```
/*
  Dica:

  - Alguns exercícios utilizarão métodos, propriedades e funcionalidades vistos
    no começo do curso. Se for necessário, reveja partes das aulas anteriores
    para relembrar.
*/

/*
  01

  - Converta a função abaixo em uma arrow function e utilize-a para exibir um
    valor no console.
*/

/*function convertToString (value) {
  return String(value)
}

console.log(convertToString('Hello World!'))*/

const convertToString = value => String(value)

console.log(typeof convertToString('Hello World!'))

/*
  02

  - Crie uma função que retorne a quantidade de caracteres que uma string
    recebida por parâmetro possui.
*/
const numberStringCharacters = string => string.length

console.log(numberStringCharacters('Hello'))
/*
  03

  - Crie uma função que retorne todos os caracteres de uma string em letras
    minúsculas;
  - Utilize a função para exibir a string abaixo no console.

  "CHOCOTONE E OVO DE PÁSCOA JUNTOS NO MERCADO EM PLENO FEVEREIRO"
*/
const stringToLowerCase = string => string.toLowerCase()


console.log(stringToLowerCase('CHOCOTONE E OVO DE PÁSCOA JUNTOS NO MERCADO EM PLENO
FEVEREIRO'))
/*
  04

  - Crie uma função que recebe 2 parâmetros: um caractere e uma string;
  - Ao ser invocada, a função deve retornar o index do caractere na string.
*/
const characterIndex = (param1, param2) => param2.indexOf(param1)

console.log(characterIndex('d', 'world'))
/*
  05


```

```
  - Crie uma função que, ao ser invocada, retorna um boolean indicando se o item
    passado por argumento existe no array (também passado por argumento).
*/
const itemExistsInArray = arr => arr.includes('idade')

console.log(itemExistsInArray(['name', 'idade', 'altura', 'peso']))
/*
  06

  - Crie uma função que retorna a concatenação de 2 arrays, passados como
    argumentos em sua invocação;
*/
const sumArray = (array1, array2) => {
  let arraySum = array1.concat(array2)

  return arraySum
}

console.log(sumArray([1, 2, 3], [4, 5, 6]))

/*
  07

  - Crie uma função que retorna o array passado como argumento em sua invocação,
    mas com o último item removido.
*/
const lastItemRemoved = arrayModify => {
  let removeItem = arrayModify.pop()  // se inserirmos um console.log na linha abaixo veremos o item removido
  // porem tem que ser nesta linha pq depois do return o programa é  encerrado.
  return arrayModify
}

console.log(lastItemRemoved(['Marcelo', 'Luiz', 'Pereira', 'Santos']))

/*
  08

  - Crie uma função que retorna se o valor passado como argumento em sua
    invocação é null.
*/
const getInfoNull = value => value === null

console.log(getInfoNull('null'))
/*
  09

  - Crie uma função que apenas invoca uma função de callback recebida por
    parâmetro;
  - Crie outra função que apenas exibe seu nome no console;
  - Invoque a função que recebe um callback por parâmetro, passando como
    argumento a função que exibe seu nome no console e veja se o nome realmente
    foi exibido.
*/
const invokeCallback = callback => {
  callback()
}

const myName = () => {
  console.log('Marcelo')
```

```
}

invokeCallback(myName)

/*
  10

  - Crie uma função que invoca uma função de callback recebida por parâmetro.
    A invocação da função recebida por parâmetro deve receber um valor como
    argumento;
  - Crie uma função que retorna o triplo de um número recebido por parâmetro;
  - Faça com que a invocação da função descrita no 1º item deste exercício (10)
    resulte no triplo de 33.
*/
const callCallback = (thirtyThree, callback) => {
  return callback(thirtyThree)
}

const triple = (number) => {
  return number * 3
}

console.log(callCallback(33, triple))
/*
  - Utilizando um forEach, baseado no array "numbers", a cada iteração, exiba a
    mensagem abaixo no console, substituindo os "X" pelas informações corretas;

  "O Xº item do array [X, X, X] é X."
*/

const numbers = [1, 2, 3]

numbers.forEach((number, index, array) => {
  console.log(`O ${index + 1}º item do array [${array}] é ${number}.`)
})

/*
  12

  - Converta o for loop abaixo em um forEach;
  - Após a conversão, verifique se a cópia do array lettersCopy realmente foi
    criada.
*/

const letters = ['v', 'e', 'p']
let lettersCopy = []

for (let i = 0; i < letters.length; i++) {
  lettersCopy.push(letters[i])
}

letters.forEach((array) => {
  lettersCopy.push(array)
})
console.log(lettersCopy)

/*
  13

  - Inclua o markup abaixo em seu index.html;
```

```
  - Gere um template HTML com parágrafos. Cada parágrafo deve conter um item do
    array "review";
  - Ao gerar o template, verifique no browser se os parágrafos foram incluídos
    dentro da section vazia do markup abaixo.

  <article>
    <header>
      <h1>Sobre "Jurassic Park"</h1>
    </header>

    <section data-js="section"></section>
  </article>
*/

const section = document.querySelector('[data-js="section"]')

const reviews = [
  'Eu sempre adorei o filme e quando descobri que tinha o livro também fiquei doido. Demorei um pouco mas acabei comprando e finalmente li \o/.',
  'O primeiro filme foi baseado nesse livro, porém o livro (como sempre) é muito mais completo, com mais personagens, mais acontecimentos e até mesmo mais dinossauros. Na verdade nesse livro tem coisas do segundo e terceiro filme também, eles mudaram bastante nos filmes, acho que pra ficar mais comercial, e se o filme é bom, o livro é 100 vezes melhor.',
  'Michael é um ótimo autor, esse sim pesquisa muito antes de escrever um livro, além da história que já prende sua atenção, ele fala bastante de genética (pra explicar como os dinossauros foram criados) e acaba falando um pouco de programação (informática), por causa dos programas avançados e modernos que o parque tinha. E isso foi uma das coisas que eu achei muito legal, ele explica as coisas com gráficos, tabelas, códigos ... enfim, o cara é foda hahaha.',
  'Recomendo esse livro pra quem curte uma boa história de ficção. Apesar de muita gente pensar que o livro não tem graça, porque o legal mesmo é ver o dinossauro no filme, com todos os efeitos especiais, eu digo pra deixar esse pensamento de lado, pois a história é tão bem contada e os detalhes são tão bem relatados, que você passa a fazer parte da história, e vive todas as emoções hahaha.'
]

let paragraphs = ''

reviews.forEach((review) => {
  paragraphs += `<p>${review}</p>`
})

section.innerHTML = paragraphs

/*
  14

  - Implemente uma função que retorna uma string com a quantidade de pessoas que
    curtiram um post, conforme descrito a seguir;
  - A função deve receber por parâmetro um array com os nomes das pessoas que
    curtiram o post/vídeo/foto;
  - Se o array recebido estiver vazio, a mensagem que a função deve retornar é
    "Ninguém curtiu isso";
  - Se o array conter apenas um nome, como "Rafael", por exemplo, a mensagem
    retornada deve ser "Rafael curtiu isso";
  - Se o array conter 2 nomes, a mensagem retornada deve ser
    "NOME_1 e NOME_2 curtiram isso";
  - Se o array conter 3 nomes, a mensagem retornada deve ser
    "NOME_1, NOME_2 e NOME_3 curtiram isso";
```

```
  - Se o array conter 4 ou mais nomes, a mensagem retornada deve ser
    "NOME_1, NOME_2 e mais X pessoas curtiram isso". O "X" deve ser substituído
    pelo restante da quantidade de pessoas que curtiram o post (além das duas
    pessoas já mencionadas no início da mensagem).
*/

const peopleLiked = (people = []) => {
  if (people.length === 0) {
    console.log('Ninguém curtiu isso')
  } else if (people.length === 1) {
    console.log(`${people[0]} curtiu isso`)
  } else if (people.length === 2) {
    console.log(`${people[0]} e ${people[1]} curtiram isso`)
  } else if (people.length === 3) {
    console.log(`${people[0]}, ${people[1]} e ${people[2]} curtiram isso`)
  } else if (people.length >= 4) {
    console.log(`${people[0]}, ${people[1]} e mais ${people.length - 2} pessoas curtiram isso`)
  }
}

peopleLiked(['Marcelo', 'Cilene', 'Julia', 'Nete'])

const getLikeMessage = (people = []) => {
  const firstName = people[0]
  const secondName = people[1]
  const thirdName = people[2]
  const totalMinusTwoName = people.length - 2

  switch (people.length) {
    case 0:
      return 'Ninguém curtiu isso'
    case 1:
      return `${people} curtiu isso`
    case 2:
      return `${firstName} e ${secondName} curtiram isso`
    case 3:
      return `${firstName}, ${secondName} e ${ thirdName} curtiram isso`
    default:
      return `${firstName}, ${secondName} e mais ${totalMinusTwoName} pessoas curtiram isso`
  }
}

console.log(getLikeMessage(['Jota', 'Nete', 'Jucilene', 'Jair', 'Jucinaldo', 'Cristiane']))


```