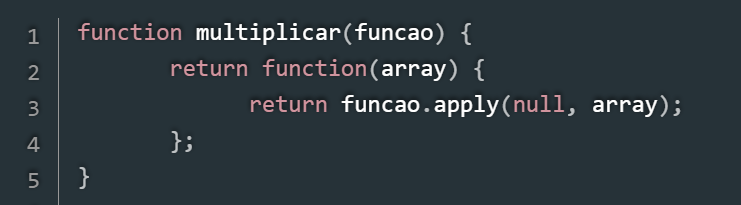
**Estudo de programação funcional em Javascript**

Grupo F - Brenda Pereira, Felipe Freitas e Luiza Heller

19 de outubro de 2023

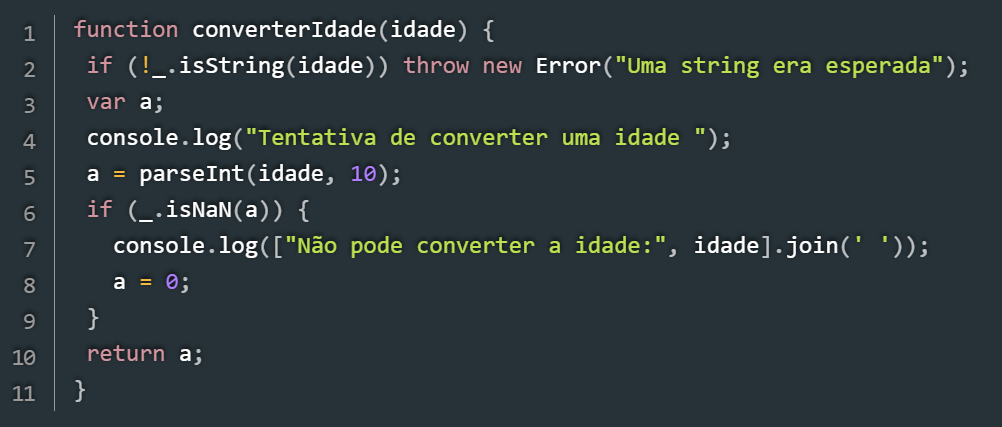
A linguagem Javascript foi criada se baseando inteiramente em funções. Até entidades como classes, objetos etc são implementados através de funções por baixo dos panos. Além de Javascript é uma linguagem de natureza runtime, seus códigos são executados em tempo de execução e não de compilação como outras linguagens como o Java.

Uma das facilidades que a programação funcional nos proporciona com javascript é a possibilidade de chamar uma função que retorna outra função, e isso é uma vantagem pois possibilita o reaproveitamento de código, e um código mais limpo:

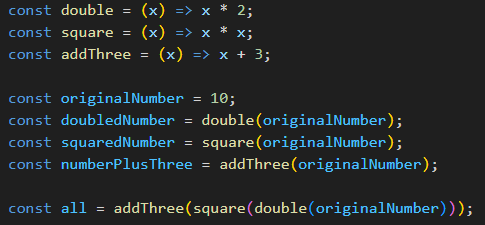


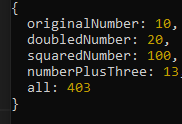
No exemplo acima, a função principal de nome multiplicar(), recebe a função a ser considerada em tempo de execução, e ela é genérica o suficiente para receber qualquer tipo de função que receba um array como parâmetro, e retorna uma outra função que usa o array a ser passado no momento de sua chamada para aplicar a função original apply. O valor null como primeiro argumento dirá ao JavaScript que todo valor nulo deverá ser substituído pelo objeto global (this) da função.

Outra vantagem é poder usar funções como método de abstração, que são funções que ocultam os detalhes de implementação, comumente usadas para tratamento/saída de erro para logs ou debug:

  
A função acima, espera receber uma string e transforma-la em um inteiro, e loga erros caso não consiga.  
*Funções Puras:* Outro aspecto muito importante das linguagens funcionais são as chamadas funções puras. Elas são funções que dependem apenas de seus argumentos para produzir um resultado. Elas não têm efeitos colaterais, como alterar variáveis globais ou parâmetros passados por referência, por exemplo.

Abaixo, segue um código implementado pelo grupo que exemplifica o assunto:



A saída do código: 

Podemos ver que nenhuma variável global (o 10, no caso), foi alterada, mesmo sendo utilizada em diversas funções.

Outro ponto que podemos observar é que não temos uma função que soma 3 ao quadrado do dobro de um número, e sim 3 funções “aninhadas” que, por retornarem cada qual seu resultado, podem se somar para gerar um valor mais útil.

# Referências

https://www.devmedia.com.br/programacao-funcional-com-javascript/34377.