

## Sobreescribir toString()

Entenderemos con detalle que significa *override* al estudiar herencia. Manejamos ahora intuitivamente este concepto para utilizar la función `toString()` ya desde nuestros primeros ejercicios de clases con kotlin.

`toString()` es una función miembro cuyo objetivo es ofrecer una representación textual del estado de un objeto. Si sobreescribimos `toString()` podemos personalizar dicha representación textual. Debemos escribir la función `toString()` con el modificador *override* para tener una representación textual del objeto a nuestro gusto.

```
In [1]: class Rectangle(val width: Int, val height: Int) {  
        val area: Int // property type is optional since it can be inferred  
        get() = this.width * this.height  
  
        override fun toString():String{  
            return "ancho: $width alto: $height area: $area"  
        }  
    }  
    val mirectangulo= Rectangle(2,3)  
    println(mirectangulo.area)  
    println(mirectangulo)  
  
6  
ancho: 2 alto: 3 area: 6
```