

ECMAScript 6

Rafael Escalfoni

Baseado no site

<https://developer.mozilla.org>

API de Armazenamento Web Storage





Quando armazenar variáveis?

- Em situações em que precisamos guardar informações para evitar retrabalho do usuário
- Exemplo:
 - autenticar toda vez que acessar aplicativo
 - manter os itens de um carrinho de compras
 - manter informações recentes do usuário





API Web Storage

- Fornece acesso ao armazenamento de sessão ou armazenamento local para um domínio específico.
- Armazenamento de sessão:
 - Window.sessionStorage
- Armazenamento local:
 - Window.localStorage
- sessionStorage e localStorage são similares. A única diferença é que localStorage não expira nunca.
- os dados no sessionstorage tem os seus dados limpos ao expirar a sessão da página. A sessão da página dura enquanto o browser está aberto e se mantém no recarregamento da página. **Abrir a página em uma nova aba ou nova janela irá gerar uma nova sessão**, o que diferencia de como os cookies trabalham.





Propriedades

- Storage.length
 - Retorna o número de itens de dados armazenados no objeto Storage





Métodos

- Storage.key()
 - retorna a n-ésima chave no armazenamento
- Storage.getItem()
 - retorna o valor de uma chave passada
- Storage.setItem()
 - Passando uma chave e valor, armazena ou atualiza o valor da chave
- Storage.removeItem()
 - apaga a chave passada do armazenamento
- Storage.clear()
 - esvazia todas as chaves





Storage.key()

```
function forEachKey(callback) {  
  for (var i = 0; i < localStorage.length; i++) {  
    callback(localStorage.key(i));  
  }  
}
```



Storage.getItem()

```
function setStyles() {  
    var currentColor = localStorage.getItem('bgcolor');  
    var currentFont = localStorage.getItem('font');  
    var currentImage = localStorage.getItem('image');  
    document.getElementById('bgcolor').value = currentColor;  
    document.getElementById('font').value = currentFont;  
    document.getElementById('image').value = currentImage;  
    htmlElem.style.backgroundColor = '#' + currentColor;  
    pElem.style.fontFamily = currentFont;  
    imgElem.setAttribute('src', currentImage);  
}
```

Storage.setItem()

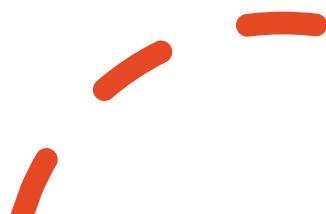
```
function populateStorage() {  
    localStorage.setItem('bgcolor', 'red');  
    localStorage.setItem('font', 'Helvetica');  
    localStorage.setItem('image', 'myCat.png');  
}
```



Storage.removeItem()

```
function populateStorage() {  
    localStorage.removeItem('bgcolor');  
    localStorage.removeItem('font');  
    localStorage.removeItem('image');  
  
    localStorage.setItem('image', 'myCat.png');
```

}





Storage.clear()

```
function populateStorage() {  
    localStorage.setItem('bgcolor', 'red');  
    localStorage.setItem('font', 'Helvetica');  
    localStorage.setItem('image', 'myCat.png');  
  
    localStorage.clear();  
}
```

