

## Introdução a Swift (I.b)

1. Definir três métodos que imprimam saudações diferentes em tela e sejam executados da seguinte forma:  
cumprimentarA(pessoa: "João")  
cumprimentar(a: "João")  
cumprimentarA("João")
2. Escrever o método **eMenor(oPrimeiro: Int, oSegundo: Int) -> Bool** deve usar dois números inteiros como parâmetros e retornar **true** se o primeiro número for menor que o segundo número ou **false**, caso contrário.
3. Invocar o método com os números 3 e 5 e imprimir na tela o resultado.
4. Invocar o método com os números 7 e 5 e imprimir na tela o resultado.
5. Invocar o método com os números 10 e 10; e imprimir na tela o resultado.
6. Escrever o método **ePar(umNumero: Int) -> Bool** deve analisar um número inteiro e retornar **true** se o número for par ou **false**, caso contrário.  
*Ajuda: o operador % calcula o resto da divisão entre dois números inteiros. (Ex.: "5 % 2" dá 1 como resultado; enquanto "4 % 2" dá 0 como resultado).*
7. Invocar o método com o número 1 e imprimir na tela o resultado.
8. Invocar o método com o número 2 e imprimir na tela o resultado.
9. Escrever o método **elmparMaiorQueDez(umNumero: Int) -> Bool** que deve analisar um número inteiro como parâmetro e retornar **true** se o número for ímpar e maior do que dez e **false**, caso contrário.
10. Invocar o método com o número 3 e imprimir na tela o resultado.
11. Invocar o método com o número 4 e imprimir na tela o resultado.
12. Invocar o método com o número 13 e imprimir na tela o resultado.
13. Invocar o método com o número 14 e imprimir na tela o resultado.
14. Escrever o método **maximoEntreTresNumeros(umNumeroA: Int, umNumeroB: Int, umNumeroC: Int) -> Int** que deve analisar três números inteiros como parâmetros e retornar o máximo entre os três números.
15. Invocar o método com os números 2, 5 e 11 e imprimir em tela o resultado.

16. Escrever o método ***imprimirImparesPositivos()*** -> ***Void*** que imprima na tela os primeiros 100 números inteiros positivos ímpares.