

Programação I

Javascript



JavaScript



Professor
Marcus Aurelius

Arrays

- É um objeto;
- Funciona como arrays de outras linguagens;
- Organiza seus dados através de índices;
- O índice começa a partir do valor zero;
- É uma estrutura heterogênea;
- Recomenda-se trabalhar com valores homogêneos;
- Caso uma posição não tenha valor é retornado undefined;
- Posições intermediárias vazias podem existir no array;

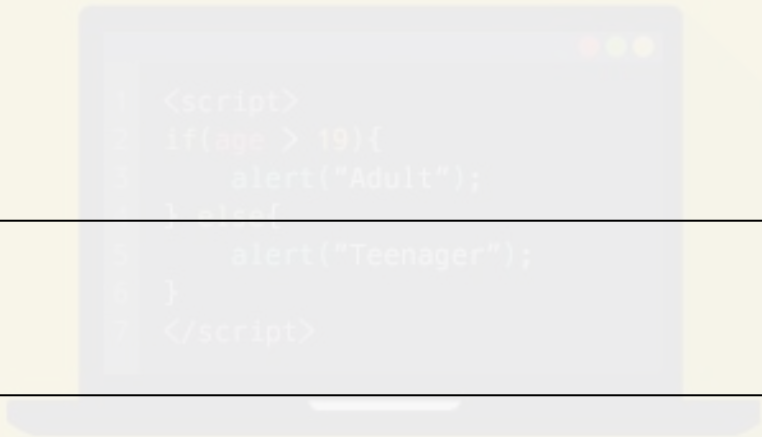


```
<script>
1 if(age > 19){
2     alert("Adult");
3 } else{
4     alert("Teenager");
5 }
6 </script>
```

Arrays

- Quando ordenados as posições vazias são colocados no final;
- Quando um elemento é deletado, o array não é reajustado mas a posição é marcada como undefined;
- Possui tamanho dinâmico;

```
...  
const notas = [8,0.5,9,9.5]  
...
```



```
1 <script>  
2 if(age > 19){  
3     alert("Adult");  
4 } else if{  
5     alert("Teenager");  
6 }  
7 </script>
```

Arrays

Funções/Métodos	Descrição
<code>.pop()</code>	Remove o último elemento do array.
<code>.push(<valor>)</code>	Adiciona um valor na última posição do array.
<code>.shift()</code>	Retira um elemento da primeira posição do array.
<code>.unshift(<valor>)</code>	Adiciona um elemento na primeira posição do array.
<code>.splice(<posição>,<remover>,<elementos>)</code>	Adiciona ou remove elementos do array. Informa a posição, a quantidade a remover, e os elementos a serem inseridos. Para remover informa-se a posição e a quantidade de elementos a remover.

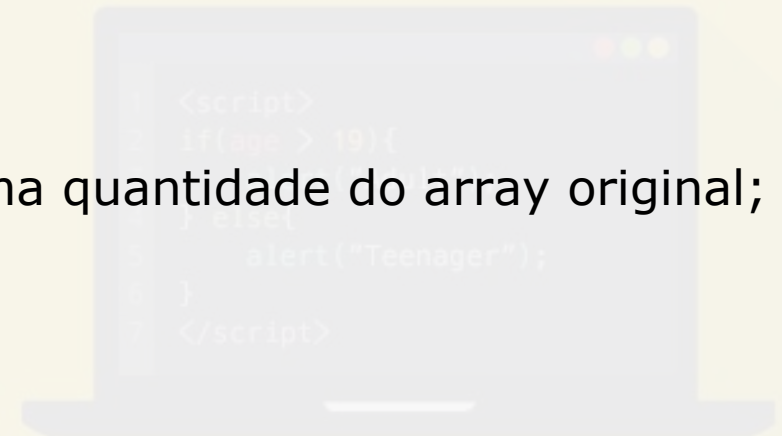
Arrays

Funções/Métodos	Descrição
.length	Retorna a quantidade de elementos em um array.
.toString()	Converte o array em String, separando os elementos por vírgula.
.join(<separador>)	Converte o array em String, passando como parâmetro o separador desejado.
.concat(<array>, ..., <array>)	Cria um novo array concatenando valores de outros arrays. Pode receber também strings como argumentos.

Arrays

Funções/Métodos	Descrição
<code>.slice(<posição>, quantidade)</code>	Gera um novo array com os valores existentes a partir da posição informada, que não entra nesse array.
<code>.slice(<valor a ser retirado>)</code>	Retira elementos de um array, criando um novo array.
<code>.slice(<início>,<fim>)</code>	Retira elementos de um array, criando um novo array, a partir de uma posição de início até a posição anterior à posição final passada como parâmetro. Se a posição final for omitida, todos os demais elementos serão removidos do array.

- Percorre todos os elementos de um array;
- Cria um novo array com os resultados de uma função chamada para cada elemento do array;
- A função não é executada em arrays vazios;
- Não altera o array original;
- Gera um novo array, também com a mesma quantidade do array original;
- Parâmetros:
 - Valor atual(obrigatório);
 - Índice atual (opcional);
 - Array (opcional);



- Sintaxe:

```
<novoArray> = <array>.map(function(valorAtual[, índice[, array]]) {  
    return <cálculos>  
})
```



JavaScript



- Filtra os dados de um array;
- O tamanho do array final sempre é menor, ou igual, ao tamanho do array original;
- O callback vai retornar true ou false, e sendo true o valor será adicionado ao array resultante do filtro;

Sintaxe:

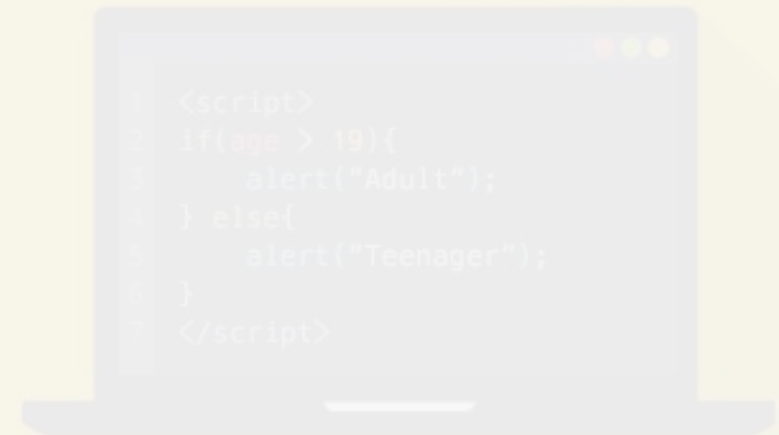


- Sintaxe:

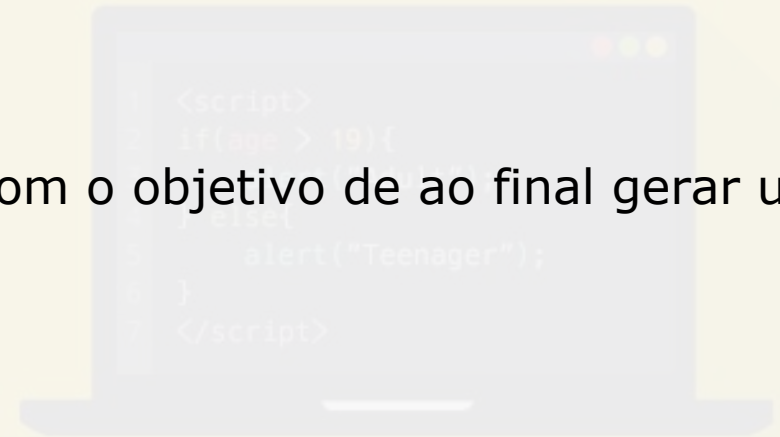
```
<novoArray> = <array>.filter(function(valor) {  
    return <condição de filtragem>  
}))
```



JavaScript



- Busca reduzir um array a um único valor;
- Esse método executa a função passada como parâmetro para cada elemento do array;
- O valor da função é armazenada em uma variável acumuladora;
- A não é executada para arrays vazios;
- Ele iterará por cada elemento dessa lista com o objetivo de ao final gerar um único valor;
- O valor gerado pode ser de qualquer tipo;
- O array original não é modificado;
- Retorna o resultado acumulado na última chamada da função callback;
- Transforma o array em outro elemento;
 - array;
 - número;



- Parâmetros:
 - callback
 - variável acumuladora (na ausência do valor inicial usa-se o primeiro elemento);
 - valor do elemento atual;
 - índice do elemento atual (não obrigatório);
 - arrayoriginal (não obrigatório);
- Reduz e agrega os valores em um só (acumulador);
- Sintaxe:

```
<novoArray> = <array>.reduce(function(valor) {  
    return <somatório dos valores>  
})
```

