = lista.filter { it % 2 != 0}

//println(resultado)
//println(resultado.size)
//println(cuadrado(2))
/*
var numero:Int = 5
println("el numero aleatorio dado es: ${ if(numero >= 5) "es mayor o igual a 5" else "es menor que 5"}.")*/

//compacta
//fun suma (x: Int, y: Int) = x + y

//anonima
/*
* la aninima hace meter la funcion en una val y la iguala a la funcion sin nombre el resto es igual*/
val suma = fun (x: Int, y: Int) = x +y

//lammda
val sumaLammda = {x:Int, y:Int ->  x+y}

    when(sumaLammda(1,2)){
    1 -> println("uno")
    2 -> println("dos")
    else -> {
        println("tres")
    }
}

//manejo de null

/*val enigma: Int? = null
val resultado = enigma?.toInt() ?: 0;*/

for(x in 50 downTo 1 step 5 ){
    println(x)
}

println("objeto")

var humano = persona(segundoApellido = "Alvarez")
humano.edad =30
humano.decir()

var numNull:Int? = null

var verNum = numNull?.toInt() ?: 5

println(verNum)

println("Micoche")
var miCoche = coche()

```

```
println(miCoche.color)
println(miCoche.marca)
println(miCoche.correr())
```

```
}
```