

# 17 - Funciones: con una única expresión

Las funciones de una única expresión se pueden expresar en Kotlin sin el bloque de llaves y mediante una asignación indicar el valor que retorna.

Recordemos que uno de los objetivos en Kotlin es permitirnos implementar los algoritmos en la forma más concisa posible.

Resolveremos algunas de las funciones ya planteadas utilizando esta nueva sintaxis.

## Problema 1

Confeccionar una función que le enviemos como parámetro el valor del lado de un cuadrado y nos retorne su superficie.

Proyecto85 - Principal.kt

```
fun retornarSuperficie(lado: Int) = lado * lado

fun main(parametro: Array<String>) {
    print("Ingrese el valor del lado del cuafrado:")
    val la = readln().toInt()
    println("La superficie del cuadrado es ${retornarSuperficie(la)}")
}
```

Como podemos ver la implementación completa de la función es una sola línea:

```
fun retornarSuperficie(lado: Int) = lado * lado
```

Disponemos el operador = y seguidamente la expresión, en este caso el producto del parámetro lado por si mismo.

No hace falta indicar el tipo de dato que retorna la función ya que el compilador puede inferir que del producto lado \* lado se genera un tipo de dato Int.

No hay problema de indicar el tipo de dato a retornar, pero en muchas situaciones el compilador lo puede inferir como es este caso:

```
fun retornarSuperficie(lado: Int): Int = lado * lado
```

La llamada a una función que contiene una única expresión no varía:

```
println("La superficie del cuadrado es ${retornarSuperficie(la)}")
```

## Problema 2

Confeccionar una función que le enviemos como parámetros dos enteros y nos retorne el mayor.

Proyecto86 - Principal.kt

```
fun retornarMayor(v1: Int, v2: Int) = if (v1 > v2) v1 else v2

fun main(parametro: Array<String>) {
    print("Ingrese el primer valor:")
    val valor1 =readln().toInt()
    print("Ingrese el segundo valor:")
    val valor2 =readln().toInt()
    println("La superficie del cuadrado es ${retornarSuperficie(1a)}}")    1
}
```

Teniendo en cuenta que la instrucción if puede disponerse como una expresión como lo vimos anteriormente está permitido su uso en las funciones con una única expresión:

```
fun retornarMayor(v1: Int, v2: Int) = if (v1 > v2) v1 else v2
```

Recordemos que el objetivo de codificar el algoritmo con esta sintaxis es hacer el código lo mas conciso, recordemos que la otra forma de expresar esta función es:

```
fun retornarMayor(v1: Int, v2: Int): Int {
    if (v1 > v2)
        return v1
    else
        return v2
}
```

## Problema 3

Confeccionar una función reciba un entero comprendido entre 1 y 5 y nos retorne en castellano dicho número o un String con la cadena "error" si no está comprendido entre 1 y 5.

Proyecto87 - Principal.kt

```
fun convertirCastelano(valor: Int) = when (valor) {
    1 -> "uno"
    2 -> "dos"
    3 -> "tres"
    4 -> "cuatro"
    5 -> "cinco"
    else -> "error"
}

fun main(parametro: Array<String>) {
    for(i in 1..6)
        println(convertirCastelano(i))
}
```

En este problema mostramos que podemos utilizar la sentencia when como expresión de retorno de la función.

## Problemas propuestos

Utilizar una única expresión en las funciones pedidas en estos problemas

- Elaborar una función que reciba tres enteros y nos retorne el valor promedio de los mismos.
- Elaborar una función que nos retorne el perímetro de un cuadrado pasando como parámetros el valor del lado.
- Confeccionar una función que calcule la superficie de un rectángulo y la retorne, la función recibe como parámetros los valores de dos de sus lados:

```
fun retornarSuperficie(lado1: Int,lado2: Int): Int
```

En la función main del programa cargar los lados de dos rectángulos y luego mostrar cual de los dos tiene una superficie mayor.

- Confeccionar una función que le enviemos como parámetro un String y nos retorne la cantidad de caracteres que tiene. En la función main solicitar la carga de dos nombres por teclado y llamar a la función dos veces. Imprimir en la main cual de las dos palabras tiene más caracteres.

#### Proyecto88

```
fun retornarPromedio(v1: Int, v2: Int, v3: Int) = (v1 + v2 + v3) / 3

fun main(parametro: Array<String>) {
    print("Ingrese primer valor:")
    val valor1 = readln().toInt()
    print("Ingrese segundo valor:")
    val valor2 = readln().toInt()
    print("Ingrese tercer valor:")
    val valor3 = readln().toInt()
    println("Valor promedio de los tres números ingresados es ${retornarPromedio(valor1, valor2, valor3)}")
}
```

#### Proyecto89

```
fun retornarPerimetro(lado: Int) = lado * 4

fun main(parametro: Array<String>) {
    print("Ingrese el lado del cuadrado:")
    val lado = readln().toInt()
    print("El perimetro es: ${retornarPerimetro(lado)}")
}
```

#### Proyecto90

```
fun retornarSuperficie(lado1: Int, lado2: Int) = lado1 * lado2

fun main(parametro: Array<String>) {
    println("Primer rectangulo")
    print("Ingrese lado menor del rectangulo:")
    val lado1 = readln().toInt()
    print("Ingrese lado mayor del rectangulo:")
    val lado2 = readln().toInt()
    println("Segundo rectangulo")
    print("Ingrese lado menor del rectangulo:")
    val lado3 = readln().toInt()
    print("Ingrese lado mayor del rectangulo:")
    val lado4 = readln().toInt()
    if (retornarSuperficie(lado1, lado2) == retornarSuperficie(lado3, lado4))
        print("Los dos rectangulos tiene la misma superficie")
    else
        if (retornarSuperficie(lado1, lado2) > retornarSuperficie(lado3, lado4))
            print("El primer rectangulo tiene una superficie mayor")
        else
            print("El segundo rectangulo tiene una superficie mayor")
}
```

#### Proyecto91

```
fun largo(nombre: String) = nombre.length

fun main(parametro: Array<String>) {
    print("Ingrese un nombre:")
    val nombre1 = readln()
    print("Ingrese otro nombre:")
    val nombre2 = readln()
    if (largo(nombre1) == largo(nombre2))
        print("Los nombres: $nombre1 y $nombre2 tienen la misma cantidad de caracteres")
    else

```

```
if (largo(nombre1) > largo(nombre2))  
    print("$nombre1 es mas largo")  
else  
    print("$nombre2 es mas largo")  
}
```