

Exercícios de Swift

- 1) Crie uma variável do tipo `Int` e atribua um valor qualquer a ela, entre -100 e 100.
- 2) Crie uma função que receba um parâmetro do tipo `Int` e retorne um `Optional de String`. Dentro dessa função, faça a seguinte análise:
 - Se o valor for negativo ou zero, retorne `nil`
 - Se o valor for par, retorne `"número par"`
 - Se o valor for ímpar, retorne `"número ímpar"`
- 3) Crie uma variável do tipo `Optional de String` que irá receber o resultado da chamada do método anterior, passando no parâmetro do método a primeira variável criada. Tente desembrulhar esta variável e imprima o seu conteúdo. Caso não seja possível desembrulhar, imprima `"Número negativo ou zero"`
- 4) Crie um dicionário chamado `errorCodes` cuja chave é um `Int` e o valor é uma `String`. Alimente esse dicionário com os seguintes valores:
 - 100: "Continue"
 - 200: "OK"
 - 204: "No Content"
 - 302: "Found"
 - 304: "Not Modified"
 - 404: "Not Found"
 - 405: "Method Not Allowed"
 - 500: "Internal Server Error"
 - 504: "Gateway Timeout"
- 5) Percorra esse dicionário utilizando `for in` e fazendo decomposição de tuplas (chamando as variáveis de `code` e `description`). Imprima cada um dos códigos e valores da seguinte maneira:
`"O código HTTP XXXX significa ZZZZ"`
- 6) Crie um enumerador chamado `Speed`, do tipo `String`, contendo os seguintes casos e seus valores padrões: `slow` (`lento`), `regular` (`normal`), `high` (`alta velocidade`) e `ultra` (`ultra rápido`)
- 7) Crie um método que receba um parâmetro do tipo `Double` que será a velocidade do carro. Este método devolverá um enumerador do tipo `Speed`. Usando `switch`, verifique em que faixa a velocidade se encontra e devolva o enumerador correspondente. A regra é:
 - 0 a 19.99999~ -> `slow`
 - 20 a 49.99999~ -> `normal`
 - 50 a 89.99999~ -> `high`
 - 90 pra frente -> `ultra`

- 8) Crie uma struct chamada **Car** contendo o nome (**name** do tipo **String**), velocidade máxima (**maxSpeed** do tipo **Double**) e ano de fabricação (**year**, do tipo **Int**).
- 9) Crie um método que receba dois parâmetros do tipo **Car** e retorne uma tupla nomeada contendo um **Int** chamado **year** e uma **String** chamada **name**. Crie dois carros e execute o método passando os dois carros. Dentro do método, verifique qual o carro mais antigo e retorne a tupla contendo o nome e o ano de fabricação do carro. Crie uma variável que irá receber o resultado da execução desse método e depois imprima a seguinte mensagem:
"0 carro XXXXX é o mais antigo dos dois. Seu ano de fabricação é YYYY"
- 10) Crie um **Array** de tuplas chamado **guestList**, cujas tuplas são **String** (**name**) e **Int** (**age**). Alimente esse array com 10 tuplas contendo nomes e idades.
- 11) Crie um método que receba 2 parâmetros, um do tipo **Array** de tuplas (**String**, **Int**) e o outro parâmetro do tipo **Int**, que será a posição desejada no array. Este método irá recuperar o convidado que se encontra no array passado, na posição passada e irá imprimir a frase: "0 convidado XXXX tem YYYY anos". Caso o array não contenha o índice passado, imprima a frase "Este índice não é válido para a lista passada". Execute o método passando o array criado no exercício anterior e um índice qualquer.