



Ciclos e Repetições

Para este exercício usaremos todos os elementos das aulas anteriores e do atual. Os objetivos são:

- Reconheça **padrões**, elementos com características gerais que interagem, ou não, com outros. Ser capaz de criar grupos e relacionamentos.
- Abstraí-lo em conceitos computáveis e objetos os elementos do problema.
- **Decomponha-se** em conceitos e objetos de menor complexidade, dividir o problema em partes menores.
- Modele o problema do slogan baixando para papel ou planilha a representação em números, textos ou gráficos desses elementos.
- Expresse o modelo em formato de código, **algoritmos** da solução.

Soa familiar? Isso porque isso não é nada além de pensamento computacional, algo em que estamos trabalhando desde que começamos (e realmente algo que fazemos na vida toda em situações cotidianas também!!)

Por isso, propomos o seguinte exercício integrativo para poder desenvolver um pouco mais essa prática no nível de programação e código!

começar...

Para essa oportunidade vamos retomar a atividade da aula anterior, você se lembra? Estávamos trabalhando em arrays, acessando-os, modificando-os, adicionando e removendo elementos, e algumas outras coisas. Mas como vimos naquela aula, muitas tarefas foram repetidas mesmo dentro da mesma função, parecia que tínhamos que executar o mesmo passo várias vezes, um número x de vezes. Vamos ver se podemos dar algum dinamismo e eficiência ao nosso código...

1. Começamos a partir de nossa variedade de filmes, que tínhamos o seguinte:

```
let peliculas = ["star wars", "totoro", "rocky", "pulp fiction", "la  
vida es bella"]
```

e nos pediram para passar todos os elementos para letras maiúsculas, o que na época tínhamos feito de forma "manual". Vamos movê-lo para uma maneira mais automática usando loops.

```
function convertirAMayusculas(array){  
  array[0] = array[0].toUpperCase()  
  array[1] = array[1].toUpperCase()  
  array[2] = array[2].toUpperCase()  
  array[3] = array[3].toUpperCase()  
  array[4] = array[4].toUpperCase()  
  return array  
}
```

2. Agora precisamos modificar a função de passagem de elemento() que nos permite adicionar o conteúdo de nossa matriz de filme animado à matriz de filme original.

```
função pasajeDeElementos(array1, array2) {  
  array1.push(array2.pop(). toUpperCase())
```



```
array1.push(array2.pop(). toUpperCase())  
array1.push(array2.pop(). toUpperCase())  
array1.push(array2.pop(). toUpperCase())  
array1.push(array2.pop(). toUpperCase())  
array de retorno1  
}
```

3. Para este ponto se você decidiu trabalhar em um arquivo diferente em lugar para modificar o anterior, lembre-se que tivemos um infiltrado dentro de nossos filmes de animação que tivemos que tirar e salvar em outra variável antes de fazer a passagem de elementos de um array para outro!
4. Finalmente, devemos modificar nossa função comparadora para os filmes como temos feito até agora.

```
const asiaScores = [8, 10, 6, 9, 10, 6, 6, 8, 4];  
const euroScores = [8, 10, 6, 8, 10, 6, 7, 9, 5];  
  
function compararCalificaciones(asia, europa) {  
  let comparacionesAsiaEuropa = []  
  comparacionesAsiaEuropa[0] = asia[0] === europa[0]  
  comparacionesAsiaEuropa[1] = asia[1] === europa[1]  
  comparacionesAsiaEuropa[2] = asia[2] === europa[2]  
  comparacionesAsiaEuropa[3] = asia[3] === europa[3]
```

Bônus extra

Se você chegou aqui você é mais do que bom! Parabéns!

Para que você não fique com o desejo ou que você possa continuar praticando se quiser, propomos esse outro exercício mais, tenha em mente que daqui os exercícios podem subir em dificuldade; como sempre dizemos, paciência, ignore a complexidade e tente resolvê-la com as ferramentas que você tem à sua disposição, você também pode procurar informações extras no google ou documentações que você conhece!!

Concurso.

Um cliente nos pede para fazer um aplicativo que possa determinar os vencedores de um concurso que foi realizado no fim de semana.

Para isso, precisaremos seguir as seguintes instruções e informações para poder desenvolver nosso aplicativo.

Cada participante tem 5 notas, dos quais suas pontuações individuais serão formadas, os participantes com suas respectivas pontuações são:

- Participante A: 5, 8, 4, 9, 5
- Participante B: 8, 7, 8, 6, 8
- Participante C: 7, 5, 10, 8, 3

O concurso consiste em 2 modalidades de seleção para um vencedor:

- Melhor média (a maior pontuação média entre os concorrentes)
- Maior e-tip(a maior pontuação entre as 5 notas de cada participante)

Com essas informações, nosso líder de Tecnologia nos pergunta o seguinte:

1. Determine qual seria a maneira ideal de representar cada participante com suas pontuações.
2. Crie uma função *pontuacaoMedia* à qual receba um participante por parâmetro e deve calcular e **devolver** a pontuação média dele.



3. Crie uma função *pontuacaoMaior* que receba um participante por parâmetro e deve calcular e **devolver** a pontuação mais alta que o participante tem.

4. Logo nosso líder tecnológico nos pede para criar essas duas funções geramos uma nova funcionalidade chamada *competição* que receberá os 3 participantes por parâmetros, e executará as duas funções criadas anteriormente para calcular as médias e pontuações mais altas de cada uma, e deve anunciar (mostrar pelo console) o vencedor de cada modalidade de pontuação.