UNINOVE
Universidade Nove de Julho

Front End
Linguagem JavaScript

...

Objetivo: Entender como o Javascript trabalha com variáveis do tipo array

UNINOVE
Universidade Nove de Julho

Arrays

...

UNINOVE
Universidade Nove de Julho

Arrays

Arrays são estruturas de dados que armazenam uma coleção de elementos de tal forma que cada um deles possa ser identificado por, pelo menos, um índice ou uma chave.

Também são conhecidos como variáveis indexadas, vetores (arrays unidimensionais) ou ainda como matrizes (arrays multidimensionais).

São variáveis que armazenam inúmeros valores na mesma porção de memória.

Comparativamente, uma variável só recebe um único valor, enquanto que os arrays podem receber milhares de valores que serão identificados por chaves.

UNINOVE
Universidade Nove de Julho

Criando arrays em Javascript

Uma das maneiras mais comuns para criar um array em Javascript:

```
const array_name = [item1, item2, ...];
```

Exemplo:

```
const uf = ["SP","RJ","MG","ES"];
```

UNINOVE
Universidade Nove de Julho

Criando arrays em Javascript

Pode ser usado declaração de array com quebra de linhas

Exemplo:

```
const uf = [
    "SP",
    "RJ",
    "MG",
    "ES"
];
```

UNINOVE
Universidade Nove de Julho

Criando arrays em Javascript

Uma vantagem dos arrays em Javascript é que os tipos de dados podem ser alternados,

em um mesmo vetor, ou array, por exemplo.

```
const valores = ["SP",7.5,2021];
```

UNINOVE
Universidade Nove de Julho

Apresentando todos os valores de um Array

O código a seguir irá mostrar todos os valores da variável array:

```
const uf = ["SP","RJ","MG","ES"];
const valores = ["SP",7.5,2021];
const saida=document.querySelector('.saida');

saida.innerHTML=`${uf} <br> ${valores}`;
```

UNINOVE
Universidade Nove de Julho

Acessando elementos de um Array

O primeiro índice de um array é o valor zero, array[0], o demais índices ou posições receberão valores incrementados (somando 1), array[1], array[2] em diante.

UNINOVE
Universidade Nove de Julho

Acessando elementos de um Array

O código a seguir irá mostrar como acessar elementos individuais da variável array:

```
const valores = ["SP",7.5,2021];
const saida=document.querySelector('.saida');

saida.innerHTML=`valores[0] = ${valores[0]}<br>`;
saida.innerHTML+=`valores[1] = ${valores[1]}<br>`;
saida.innerHTML+=`valores[2] = ${valores[2]}<br>`;
```

UNINOVE
Universidade Nove de Julho

Acessando elementos de um Array

Cada posição de uma variável array terá o comportamento similar de uma variável, podendo ser alteradas.

```
const x=[2020,2,"Hoje"];
const saida=document.querySelector('.saida');

saida.innerHTML=`x = ${x}<hr>`;

x[0]++;
x[1]=x[0]%2;
x[2]+=" em dia";

saida.innerHTML+=`x[0] = ${x[0]}<br>`;
saida.innerHTML+=`x[1] = ${x[1]}<br>`;
saida.innerHTML+=`x[2] = ${x[2]}<br>`;
```

UNINOVE
Universidade Nove de Julho

Formas para criar variáveis array

```
const saida=document.querySelector('.saida');
const nota=[10,8.5,7.75,9.5];
const num=[];
num[0]=120;
num[1]=200;
num[2]=2020;
const nome=new Array("Jose","Paula","Carla");

saida.innerHTML=`${nota}<hr>`;
saida.innerHTML+=`${num}<hr>`;
saida.innerHTML+=`${nome}<hr>`;
```

UNINOVE
Universidade Nove de Julho

Formas para criar variáveis array

Cuidado com as formas de criação de array abaixo:

```
const saida=document.querySelector('.saida');
const x=[10];
const y=new Array(10);

saida.innerHTML=`${x}<hr>`;
saida.innerHTML+=`${y}<hr>`;
```

Se puder evite usar new Array!

UNINOVE
Universidade Nove de Julho

Looping com Arrays

...

UNINOVE
Universidade Nove de Julho

Looping com array

Podemos utilizar qualquer das estruturas de looping para trabalhar com array. Abaixo seguem as mais utilizadas:

- for
- for of
- for in

UNINOVE
Universidade Nove de Julho

Looping com array - for

```
const saida=document.querySelector('.saida');
const semana = ["DOMINGO", "SEGUNDA", "TERÇA", "QUARTA", "QUINTA", "SEXTA",
"SÁBADO"];
for (let i = 0; i < 7; i++){
    saida.innerHTML+=`${semana[i]}  é o ${i+1}o dia da semana!<br>`;
}
```

UNINOVE
Universidade Nove de Julho

Looping com array - for of

```
const saida=document.querySelector('.saida');
const estudante=['Astrogildo','Belarmina','Pafuncia'];

for (let n of estudante){
    saida.innerHTML+=`${n}<br>`;
}
```

UNINOVE
Universidade Nove de Julho

Looping com array - for in

```
const saida=document.querySelector('.saida');
const estudante=['Astrogildo','Belarmina','Pafuncia'];

for (let i in estudante){
    saida.innerHTML+=`${i} - ${estudante[i]}<br>`;
}
```

UNINOVE
Universidade Nove de Julho

Métodos e Propriedades - Arrays

...

UNINOVE
Universidade Nove de Julho

Métodos e Propriedades - Array

Para obter informações sobre o array utilizamos diversos métodos e propriedades que estão disponíveis para auxiliar.

UNINOVE
Universidade Nove de Julho

Propriedade length

Informa o tamanho de um array, de acordo com a quantidade de elementos existentes.

```
const saida=document.querySelector('.saida');
const num=[10,20,30,40,50];

saida.innerHTML=`Todos os valores de num: ${num}<hr>`;
saida.innerHTML+=`num possui ${num.length} valores<hr>`;
saida.innerHTML+=`Valor da primeira posição de num: ${num[0]} <hr>`;
saida.innerHTML+=`Valor da última posição de num: ${num[num.length-1]} <hr>`;
```

UNINOVE
Universidade Nove de Julho

Propriedade length

No exemplo a seguir utilizamos uma estrutura de looping para resgatar todos os valores da variável array, utilizando a propriedade length para checar a quantidade de elementos na condição do looping..

```
const saida=document.querySelector('.saida');

const uf=["SP","RJ","ES","MG"];
for (let i=0;i<uf.length;i++){
    saida.innerHTML+=`${i} - ${uf[i]}<br>`;
}
```

UNINOVE
Universidade Nove de Julho

Método toString()

O método toString() converte um array em uma string com valores separados por vírgula.

```
const saida=document.querySelector('.saida');

let frutas = ["Laranja", "Morango", "Melão", "Figo"];
let junto = frutas.toString();
saida.innerHTML=junto;
```

UNINOVE
Universidade Nove de Julho

Método join()

O método Join adiciona uma String entre os elementos de um array, compondo uma string com os elementos do array.

```
const saida=document.querySelector('.saida');

let x = [21,11,2022];
let novaLista = x.join('/');
saida.innerHTML=novaLista;
```

UNINOVE
Universidade Nove de Julho

Método concat( )

O método concat() cria um novo array, concatenando, adicionando uma nova sequência a um array.

```
const saida=document.querySelector('.saida');

let notas=[10,9,8];
let outras_notas=[4,3,2,1];
saida.innerHTML=notas+"<hr>";
let novas_notas=notas.concat(7,6,5);
saida.innerHTML+=novas_notas+"<hr>";
let junta_tudo=novas_notas.concat(outras_notas);
saida.innerHTML+=junta_tudo+"<hr>";
```

UNINOVE
Universidade Nove de Julho

Método push() e unshift()

O método push(valor) adiciona um elemento, ou valor, ao final do array.

O método unshift(valor) adiciona elementos ao início de um array.

```
const saida=document.querySelector('.saida');

const login=["admin","root"];
saida.innerHTML=login+"<hr>";
login.push("astrogildo");
saida.innerHTML+=login+"<hr>";
login.unshift("belinha");
saida.innerHTML+=login+"<hr>";
```

UNINOVE
Universidade Nove de Julho

Métodos pop( ) e shift( )

O método pop() retira o último elemento de um determinado array.

O método shift() retira o primeiro elemento de um array.

```
const saida=document.querySelector('.saida');

const valores=[10,20,30,40,50];
saida.innerHTML=valores+"<hr>";
valores.shift();
saida.innerHTML+=valores+"<hr>";
valores.pop();
saida.innerHTML+=valores;
```

UNINOVE
Universidade Nove de Julho

Métodos sort() e reverse()

O método sort() ordena os valores de um array, com valores strings.

O método reserve() inverte a ordem dos elementos de um array.

```
const saida=document.querySelector('.saida');

var uf=["SP","RJ","AC","TO","MG"];
saida.innerHTML=uf+"<hr>";
uf.sort();
saida.innerHTML+=uf+"<hr>";
uf.reverse();
saida.innerHTML+=uf+"<hr>";
```

https://www.w3schools.com/js/js_array_sort.asp

UNINOVE
Universidade Nove de Julho

Array multidimensional

...

UNINOVE
Universidade Nove de Julho

Array multidimensional

A utilização de array multidimensionais se dá com a criação de um array “dentro” de

outro array. A ideia é a mesma da utilização de uma matriz, onde possuímos linhas e

colunas.

UNINOVE
Universidade Nove de Julho

Array multidimensional

```
const saida=document.querySelector('.saida');

const m = [[1,2,3],[4,5,6],[7,8,9]];

saida.innerHTML+=`${m[0][0]} ${m[0][1]} ${m[0][2]} <br>`;
saida.innerHTML+=`${m[1][0]} ${m[1][1]} ${m[1][2]} <br>`;
saida.innerHTML+=`${m[2][0]} ${m[2][1]} ${m[2][2]} <br>`;
```

UNINOVE
Universidade Nove de Julho

Manipulação de arrays

A seguir seguem links com a relação completa de funções para manipulação de arrays em Javascript.

https://www.w3schools.com/js/js_arrays.asp

https://www.w3schools.com/js/js_array_methods.asp

UNINOVE
Universidade Nove de Julho

Referências

Referências

MORRISON, M. Use a cabeça JavaScript. 5o Ed. Rio de Janeiro: Alta Books, 2012. 606 p.

OLIVIERO. C. A. J. Faça um site JavaScript orientado por projeto. 6o ed. São Paulo: Érica, 2010. 266 p.

ZAKAS, Nicholas C. JavaScript de alto desempenho. 8o Ed. São Paulo: Novatec, 2010. 245 p.