

Arrays - Javascript

Adicionando elementos com push()

O método `push()` adicionará um elemento ao final de um vetor, enquanto sua função dupla, o método `pop()`, removerá um elemento do final da matriz. Para adicionar um elemento ao final de um array usando `push()`, você precisa fazer isso:

```
var list = ["foo", "bar"];

list.push("baz");

["foo", "bar", "baz"] // resultado
```

Você também pode adicionar vários elementos a um array usando `push()`, como mostrado abaixo:

```
var list = ["foo", "bar"];

list.push("baz", "qux", "etcetera");

["foo", "bar", "baz", "qux", "etcetera"] // resultado
```

Se você precisar adicionar um elemento ou vários elementos ao final de uma matriz, o método `push()` será quase sempre sua opção mais simples e rápida.

Adicionado elemento com unshift()

O método `unshift()` adicionará um elemento ao início de um vetor, enquanto sua função dupla, `shift()`, removerá um elemento do início do vetor. Para adicionar um elemento, execute o método da seguinte forma:

```
var list = ["foo", "bar"];

list.unshift("baz");

["baz", "foo", "bar"] // resultado
```

Para adicionar vários elementos ao início de um array usando `unshift()`, tente isto:

```
var list = ["foo", "bar"];

list.unshift("baz", "qux");

["baz", "qux", "foo", "bar"] // resultado
```

Adicionando elemento com splice()

O método splice() modifica o conteúdo de um array removendo elementos existentes e / ou adicionando novos elementos. Abaixo, você encontrará a sintaxe correta da função splice():

```
array.splice(inicio, quantidade [, item1 [, item2 [, ...]]])
```

Se você quiser inserir um elemento (ou elementos) em um determinado ponto em algum lugar dentro do array, além do começo ou fim, então você deve estar usando o método splice(). Para usar splice() seu código deve ficar assim:

```
var list = ["foo", "bar"];

list.splice(1, 0, "baz"); // na posição de índice 1, remova 0 elementos e adicione "baz" a
essa posição

// o elemento "bar" agora será movido automaticamente para a posição de índice 2

["foo", "baz", "bar"] // resultado
```

Para adicionar vários elementos no meio de um array usando splice() tente isto:

```
var list = ["foo", "bar"];

list.splice(1, 0, "baz", "qux");

["foo", "baz", "qux", "bar"] // resultado
```

O comando splice é como o canivete suíço de manipulação do vetor; no entanto, você deve primeiro tentar usar os comandos push ou unshift muito mais simples antes de usar splice() para adicionar a uma matriz.

Adicionando elementos com concat()

O método `concat()` retorna um novo array composto do array original mais a que for passado por parametro no método. Para adicionar alguns elementos a outro array usando `concat()`, faça o seguinte:

```
var list = ["foo", "bar"];

var newlist = list.concat(["baz", "qux"]);

["foo", "bar", "baz", "qux"] // resultado da nova lista
```

Também é possível adicionar diretamente valores que não sejam vetores usando `concat()`:

```
var list = ["foo", "bar"];

var newlist = list.concat("baz", "qux");

["foo", "bar", "baz", "qux"] // resultado da nova lista
```

O método `concat()` é uma solução simples em situações em que você precisa combinar alguns arrays juntos, sem inchar seu código com loops como o `for` ou outros loops iterativos.

Adicionando elementos usando o índice do vetor

Também podemos manipular diretamente o array, sem o uso de qualquer método, referindo-se à posição do índice dentro do array. Aqui nós adicionamos dois novos elementos em posições especificadas:

```
var list = ["foo", "bar"];

list [2] = "baz"; // adiciona o elemento "baz" à posição do índice 2 na matriz

list [3] = "qux";

["foo", "bar", "baz", "qux"] // resultado
```

Aqui nós adicionamos um elemento ao final do array:

```
var list = ["foo", "bar"];

list [list.length] = "baz"; // adiciona o elemento "baz" ao final do array

["foo", "bar", "baz"] // resultado
```

O método de notação de índice geralmente é útil quando você sabe exatamente onde colocar um elemento na matriz, independentemente de onde essa posição possa estar. Essa situação geralmente se aplica a muitos tipos de implementações de algoritmos.