

Funções de alta ordem

Até então, vimos strings, caracteres, inteiros e reais como argumentos de funções, mas e se pudéssemos passar uma função como argumento na chamada de outra? Essas funções, que recebem outras funções como parâmetros, são chamadas de "Funções de alta ordem" ou "Funções de ordem superior".

```
menorQue20 :: Int -> Bool
menorQue20 x = x < 20
```

A função abaixo recebe uma função 'f' como parâmetro, uma lista de inteiros e retorna outra lista de inteiros. Essa função 'f', por sua vez, recebe um inteiro e retorna um booleano.

```
certamenteMenorQue20 :: (Int -> Bool) -> [Int] -> [Int]
certamenteMenorQue20 _ [] = []
certamenteMenorQue20 f (a:as) | f a = (a : (certamenteMenorQue20 f as))
                               | otherwise = (certamenteMenorQue20 f as)
```

```
λ > certamenteMenorQue20 menorQue20 [1..23]
[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19]
```