

## When Expression

Além do If Expression, o Kotlin também disponibiliza outra feature para controle de fluxo conhecida como [When Expression](https://kotlinlang.org/docs/reference/control-flow.html#when-expression) (<https://kotlinlang.org/docs/reference/control-flow.html#when-expression>). Basicamente é uma estrutura que substitui o famoso Switch Case muito comum em diversas linguagens.

Considerando a implementação da função para pegar uma cor:

```
private fun corPor(tipo: Tipo): Int {  
    if (tipo == Tipo.RECEITA) {  
        return ContextCompat.getColor(context, R.color.receita)  
    }  
    return ContextCompat.getColor(context, R.color.despesa)  
}
```

Convertendo o mesmo código com o When Expression, temos o seguinte resultado:

```
private fun corPor(tipo: Tipo): Int {  
    return when (tipo) {  
        Tipo.RECEITA -> ContextCompat.getColor(context, R.color.receita)  
        else -> ContextCompat.getColor(context, R.color.despesa)  
    }  
}
```

Inclusive, podemos ser mais próximos de um Switch Case:

```
private fun corPor(tipo: Tipo): Int {  
    return when (tipo) {  
        Tipo.RECEITA -> ContextCompat.getColor(context, R.color.receita)  
        Tipo.DESPESA -> ContextCompat.getColor(context, R.color.despesa)  
    }  
}
```

Perceba que fica bem explícito o retorno para cada tipo possível de transação. Existem outras técnicas possíveis com o When Expression, entretanto, não veremos aqui no curso justamente por ele aumentar a complexidade no código em um único ponto tendendo a aumentar cada vez mais, algo similar ao que vemos no Switch Case.

Em outras palavras, fique à vontade em decidir qual implementação deseja aplicar no projeto, pois o resultado será o mesmo :)

Claro, para outros exemplos, recomendo fortemente que dê uma lida na documentação, principalmente na sessão relacionada ao [Sealed Class](https://kotlinlang.org/docs/reference/sealed-classes.html) (<https://kotlinlang.org/docs/reference/sealed-classes.html>) que combina suas características com o When Expression.

