

Linguagem JavaScript

28/10/14

Prof. Felipe

Vetores

- Vetores permitem armazenar mais de um valor. No javascript são dinâmicos, isto é, para utilizá-los não precisamos especificar seu tamanho. Além disso, ele pode ter seu tamanho modificado durante a execução do script.

```
/* exemplo de declaração de uma variável  
do tipo vetor com tamanho zero */  
var vetor = [];
```

Declaração vetores

- Exemplo de declaração de vetores:

```
/* declaração de um vetor de strings com  
tamanho 4 */  
var conceitos_web = ["A", "B", "A", "C"];  
  
/* acessando a posição de cada elemento: */  
alert(conceitos_web[0]); // A  
alert(conceitos_web[3]); // C
```

Percorrendo vetores

- Para percorrer um vetor, basta usar o atributo `length` e obter o tamanho do mesmo. Usar um `for` para imprimir cada posição:

```
var notas = [6,8,9,2,4,10,7,8,9,4];  
for(i=0;i<notas.length;i++){  
    // apertar F12 para visualizar |  
    console.log(notas[i]);  
}
```

Métodos

- Todo array é na verdade um objeto. Objetos do tipo array possuem métodos para auxiliar na manipulação dos elementos do vetor:

```
var vetor = ["pera", "banana", "uva", "abacate"];  
// adiciona a fruta amora  
vetor.push("amora");  
// remove o último elemento (amora)  
vetor.pop();  
// remove o primeiro elemento (pera)  
vetor.shift();  
// adiciona laranja logo no início do vetor  
vetor.unshift("laranja");  
// ordena o vetor em ordem alfabética  
vetor.sort();
```

Remover um determinado item

- Para remover um elemento em determinada posição do vetor, podemos usar a combinação da função `indexOf` para obter a posição do elemento dentro do vetor e a função `splice` que remove o elemento. O segundo parâmetro da função `splice` refere-se a quantidade de elementos que serão removidos.

```
var alunos = ["julia", "fernando", "diego"];  
// obter a posicao do elemento no vetor:  
var posicao = alunos.indexOf("fernando");  
// usar a função splice para remover o elemento  
alunos.splice(index, 1);
```

Função `setInterval()`

- Função do javascript que permite executar um código ou função repetidamente.
- Vamos criar uma função contadora, que incrementa `i` e atualiza o conteúdo na div `cont`:

```
<script>
var i = 0;
function contador(){
    i++;
    document.getElementById("cont").innerHTML = i;
}
</script>
```

Função `setInterval()`

- Para chamar a função contador repetidamente usamos o `setInterval()` passando como parâmetro o nome da função que será executada e o intervalo de tempo em milissegundos que a função será chamada:

```
<body onLoad="setInterval(contador,1000);" >  
    <div id="cont">0</div>  
</body>
```

Cancelando o setInterval

- Para isso, é necessário criar uma referência para a função setInterval():

```
var interval = setInterval(someFunction, 1000);
```

- Com a variável interval, podemos usar a função clearInterval para cancelar a repetição:

```
clearInterval(interval);
```

Atividades

- 1) Faça um script que lê 4 notas digitadas em 4 inputs e armazena esses valores num vetor. Depois mostre a médias das notas percorrendo o vetor quando o usuário clica num botão exibir média.
- 2) Faça um script que sorteia quem irá fazer as compras do supermercado. Coloque 5 nomes num vetor e sorteie um número de 1 a 5 e mostre quem foi o escolhido.
- 3) Implemente um relógio no javascript que utiliza o setInterval para atualizar as horas, minutos e segundos na tela.
- 4) Faça um script que lê de uma textarea um texto qualquer, e executa uma análise sintática do mesmo, ou seja, deve decompor o texto em frases. Cada frase deve ser exibida em uma linha de forma numerada (use listas). Dica: use a função Split.

Atividades

- 5) Faça uma agenda, que permite adicionar compromisso colocando-os numa lista

- 14h - estudar para a prova
- 18h - fazer trabalho web

- 6) Faça um script que gera 1000 números inteiros aleatoriamente dentro do intervalo de 0 a 100 e salva estes números num vetor. Verifique nesse array quantos números são ímpares e quantos são pares. Mostre o resultado na tela.

Pares: 493

Ímpares: 507