

Introdução ao Array em JavaScript

Array ou vetor é um conjunto de variáveis¹ indexadas, permitindo a declaração de uma referência ou variável com a possibilidade de índices para controle autônomo dos conteúdos. Observa-se que o primeiro índice é o zero.

Pode-se declarar um array e inicializá-lo com valores, sendo que os valores deverão estar dentro de colchetes e separados por vírgula, como exemplifica-se:

```
<!DOCTYPE html>
<html>
<body>

  <h2>Array - Vetor</h2>

  <p>
    Todos os itens do array <b id="todos"></b>
  </p>
  <p>
    Item 0 do array <b id="item0"></b>
  </p>
  <p>
    Item 1 do array <b id="item1"></b>
  </p>
  <p>
    Item 2 do array <b id="item2"></b>
  </p>

<script>
  var x = ["HTML", "CSS", "JS"];
  document.getElementById("todos").innerHTML = x;
  document.getElementById("item0").innerHTML = x[0];
  document.getElementById("item1").innerHTML = x[1];
  document.getElementById("item2").innerHTML = x[2];
</script>

</body>
</html>
```

Array - Vetor

Todos os itens do array **HTML, CSS, JS**

Item 0 do array **HTML**

Item 1 do array **CSS**

Item 2 do array **JS**

Observa-se na imagem acima que o array **x** foi declarado e inicializado com três valores que estão armazenados em posições diferentes no array e para acessá-los pode-se optar pelo acesso total, por meio do nome do array **x** ou individualmente, por meio do nome associado ao índice que estará dentro de colchetes ao lado do nome, **x[indice]**, como deve-se imaginar:

x	
Índice	Dado
0	"HTML"
1	"CSS"
2	"JS"

¹ Em programação computacional, variável é a alocação de um tipo de espaço na memória volátil de uma estrutura computacional, i.e., a reserva de um espaço na memória ram.

Há como declarar o array e realizar as inserções de dados após a declaração, conforme exemplifica-se:

```
<!DOCTYPE html>
<html>
<body>

  <h2>Array - Vetor</h2>

  <p>
    Todos os itens do array <b id="todos"></b>
  </p>
  <p>
    Item 0 do array <b id="item0"></b>
  </p>
  <p>
    Item 1 do array <b id="item1"></b>
  </p>
  <p>
    Item 2 do array <b id="item2"></b>
  </p>

<script>
  var x = new Array();
  x[0] = "HTML";
  x[1] = "CSS";
  x[2] = "JS";
  document.getElementById("todos").innerHTML = x;
  document.getElementById("item0").innerHTML = x[0];
  document.getElementById("item1").innerHTML = x[1];
  document.getElementById("item2").innerHTML = x[2];
</script>

</body>
</html>
```

Array - Vetor

Todos os itens do array **HTML, CSS, JS**

Item 0 do array **HTML**

Item 1 do array **CSS**

Item 2 do array **JS**

Uma propriedade muito importante no array é a **length** que determina a quantidade de elementos, pois a propriedade sempre retornará um número inteiro correspondente a quantidade de elementos que detém o array, mesmo que os valores não tenham sido definidos para uma ou mais posições do array, como demonstra-se:

```
<!DOCTYPE html>
<html>
<body>
  <h2>Array - Vetor</h2>
  <p>
    Todos os itens do array <b id="todos"></b>
  </p>
  <p>
    Item 0 do array <b id="item0"></b>
  </p>
  <p>
    Item 1 do array <b id="item1"></b>
  </p>
  <p>
    Item 2 do array <b id="item2"></b>
  </p>
  <p>
    Quantidade de itens do array x: <b id="qtdX"></b>
  </p>
<script>
  var x = new Array(5);
  x[2] = "JS";
  document.getElementById("todos").innerHTML = x;
  document.getElementById("item0").innerHTML = x[0];
  document.getElementById("item1").innerHTML = x[1];
  document.getElementById("item2").innerHTML = x[2];
  document.getElementById("qtdX").innerHTML = x.length;
</script>
</body>
</html>
```

Array - Vetor

Todos os itens do array „JS,,

Item 0 do array **undefined**

Item 1 do array **undefined**

Item 2 do array **JS**

Quantidade de itens do array x: **5**

Outro recurso interessante é o **join** que possibilita a separação dos valores do array no momento da exibição, pois no exemplo abaixo podemos notar que o hífen é utilizado para separar os elementos do array **dsw**:

```
<!DOCTYPE html>
<html>
<body>
  <p id="ver"></p>
  <script>
    const dsw = ["WEB", "Internet", "QI", "TI"];
    document.getElementById("ver").innerHTML = dsw.join(" - ");
  </script>
</body>
</html>
```

WEB - Internet - QI - TI

O recurso **pop** exclui o último elemento do array e o “retorna”, permitindo exibição ou utilização, como podemos observar no exemplo abaixo:

```
<!DOCTYPE html>
<html>
<body>
  <p id="ver1"></p>
  <p id="ver2"></p>
  <p id="ver3"></p>

  <script>
    const dsw = ["WEB", "Internet", "QI", "TI"];
    document.getElementById("ver1").innerHTML = dsw;
    document.getElementById("ver2").innerHTML = dsw.pop();
    document.getElementById("ver3").innerHTML = dsw;
  </script>
</body>
</html>
```

WEB,Internet,QI, TI

TI

WEB,Internet,QI