

# DESAFIO 1 – PALINDROMO

## O que é Palíndromo:

Um palíndromo é uma palavra ou frase que pode-se ler normalmente de sua direção de leitura. O que significa que a palavra continua a mesma. Independente se for lido da esquerda pra direita ou da direita pra esquerda.

## Qual o objetivo do algoritmo:

O Algoritmo tem o objetivo de analisar qual o maior palíndromo encontrado em uma frase ou em uma palavra:

**Exemplo** -> **mariiaovo** -> o maior palíndromo entre essa palavra é “**ovo**” por que é o maior trecho que você pode ler a mesma coisa de trás pra frente

## Abordagem:

- 1 – cria uma função chamada **function\_palindrome**. Ela recebe um texto (ou palavra) que a gente chama de **s**.
- 2 – Logo após eu vou verificar se o texto que foi recebido está vazio, caso ele esteja vazio ele mostra na tela um texto vazio e finaliza o algoritmo, caso não ele vai seguir com o item 3
- 3 – Depois eu vou pegar uma “Caixinha” chamada de **longest** essa “Caixinha” ela vai ser responsável por guardar o maior palíndromo
- 4 – Mediante a isso eu vou passar letra por letra da palavra recebida que a gente chamou de **s**
- 5 – Após olhar letra por letra, vai ser dividido de duas formas diferentes de análises: **uma letra do meio e Duas letras do meio**, assim ele consegue encontrar palíndromos com número ímpar e par de letras.
- 6 – Depois que eu analiso o meio, vou “**esticar**” aos lados pra ver se o que está ao redor também forma um palíndromo.  
  
Se as letras nas pontas ainda forem iguais, ele continua.  
  
Quando elas **deixam de ser iguais**, ele para.
- 7 - Ele guarda o pedacinho em outra caixinha com nome **palindrome**, onde conseguiu encontrar como palíndromo alcançando o limite da palavra.

8 - Vai ser verificado:

***palindrome***: o novo palíndromo que acabou de encontrar

***longest***: o maior palíndromo encontrado até agora

Se o novo for **maior**, ele **substitui** o antigo.

Assim, no final, ***longest*** guarda **o** maior palíndromo de todos.

9 - No final, ele devolve o **maior palíndromo** que encontrou!