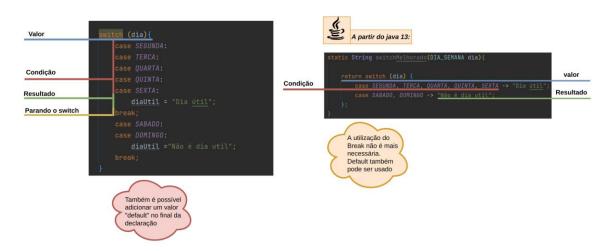


Simplesmente um controle de fluxo que permite a execução de um bloco de comandos se a condição passada for verdadeira. Por exemplo, uma classe de avaliação de meta é instanciada especificamente se a conclusão de uma meta for feita em determinado tempo:





Uma alternativa mais legível em para o "if", se as opções se basearem somente no valor e não em uma expressão. A ideia é decidir que comando vai ser executado dependendo do que for passado como argumento:





For é um loop controlado em uma única expressão. O statement do "for each" é um loop "for" enxuto e inteligente o suficiente para saber o escopo de iteração (a quantidade de iteração é baseada no tamanho da coleção), ele é utilizado em Lists no geral:



O for é utilizado quando você sabe quantas vezes precisa iterar, não à toa o **"for each" funciona baseado no escopo do tamanho de uma lista**, já que é um valor conhecido.

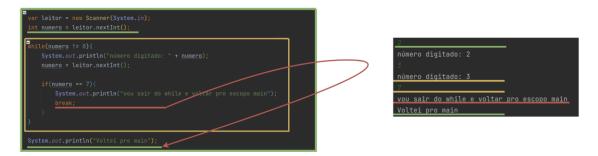


Quando você não tem certeza de quantas vezes precisa iterar, você utiliza o while. Ele é uma estrutura de loop que só tem a condição de parada:





Em casos específicos você **precisa que um fluxo natural seja quebrado**, pra esse comportamento existe a **palavra reservada "break"**, ela para imediatamente qualquer fluxo se a **condição para sua execução for atingida**. Quando ele para o fluxo ele volta para o **escopo maior**:



Essa parte do escopo é bem importante. Eu vejo o *uso dessa palavra reservada como uma falha no design de código*, ela tende a aumentar a complexidade do fluxo de maneira exponencial.

O **continue** faz com que você "pule" uma etapa da iteração, pode ser usado numa **lógica onde você não queira iterar sobre um determinado elemento**:

