

Professora Furiosa

Escreva as seguintes funções: 1. `acontece`, que diz se a aula sucederá de acordo com o array dos estudantes que entraram. 2. `aberturas`, que utiliza um array com os arrays dos estudantes que entraram nos outros dias, e a quantidade mínima de estudantes, e diga quais os dias em que as aulas aconteceram e quais não. Por exemplo:

```
aberturas([alunosDaSegunda, alunosDaTerça, alunosDaQuarta], 2)
[true, false, false]
```

```
function acontece(arrayEstudantes, quantMinina){
  var contador = 0;
  for(var i=0; i<arrayEstudantes.length; i++){
    if(arrayEstudantes[i].length <= 0){
      contador++;
    }
  }
  if(contador >= quantMinina){
    return true;
  }
  else {
    return false;
  }
}

function aberturas(arrayEstudantes, quantMinina){
  var dias = [];
  for(var j=0; j<arrayEstudantes.length; j++){
    dias.push(acomece(arrayEstudantes[j], quantMinina));
  }
  return dias;
}
```

Produto

Necessitamos uma função `produto` que receba um array de números e retorne o `produto`: o resultado de multiplicar todos os elementos entre si.

Por exemplo, `produto([1, 4, 7])` deve retornar `28`, que é `1 * 4 * 7`.

Escreva a função `produto`

```
1 function produto(numeros){
2   var res = 1;
3   for(var i=0; i<numeros.length; i++){
4     res = res * numeros[i];
5   }
6   return res;
7 }
```

Fatorial

Desafio - Fatorial

Escreva uma função `fatorial`, que calcule o `fatorial` de um número.

```
function fatorial(numero){
  var res = 1;
  if(numero==0) {
    return 1;
  } else {
    for(var i=1; i<=numero; i++){
      res = res * i;
    }
  }
  return res;
}
```

Mais ou menos

Desafio - Mais Menos

Necessitamos uma função `maisMenos` que receba um array e retorne outro com os seguintes três números:

- na primeira posição, a fração de números que são positivos
- na segunda posição, a fração de números que são zero
- na última posição, a fração de números que são negativos

Por exemplo, `maisMenos([1, 2, 0, -1])` deveria retornar `[0.5, 0.25, 0.25]`, devido a que há 50% de positivos, 25% de zeros, e 25% de negativos.

Escreva a função `maisMenos`.

```
function maisMenos(numeros){
  var res = [];
  var pos = 0;
  var neg = 0;
  var zero = 0;
  for(var i=0; i<numeros.length; i++){
    if(numeros[i] == 0) { zero++; }
    if(numeros[i] > 0) { pos++; }
    if(numeros[i] < 0) { neg++; }
  }
  zero = zero/numeros.length;
  pos = pos/numeros.length;
  neg = neg/numeros.length;
  res.push(zero);
  res.push(pos);
  res.push(neg);
  return res;
}
```

Escada

Desafio - Escada

Queremos representar uma escada com altura variável, utilizando um array de strings.

Por exemplo, uma escada com altura 3, representaremos com o seguinte array:

```
var escada3 = [
  "  ",
  " ##",
  "###"
]
```

E uma escada com altura 5, da seguinte forma:

```
var escada5 = [
  "    ",
  "   ##",
  "  ###",
  " ###",
  "####"
]
```

Escreva uma função `escada` que utilize uma altura (um número) e retorne um array que represente a escada correspondente. Não esqueça que o espaço " " faz parte da resposta :)

```
function escada(numero){
  var res = [];
  for(var i=1; i<=numero; i++){
    res.push(" ".repeat(numero-i) + "#".repeat(i));
  }
  return res;
}
```

Arvore Utópica

Desafio - Árvore utópica

As árvores utópicas crescem de uma forma particular, em dois ciclos:

- cada primavera dobram seu tamanho
- cada verão crescem um metro

Se Laura planta uma árvore utópica com um metro, no final do outono, qual seria sua altura depois de N ciclos?

Alguns exemplos:

- si N = 0, sua altura será 1 metro (não cresceu nada)
- si N = 1, sua altura será de 2 metros (dobrou a altura na primavera)
- si N = 2, sua altura será de 3 metros (cresceu um metro mais no verão)
- si N = 3, sua altura será de 6 metros (dobrou a altura na primavera seguinte)
- E assim ...

Escreva a função `alturaArvoreUtopica`, que utilize uma quantidade de ciclos de crescimento, e retorne a altura resultante da árvore de Laura.

```
function alturaArvoreUtopica(numero){
  var res = 1;
  for(var i=1; i<=numero; i++){
    if(i%2 != 1){
      res++;
    }else{
      res = res*2;
    }
  }
  return res;
}
```