
Os Objetos do navegador

Continuando a tratar dos objetos em JavaScript, veremos detalhes dos objetos de `window` (`location` e `history`).

Conheceremos os comandos `with` e `for...in`.



Objetivos da aula:

Nessa aula você aprenderá:

- como é a hierarquia dos objetos em JavaScript;
- os comandos:
 - `with`
 - `for...in`
- as características e métodos dos objetos:
 - `location`
 - `history`

Objetos do navegador ou *browser*

- ❖ Os objetos do *browser* são criados pelo próprio *browser* para definir características do navegador (objeto **navigator**) e da página em exibição (objeto **window**).
- ❖ O objeto **navigator** representa o próprio navegador e através dele é possível controlar o *browser* e obter informações sobre suas características.
- ❖ O objeto **window** representa uma janela do *browser* e contém nele todos os elementos da janela.

O objeto global de JavaScript

- ❖ O objeto global está **no topo** da cadeia de **Hierarquia dos objetos**, isso quer dizer que todas as funções predefinidas e propriedades do ambiente JavaScript também são suas propriedades.
- ❖ Objeto **window** é o **objeto global** para todo o código **contido** na janela do navegador, é o objeto **principal** e os outros objetos do *browser* podem ser **acessados através dele (pelo operador .)**.

Entendendo Hierarquia

```
<HTML> <HEAD> <TITLE>Hierarquia</TITLE> </HEAD> <BODY> <FORM>
Nome : <INPUT TYPE=TEXT VALUE="Fulano"> //elemento 0
E-mail: <INPUT TYPE=TEXT VALUE="fulano@com.br"> //elemento 1
Novo : <INPUT TYPE=TEXT SIZE=30> //elemento 2
<INPUT TYPE=BUTTON Value="Não trocou nome"> //elemento 3
<INPUT TYPE=BUTTON Value="Não trocou email"> //elemento 4
</FORM> <SCRIPT LANGUAGE="javascript">
muda=confirm("Ok=Troca nome e Cancelar=Troca e-mail");
novo=document.forms[0].elements[2].value;
if (muda) { document.forms[0].elements[3].value="Trocou nome";
            document.forms[0].elements[0].value = novo; }
else { document.forms[0].elements[4].value= "Trocou email";
       document.forms[0].elements[1].value = novo; }
</SCRIPT> </BODY> </HTML>
```

Entendendo Hierarquia

```
<HTML> <HEAD> <TITLE>Hierarquia</TITLE> </HEAD> <BODY> <FORM>
```

```
Nome : <INPUT TYPE=TEXT VALUE="Fulano"> //elemento 0
```

```
E-mail: <INPUT TYPE=TEXT VALUE="fulano@com.br"> //elemento 1
```

```
Novo : <INPUT TYPE=TEXT SIZE=30> //elemento 2
```

```
<INPUT TYPE=BUTTON Value="Não trocou nome"> //elemento 3
```

```
<INPUT TYPE=BUTTON Value="Não trocou email"> //elemento 4
```

```
</FORM> <SCRIPT LANGUAGE="javascript">
```

```
muda=confirm("Ok=Troca nome e Cancelar=Troca e-mail");
```

```
novo=document.forms[0].elements[2].value;
```

```
if (muda) { document.forms[0].elements[3].value="Trocou nome";
```

```
    document.forms[0].elements[0].value = novo; }
```

```
else { document.forms[0].elements[4].value= "Trocou email";
```

```
    document.forms[0].elements[1].value = novo; }
```

```
</SCRIPT> </BODY> </HTML>
```

Comando `with`

- **facilita a manipulação dos objetos,**
- **permite uma economia de digitação.**

sintaxe: `with (objeto) comandos`

função: adicionar `objeto` para a frente na cadeia de escopo de `comandos`.

exemplo anterior com with

- ❖ pode-se substituir o texto entre as tags **<SCRIPT....>...</SCRIPT>** por:

```
muda = confirm("Ok=Troca nome e Cancelar=Troca e-mail");
```

```
with ( document . forms [0] ) {
```

```
  novo = elements[2] . value;
```

```
  if (muda) { elements [3] . value = "Trocou nome";
```

```
    elements [0] . value = novo; }
```

```
  else { elements [4] . value = "Trocou email";
```

```
    elements [1] . value = novo; }
```

```
  }
```


Comando for...in

Permite percorrer todas as propriedades de um objeto.

sintaxe: `for (variável in objeto) comandos`

- ❖ `variável` pode ser alguma coisa possível de estar no lado esquerdo de uma expressão de atribuição
- ❖ `objeto` é o nome de um objeto ou uma expressão que pode ser equivalente a um objeto.

```
el0 = document . forms [0] . elements [0];
```

```
for (i in el0 ) document.write(i+"- "+el0[i]+"<br>");
```

Comando for...in

Permite percorrer todas as propriedades de um objeto.

sintaxe: `for (variável in objeto) comandos`

- ❖ `variável` pode ser alguma coisa possível de estar no lado esquerdo de uma expressão de atribuição
- ❖ `objeto` é o nome de um objeto ou uma expressão que pode ser equivalente a um objeto.

```
e10 = document . forms [0] . elements [0];
```

```
for ( i in e10 ) document.write(i+"- "+e10[i]+"<br>");
```



Objeto history

- ❖ O objeto `history` dá acesso à **lista de URLs navegadas**.
- ❖ Este objeto tem **4 propriedades**:
 - `length` Número de páginas armazenadas
 - `current` URL da página atual.
 - `next` URL da página posterior a atual na lista.
 - `previous` URL da página anterior a atual na lista.
- ❖ Este objeto tem **3 métodos**:
 - `go(n)` vai para a *n-ésima* página.
 - `back()` Volta para a página anterior.
 - `forward()` Vai para a página seguinte.

ejemplo de uso de history

```
<HTML><HEAD><TITLE>History</TITLE></HEAD>
<BODY><form>
<input type=button value="Anter." onclick="history.go(-2)">
<input type=button value="Prev." onclick="history.back()">
</form><script language=JavaScript>
document.write(history.length + " páginas visitas <br>" );
prop = window.history;
for (i in prop ) document.write(i+"- "+prop[i]+"<br>");
</script> </BODY> </HTML>
```

Clicar

ejemplo de uso de history

```
<HTML><HEAD><TITLE>History</TITLE></HEAD>
<BODY><form>
<input type=button value="Anter." onclick="history.go(-2)">
<input type=button value="Prev." onclick="history.back()">
</form><script language=JavaScript>
document.write(history.length + " páginas visitas <br>" );
prop = window.history;
for (i in prop ) document.write(i+"- "+prop[i]+"<br>");
</script> </BODY> </HTML>
```

Objeto location

- ❖ Este objeto é utilizado para armazenar informações relacionadas a **URL corrente**.
- ❖ A **propriedade** mais importante deste objeto é a ***string*** que contém o **endereço completo** da URL corrente: a propriedade `href`.
- ❖ Este objeto contém dois métodos.

`reload()` permite **recarregar** a página corrente.

`replace()` **carrega e mostra** uma URL especificada substituindo a página antiga pela nova na lista `history` do *browser*.

finalizando....

Nesta aula você foi apresentado ao conceito de hierarquia dos objetos da linguagem JavaScript. Conheceu os comandos **with** e **for...in** e viu em detalhes as características de duas classes de objetos do navegador : **location** e **history**.

Na próxima iremos ver como funcionam os **objetos do navegador document** e **navigator** !

