

Para saber mais: Pontos de atenção em scope functions

Mesmo vendo as diferenças nas funções de escopo, é natural, em algum momento do código, não utilizar a função correta. Sendo assim, seguem mais alguns tópicos que podem ajudar a identificar os pontos:

- Chama um ou mais membros do objeto de contexto -> Utilize o objeto de contexto receiver.
- Não chama membros do objeto do contexto -> Utilize o objeto de contexto como argumento.
- Não faz a configuração do objeto, inicializações ou chamadas de membros -> Utilize `let()` ou `also()` .

Além disso, também existe a boa prática no uso de funções de escopo descrita na documentação:

“Although the scope functions are a way of making the code more concise, avoid overusing them: it can decrease your code readability and lead to errors. Avoid nesting scope functions and be careful when chaining them: it's easy to get confused about the current context object and the value of `this` or `it`.” [Referência](https://kotlinlang.org/docs/reference/scope-functions.html#function-selection)
(<https://kotlinlang.org/docs/reference/scope-functions.html#function-selection>).

A mensagem indica que mesmo tendo a ideia de deixar o código mais conciso e legível, o mal uso das funções de escopo pode diminuir a legibilidade e levar a erros. Portanto, a recomendação é evitar técnicas como funções de escopo aninhadas e tomar cuidado quando usar correntes (chains), pois facilita a confusão do objeto de contexto e o valor do `this` ou `it` .