



Arrays II

Nesta aula, vamos ver como podemos desenvolver programas JavaScript que se repetem. Mais do que desenvolver programas sequências e com desvios de decisão de quais blocos de comandos serão executados, para simplificar os programas JavaScript, utilizamos estruturas de repetição.

Em todas as estruturas de repetição, há três informações importantes que precisam ser consideradas: a inicialização de uma variável de controle para garantir que a entrada na estrutura de repetição aconteça adequadamente; a condição para garantir a saída da estrutura de repetição; e a atualização da variável de controle para garantir que a estrutura de repetição não fique no que a gente chama de loop infinito.

Repetição for, while e do/while em JavaScript

Temos a estrutura de repetição do while, que é utilizada quando precisamos verificar se uma condição é verdadeira para entrar e continuar na estrutura de repetição do while, como segue:

```
<inicialização da variável de controle>;
```

```
while (<condição>) {
```

```
    <comandos>;
```

```
    <atualização da variável de controle>;
```



```
}
```



Por exemplo:

```
var x = 0;
```

```
while (x<5) {
```

```
    document.write ("O valor de x é: " + x);
```

```
    x = x + 1;
```

```
}
```

O resultado da execução deste programa JavaScript fica:

Fonte: Autoral

Temos a estrutura de repetição do do/while, que é utilizada quando precisamos executar os comandos da estrutura de repetição, pelo menos uma vez, independente se a condição for verdadeira ou falsa, como segue:

```
<inicialização da variável de controle>;
```

```
do{
```

```
    <comandos>;
```

```
    <atualização da variável de controle>;
```

```
while (<condição> ) ;
```

Por exemplo:



```
var x = 0;  
  
do {  
  
    document.write ("O valor de x é: " + x);  
  
    x = x + 1;  
  
}while (x<5);
```

O resultado da execução deste programa JavaScript, fica:

Fonte: Autoral

No JavaScript, temos a estrutura de repetição do for, que é utilizada quando sabemos a quantidade de repetições que serão executadas, como segue:

```
for (<inicialização>; <condição>; <atualização> )  
  
    <comandos>
```

Por exemplo:

```
var x;  
  
for (x=0 ; x<5 ; x++){  
  
    document.write ("O valor de x é: " + x);  
  
};
```

O resultado da execução do programa JavaScript, fica:



Estrutura de Repetição em JavaScript

Para entender a aplicação das estruturas de repetição, vamos desenvolver um programa em JavaScript que calcula a soma de todos os números inteiros de 11 a 19, usando a estrutura de repetição do while.

Para isso, é importante entender que precisamos de uma variável de controle `i` e uma variável que vai armazenar o acúmulo da soma dos números `s` inicializado com 0, que é o elemento neutro da adição.

Fonte: Autoral

Veja o resultado da execução do programa JavaScript.

Fonte: Autoral

Desenvolva um programa JavaScript que calcula a soma de todos os números compreendidos entre 10 e 20 usando a estrutura de repetição do do/while:

Fonte: Autoral

Fonte: Autoral



Agora, vamos entender a aplicação da estrutura de repetição do `for` desenvolvendo um programa JavaScript que calcula a soma de todos os números inteiros de 11 a 19, usando a estrutura de repetição do `for`.

Para isso, é importante entender que precisamos de uma variável de controle `i` e uma variável que vai armazenar o acúmulo da soma dos números `s` inicializado com 0, que é o elemento neutro da adição. Neste caso, sabemos exatamente que queremos repetir a estrutura para de 11, de um a um até o 19.

Fonte: Autoral

Veja o resultado da execução do programa em JavaScript:

Fonte: Autoral

Vetores e Matrizes em JavaScript

Um vetor é uma condição especial de matriz. Uma matriz é um conjunto de dados dispostos em linhas e colunas. Um vetor é uma matriz que possui uma única linha com várias colunas.

Para declarar e atribuir elementos num vetor, acessar e mostrar estes elementos em JavaScript, segue:

Fonte: Autoral



Para receber dados do usuário, armazenar num vetor e acessar este vetor para apresentar os dados para o usuário, em JavaScript, podemos realizar utilizando uma estrutura de repetição da seguinte forma:

Fonte: Autoral

Vamos entender mais sobre uma matriz que é um conjunto de dados dispostos em linhas e colunas.

Em JavaScript, para declarar, atribuir elementos e mostrar estes dados de uma matriz, podemos realizar da seguinte forma:

Fonte: Autoral

Podemos receber os elementos para armazenar cada posição da matriz do usuário ao invés de alimentar a matriz dentro do programa.

Fonte: Autoral

Observe os resultados da execução deste programa que exemplifica a manipulação dos vetores em JavaScript:

Fonte: Autoral

Estrutura de Vetores e Matrizes em JavaScript



Vamos exemplificar praticando a utilização dos vetores em JavaScript, desenvolvendo um programa que recebe 10 valores inteiros e mostra a soma destes 10 números.

Entendendo a situação-problema, os 10 valores que recebemos do usuário, vamos armazenar num vetor de números inteiros, com 10 posições. O processamento será calcular a soma dos 10 inteiros e a saída de resultados e mostrar o resultado desse cálculo.

Veja que utilizamos a estrutura de repetição do for e os comandos de entrada e saída de dados em JavaScript:


Fonte: Autoral

Observe os resultados da execução deste programa que exemplifica a manipulação dos vetores em JavaScript:

Fonte: Autoral

Fonte: Autoral

Agora, vamos praticar matrizes, desenvolvendo um programa JavaScript que recebe 16 valores numéricos reais numa matriz 4x4 e mostra esses números.

Entendendo a situação-problema, os 16 valores que vamos receber do usuário, armazenaremos numa matriz de 4 linhas e 4 colunas  de números reais, com $4 \times 4 = 16$ posições. O processamento será receber esses números e depois mostrar essas informações.

Fonte: Autoral

Observe os resultados da execução deste programa que exemplifica a manipulação dos vetores em JavaScript.

Fonte: Autoral

Atividade Extra

Texto: JavaScript Array Iteration

Link: https://www.w3schools.com/js/js_array_iteration.asp **(Acesso em 03/08/2022)**

Referência Bibliográfica

Deitel, Paul J. e Deitel, Harvey M. Ajax, Rich. **Internet Applications e Desenvolvimento Web para programadores**. São Paulo. Pearson Prentice Hall. 2008.

Morrison, Michael. **Use a Cabeça! JavaScript**. 1a Edição. Rio de Janeiro. Alta Books. 2009.



Ir para exercício