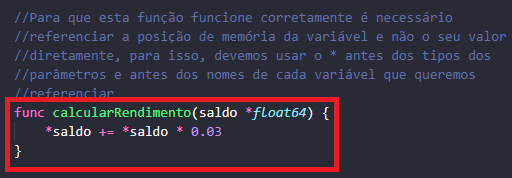
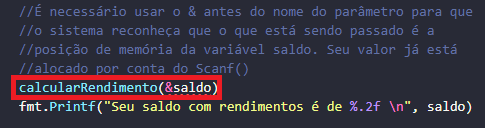
**Ponteiros em Go**

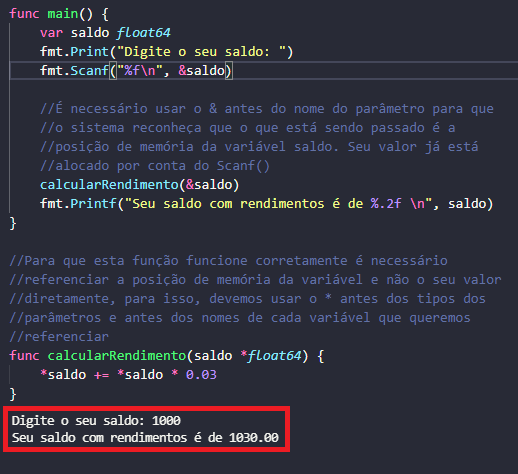
Para que uma função do tipo void que faz operações com seus parâmetros funcione corretamente, é necessário fazer com que seus parâmetros apontem para a posição de memória daquela variável e não necessariamente para os valores das variáveis em si, para isso, utilizamos o \* antes do tipo do parâmetro. Após isso, dentro do bloco de execução da função, vamos colocar o \* antes de cada variável, para que seja retornado seu valor e não o ponteiro.



Após aplicar esta syntax às funções desejadas, devemos usar o & nos parâmetros da chamada da função.

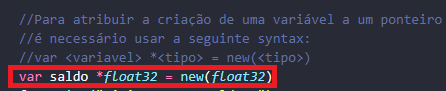


Após estes ajustes na syntax, veremos a função ser executada corretamente:



**Outra forma de referenciar posição de memória**

Podemos também referenciar a posição de memória logo na inicialização da variável, através de uma função de built-in (new()), usando a seguinte syntax:



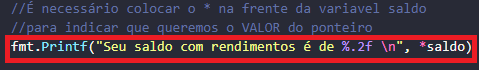
Após essa alteração, não precisaremos mais usar o & no Scanf() do valor da nossa variável, já que a posição de memória (ponteiro) dessa variável já foi referenciada em sua criação:



O mesmo ocorre com os parâmetros da função durante a sua chamada:



Quando vamos printar a variável, preciamos indicar que queremos o valor da variável em si e não o ponteiro.



Após estes ajustes, veremos a função ser executada normalmente:

