



## Expressões Condicionais

1. Faça uma função que tenha como saída o nome do mês, de acordo com um número de entrada; ou imprima a mensagem "Erro" se o número não corresponder a nenhum mês. **Utilize if/else.**

Exemplo:

Entrada: 4

Saída: Abril

-- Ex01

-- com if/else

nomeMes :: Int -> String

nomeMes x =

```
    if x == 1 then "Janeiro" else
    if x == 2 then "Fevereiro" else
    if x == 3 then "Marco" else
    if x == 4 then "Abril" else
    if x == 5 then "Maio" else
    if x == 6 then "Junho" else
    if x == 7 then "Julho" else
    if x == 8 then "Agosto" else
    if x == 9 then "Setembro" else
    if x == 10 then "Outubro" else
    if x == 11 then "Novembro" else
    if x == 12 then "Dezembro" else
    "Erro"
```

-- Com guardas

nomeMes' :: Int -> String

nomeMes' x

```
    | x == 1 = "Janeiro"
    | x == 2 = "Fevereiro"
    | x == 3 = "Marco"
    | x == 4 = "Abril"
```

```
-- | ...
    | x == 12 = "Dezembro"
    | otherwise = "Erro"
```

2. Faça uma função que receba dois valores e informe o maior. **Obs:** Utilize guardas.

```
-- Ex02
-- com guardas
maior :: Int-> Int-> Int
maior x y
  | x >= y = x
  | otherwise = y

-- com if/else
maior' :: Int-> Int-> Int
maior' x y = if x >= y then x else y
```

3. Faça uma função que receba três valores e informe o maior. **Obs:** Utilize if/else.

```
-- Ex03
-- com if/else (como pedido no exercicio)
maior3 :: Int -> Int -> Int -> Int
maior3 x y z = if x >= y && x >= z then x else
               if y >= x && y >= z then y else z

-- com guardas
maior3' :: Int -> Int -> Int -> Int
maior3' x y z
  | x >= y && x >= z = x
  | y >= x && y >= z = y
  | otherwise = z
```

4. Faça uma função que recebe um inteiro e retorna verdadeiro se o número for par. **Faça duas implementações, uma com if/else e outra com guardas.**

```
--ifelse
ehPar :: Int -> Bool
ehPar x = if mod x 2 == 0 then True else False
```

```

-- usando guardas
ehPar' :: Int -> Bool
ehPar' x
  | mod x 2 == 0 = True
  | otherwise = False

```

5. Faça uma função que recebe três inteiros x, y e z e imprime “O ultimo numero eh multiplo” caso z seja múltiplo de x ou de y, ou, caso contrário, “O ultimo numero nao eh multiplo de nenhum dos dois”. **Utilize guardas.**

Exemplos:

```

*Main> multiplo 3 5 20
"O ultimo numero eh multiplo"
*Main> multiplo 7 2 21
"O ultimo numero eh multiplo"
*Main> multiplo 10 3 22
"O ultimo numero nao eh multiplo de nenhum dos dois"

```

```

-- if/else
multiplo :: Int -> Int -> Int -> String
multiplo x y z = if mod z x == 0 || mod z y == 0 then "O numero eh
multiplo" else "O numero nao eh multiplo de nenhum dos dois"

```

```

-- guardas
multiplo' :: Int -> Int -> Int -> String
multiplo' x y z
  | mod z x == 0 || mod z y == 0 = "O numero eh multiplo"
  | otherwise = "O numero nao eh multiplo de nenhum dos dois"

```

6. Considere que o preço de uma passagem de avião em um trecho pode variar dependendo da idade do passageiro. Pessoas com 60 anos ou mais pagam apenas 60% do preço total. Crianças até 10 anos pagam 50% e bebês (abaixo de 2 anos) pagam apenas 10%. Faça uma função que tenha como entrada o valor total da passagem e a idade do passageiro e produza o valor a ser pago. **Utilize if/else.**

```

-- if/else
valorPago :: Float -> Int -> Float
valorPago valor idade = if idade >= 60 then 0.6*valor else
  if idade >= 2 && idade <= 10 then 0.5*valor else
  if idade < 2 then 0.1*valor else valor

```

```
-- guardas
valorPago' :: Float -> Int -> Float
valorPago' valor idade
  | idade >= 60 = 0.6*valor
  | idade >= 2 && idade <= 10 = 0.5*valor
  | idade < 2 = 0.1*valor
  | otherwise = valor
```

7. Uma empresa decidiu dar a seus funcionários um abono de salário, baseando-se nos pontos obtidos durante o mês, de acordo com a tabela:

<i>Pontos Obtidos</i>	<i>Prêmio em RS</i>
<i>1 a 10</i>	<i>100,00</i>
<i>11 a 20</i>	<i>200,00</i>
<i>21 a 30</i>	<i>300,00</i>
<i>31 a 40</i>	<i>400,00</i>
<i>A partir de 41</i>	<i>500</i>

Faça uma função que informe o abono de salário de um funcionário.  
**Utilize guardas.**

```
-- if/else
abono :: Int -> Int
abono pontos = if pontos >= 1 && pontos <= 10 then 100 else
  if pontos >= 11 && pontos <= 20 then 200 else
  if pontos >= 21 && pontos <= 30 then 300 else
  if pontos >= 31 && pontos <= 40 then 400 else
  500
```

```
-- guardas
abono' :: Int -> Int
abono' pontos
  | pontos >= 1 && pontos <= 10 = 100
  | pontos >= 11 && pontos <= 20 = 200
  | pontos >= 21 && pontos <= 30 = 300
  | pontos >= 31 && pontos <= 40 = 400
  | otherwise = 500
```