

## Kotlin – Single Expression Functions

Imagine que você está desenvolvendo um software relacionado a vendas, e, no momento do checkout, precisa aplicar um desconto em cima do valor de final da compra do cliente. Fácil? Mas é claro que, você como desenvolvedor sabe que os requisitos do seu cliente estão sempre mudando, então para garantir que a menor quantidade possível de modificações serão feitas no decorrer do processo, você resolve desenvolver um método para realizar esse cálculo, que em Java teria uma sintaxe provavelmente similar a essa:

```
public double calcularValorFinal(double total, double desconto) {  
    return total - (total * desconto);  
}
```

Se transcrevermos esse método para Kotlin melhora um pouco a sintaxe:

```
fun calcularValorFinal(total: Double, desconto: Double): Double {  
    return total - (total * desconto)  
}
```

Mas mesmo assim algo não parece estar certo! E então, qual o problema? A questão é que você precisou escrever muito código para satisfazer o compilador, e pouco código para resolver o problema que você realmente queria resolver. A única parte que realmente importa são os parâmetros e o valor retornado, então por que escrever mais do que isso, certo? É aí que as Single Expression Functions entram em ação, veja:

```
fun calcularValorFinal(total: Double, desconto: Double) = total -  
    (total * desconto)
```

Transcrevemos várias linhas de código em Java para só uma em Kotlin! Mas o que nós acabamos de fazer? Nós definimos uma função que, por realizar uma operação simples não existia necessidade do uso de chaves para o corpo, e melhor ainda, omitimos o retorno do tipo Double, uma vez que o compilador é esperto o bastante para descobrir isso por sozinho.

Por Hoje é só pessoal, sucesso nos estudos.