

Objeto Arrays

Assim como os demais objetos o objeto array também possui duas maneiras de ser declarada.

Literal:

```
var arr = [1, 2, 3, 4, 5];
```

Formal:

```
var arr = new Array(1, 2, 3, 4, 5);
```

O objeto array também vai ter alguns métodos e propriedades que serão abordados.

Propriedade Length

A propriedade length vai retornar a quantidade de elementos dentro do array.

Método Every

O método every é utilizado para realizar uma verificação em todos os elementos do array. Ela tem como retorno um boolean, caso todos os elementos passem na verificação ela irá retornar true, caso um elemento não passe na verificação ela irá retornar false.

Esse método recebe como argumento uma outra função, dentro dessa função é passado alguns argumentos como elemento, índice e objeto, sendo os dois últimos opcional.

Exemplo da utilização do every:

```
var arr = [1, 2, 3, 4, 5];

arr.every(function(elemento, indice, objeto){
    return typeof elemento === 'number'
})
```

No exemplo, como todos os elementos são do tipo número a função irá retornar true.

Método Some

O método `some` é muito semelhante ao `every`, sua diferença se dá porque nele apenas um elemento precisa passar na verificação para retornar verdadeiro.

Assim como no `every` ele espera receber uma função como argumento e dentro dessa função é passado os argumentos elemento, índice e objeto.

Exemplo de sua utilização:

```
var arr = [1, '2', 3, 'hello', 4];

arr.some(function(elemento, indice, objeto){
  return typeof elemento === 'number';
});
```

Nesse exemplo a função irá retornar `true`, pois teremos ao menos um elemento que irá satisfazer a condição.

Método Filter

O método `filter` funciona de forma semelhante aos métodos anterior, mas aqui ao invés de retornar um boolean, ele irá retornar o próprio elemento.

A sintaxe para sua utilização é a mesma dos anteriores.

Exemplo de sua utilização:

```
var arr = ['a', 1, 2, 3, 'c', 4];

arr.filter(function(elemento, indice, objeto){
  return typeof elemento === 'number';
});
```

Nessa função será retornado um novo array com os elementos que satisfaz a condição.

Método forEach

O método `forEach` é semelhante a um loop `for`. Ele é utilizado quando desejá-se fazer um loop dentro de um array. Sua vantagem é que não precisa ser criada uma variável de incremento.

Exemplo de sua utilização:

```
var arr = [1, 2, 3, 4, 5];

arr.forEach(function(elemento, indice, objeto){
    console.log(elemento);
})
```

Nesse exemplo a cada interação do `forEach` um elemento será impresso no console.

indexOf e lastIndexOf

As funções `indexOf` e `lastIndexOf` serve para varrer p array e retornar o índice do elemento buscado.

No `indexOf` a busca é feita do primeiro elemento ate o último, assim que encontrado a primeira ocorrência do elemento é retornado a sua posição.

Já o `lastIndexOf` possui o mesmo principio, sua diferença se da pelo fato de começar a varrer o elemento de trás para frente.

Em ambas as funções pode-se passar três argumentos, sendo o primeiro o elemento buscado, o segundo a posição inicial de varredura (incluída) e a posição final (excluída).

Em ambos os casos também, caso o elemento não seja encontrado é retornado o valor “-1”.

Map

O método `map` é utilizado para modificar o alterar um elemento a cada interação e retorna a um novo array com todos os elementos modificado.

Assim como as demais ele recebe uma função como argumento e dentro dessa função pode ser passado o elemento, índice e o array.

Exemplo de sua utilização:

```
var numeros = [1, 2, 3, 4, 5]

var novosNumeros = numeros.map(function(elemento, indice, objeto){
    return elemento * indice;
})
```

Método Concat

O método concat serve para juntar dois array, mas não só apenas isso. Ele permite também juntar um array a elemento soltos passados por parâmetro.

Exemplo de sua utilização:

```
var array_01 = [1,2,3,4,5,6];  
var array_02 = ["a", "b", "c", "d"];  
  
var novoArray = array_01.concat(array_02);
```

Método Join

O método join ele converte um array para string, assim como o método toString. Sua diferença se dá pelo fato que no método join é possível passar um elemento que vai ser utilizado para separa um elemento do outro na string retornada.

Exemplo de sua utilização:

```
var arr = [1, 2, 3, 4, 5];  
  
var str = arr.join(' -- ');
```

Método Push

Método push ele serve para adicionar novos elementos ao array. Esse método é destrutivo, significa que o array original será modificado.

Exemplo de sua utilização:

```
var array = [1, 2, 3, 4, 5];  
  
array.push('Hello World');
```

Método Pop

O método pop ele serve para remover o último elemento do array e o retorna. Esse método também é um método destrutivo.

Exemplo de sua utilização:

```
var array = [1, 2, 3, 4, 5];  
  
array.pop();
```

Método shift

O método shift é parecido com o método pop, sua diferença se dá porque neste método o elemento removido será o primeiro.

Exemplo de sua utilização:

```
var array = [1, 2, 3, 4, 5]  
  
array.shift();
```

Método unshift

Este método é muito semelhante ao push, sua diferença é que aqui ao invés de adicionar ao final os elementos passados são adicionados no início.

Exemplo:

```
var array = [1, 2, 3, 4, 5];  
  
array.unshift("hello", "world");
```

Método Reduce e ReduceRight

O método reduce serve para reduzir o array a um único elemento. Ao ser utilizado ele recebe alguns parâmetros, como, uma função que determinará o que será retornado e dentro da função pode ser passados alguns parâmetros, como posição inicial e final. A posição inicial é o primeiro elemento do array, e a posição atual recebe de início o primeiro elemento e a cada iteração vai alterando. Pode-se também passar um valor

após a função para passar um valor inicial, nesse caso o valor atual passa a ser o primeiro elemento do array.

Exemplo de sua utilização:

```
var numeros = [1, 2, 3, 4, 5];

var somaArr = numero.reduce(function(inicial, atual)){
  return inicial + atual
}
```

O método `reduceRight` é bem semelhante, sua diferença é que sua ordem é invertida, o valor inicial passa a ser o último elemento e o atual passa a ser o penúltimo elemento.

Método Reverse

O método `reverse` ele inverte a ordem dos elementos presente no array, no qual os primeiros passam a ser os últimos e os últimos passam a ser os primeiros.

Exemplo de sua utilização:

```
var arr = [1, 2, 3, 4, 5];

arr.reverse
```

Método slice

O método `slice` ele retorna um pedaço do array a partir das posições selecionadas. Ele recebe dois parâmetros, sendo a posição inicial e final. A posição inicial é excluída, e a final é incluída. Diferente dos demais métodos aqui a posição inicial e final não é o índice do array, mas sim a posição.

exemplo de sua utilização:

```
var arr = [1, 2, 3, 4, 5, 6, 7, 8];

arr.slice(3, 6);
```

Método splice

O método splice serve para modificar o array, seja retirando elemento ou incluindo elementos.

Ele pode receber três parâmetros. Sendo o primeiro a posição que vai iniciar a modificação, o segundo a quantidade de itens que será removido e o terceiro os elementos que será adicionado.

Exercício 01

Criar uma função que receba várias notas e calcule a media entre elas. Contudo, se houver pelo menos uma nota zero, reprovar automaticamente.

Resolução:

```
var existeZero = null

calcularMedia(10, 9.5, 4.5, 6);

function calcularMedia(){
  var arrayNotas = converterParaArray(arguments);

  existeZero = arrayNotas.some(function(item){
    return item === 0;
  });

  if(existeZero){
    console.log('Infelizmente você tem uma nota 0. Está reprovado, estude mais!');
    return;
  }

  var somaNotas = arrayNotas.reduce(function(inicial, atual){
    return inicial + atual;
  })

  var media = somaNotas / arguments.length;

  return media;
}

function converterParaArray(elemento){
  var array = []

  for(var i = 0; i < elemento.length; i++){
    array.push(elemento[i]);
  }

  return array;
}
```

Exercício 02

Selecionar todos os parágrafos da página e, se tiver mais de 10 palavras mostrar as primeiras 10 seguido de reticências.

Resolução:

```
var $paragrafos = document.querySelectorAll('p');

percorrerParagrafos($paragrafos);

function percorrerParagrafos(paragrafos){
    for(var i = 0; i < paragrafos.length; i++){
        var paragrafo = paragrafos[i].textContent;
        var arrayPalavrasParagrafos = paragrafo.split(' ');
        var paragrafosReduzidos = arrayPalavrasParagrafos.slice(0, 10)
        paragrafosReduzidos.push('...');
        paragrafos[i].textContent = paragrafosReduzidos.join(' ')

        console.log(paragrafosReduzidos.join(' '));
    }
}
```

Exercício 03

Selecionar todos os parágrafos da página e adicionar uma classe de destaque no parágrafo que contém a palavra javascript.

```
var $paragrafos = document.querySelectorAll('p');

console.log($paragrafos);

for(var i = 0; i < $paragrafos.length; i++){
    var paragrafo = $paragrafos[i].textContent;
    var array = paragrafo.split(' ');
    var existeJs = array.some(elemento => elemento === 'javascript');

    if(existeJs){
        $paragrafos[i].classList.add('js');
    }
}
```