

MiniCurso



JavaScript

Thassio Ramon Vieira Lucena

Roteiro

Sintaxe

- Declarações
- Estrutura de dados e Tipos

Tratamento de Eventos

- Eventos Comuns
- Praticando

Funções

- O que são?
- Criando Funções
- Praticando



JavaScript

Sintaxe

Sintaxe básica

- JavaScript pega emprestado a maior parte de sua sintaxe do Java, mas também é influenciado por Awk, Perl e Python;
- JavaScript é **case-sensitive** e usa o conjunto de caracteres **Unicode**;
- Instruções são chamadas de declaração e são separadas por um ponto e vírgula (;)

Comentários

- A sintaxe dos comentários em JavaScript é semelhante como em C++ e em muitas outras linguagens:

```
1 // comentário de uma linha
2
3 /* isto é um comentário longo
4    de múltiplas linhas.
5    */
6
7 /* Você não pode, porém, /* aninhar comentários */ SyntaxError */
```

Declarações

Existem três tipos de declarações em JavaScript.

- **var:** Declara uma variável, opcionalmente, inicializando-a com um valor;
- **let:** Declara uma variável local de escopo do bloco, opcionalmente, inicializando-a com um valor;
- **const:** Declara uma constante apenas de leitura.

Estrutura de dados e Tipos

O mais recente padrão ECMAScript define sete tipos de dados:

- **Boolean:** True ou False;
- **Null:** Uma palavra-chave que indica valor nulo;
- **Undefined:** Uma propriedade superior cujo valor é indefinido;
- **Number:** 13 ou 15.929;
- **String:** “Hello Word”;
- **Symbol:** (novo em ECMAScript 6). Um tipo de dado cuja as instâncias são únicas e imutáveis;
- **Object;**

Estrutura de dados e Tipos

Embora esses tipos de dados sejam uma quantidade relativamente pequena, eles permitem realizar funções úteis em suas aplicações.

Objetos e funções são outros elementos fundamentais na linguagem.





JavaScript

Eventos

Eventos: Introdução

- Eventos são procedimentos executados em consequência a uma ação.

Quando o usuário clica em um botão, é disparado um evento deste elemento chamado “click”



Tratamento de Eventos

- O tratamento de eventos geralmente é feito a partir de funções nas quais se implementa o código que deve ser executado quando o evento ocorrer.
- Essas funções são chamadas de “event handlers” (tratadores de evento).

Tratamento de Eventos

- JS dá suporte a captura de vários eventos do usuário, que vão do arrasto do mouse ao apertar de uma tecla;
- Os eventos geralmente vem inseridos em elementos HTML, e são precedidos pelo prefixo “on”.

```
<FORM>  
<INPUT TYPE="button"  
  onClick="alert('Oh não,  
  você acionou o sistema de  
  autodestruição!')"  
  VALUE="Não aperte este  
  botão">  
</FORM>
```



Principais eventos JS

Atributo	Quando o procedimento é executado	Descritores HTML onde é permitido
onclick	Quando um objeto é clicado pelo mouse	<a>, <input>
onselect	Quando um objeto é selecionado	<input type=“text”>, <textarea>
onchange	Quando uma seleção ou campo de texto tem seu conteúdo modificado	<input type=“text”>, <textarea>, <select>
onfocus	Quando um componente de formulário ou janela se torna ativa	<textarea>, <body>, <form>, <input>, <select>, <option>
onblur	Quando um componente de formulário ou janela se torna inativa	<textarea>, <body>, <form>, <input>, <select>, <option>
onmouseover	Quando o mouse está sobre um link	<a>, <area>
onmouseout	Quando o mouse deixa um link	<a>, <area>
onsubmit	Antes de enviar um formulário	<form>
onreset	Antes de limpar um formulário	<form>
onload	Após carregar uma página, janela ou frame	<body>
onunload	Ao deixar uma página, janela ou frame	<body>
onerror	Quando um erro ocorre durante a carga de uma imagem ou página	, <body>
onabort	Quando a carga de uma imagem é abortada	

Principais Eventos

- onselect

```
<! DOCTYPE html>  
< html >  
< corpo >
```

Selecione um pouco do texto: < input type = "text" value = "Hello world!"
onselect = "myFunction ()" >

```
< script >  
function myFunction () {  
    alerta ( "Você selecionou algum texto!" );  
}  
</ script >  
  
</ body >  
</ html >
```

Principais Eventos

- onchange

Ocorre quando o valor de um elemento foi alterado.

```
<! DOCTYPE html>
< html >
< corpo >

< p > Selecione um carro novo na lista. </ p >

< select id = "mySelect" onchange = "myFunction ()" >
  < valor da opção = "Audi" > Audi
  < valor da opção = "BMW" > BMW
  < valor da opção = "Mercedes" > Mercedes
  < valor da opção = "Volvo" > Volvo
</ select >

< p > Quando você seleciona um carro novo, uma função é acionada, que exibe
o valor do carro selecionado. </ p >

< p id = "demo" > </ p >

< script >
function myFunction () {
  var x = documento . getElementById ( "mySelect" ). valor ;
  documento . getElementById ( "demo" ). innerHTML = "Você selecionou:" +
x ;
}
</ script >
```



JavaScript

Funções

O que são Funções?

Um conjunto de instruções que executa uma tarefa ou calcula um valor.

Para usar uma função, você deve defini-la em algum lugar no escopo do qual você quiser chamá-la.



Por que Funções?



Você pode reutilizar o código:
defina o código uma vez e
use-o muitas vezes.

Criando Funções

- Com o uso da declaração function:
 - `Function minhaFuncao(){
//aqui bloco de codigo
}`
- Com o uso do objeto construtor precedido da palavra-chave *new*:
 - `Var minhaFuncao = new function (/*aqui bloco de codigo*/)`
- Com o uso da sintaxe literal:
 - `Var minhaFuncao = function(){
//aqui bloco de código
}`

Praticando

1 passo: Criar um documento index.html com o seguinte código:

```
<html>
  <head>
    <title>JavaScript</title>
    <meta charset="UTF-8">
    <script type="text/javascript" src="javascript.js"></script>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <center>
      <br>
      <h1 id ="titulo">Bem Vindos ao minicurso de JavaScript</h1>

    </center>
    <button onclick="primeirafuncao()">CLIQUE</button>
    <button onclick="escondeTitulo()">esconder</button>
    <button onclick="apareceTitulo()">aparecer</button>

  </body>
</html>
```

Praticando

2 passo: Criar um documento javascript.js com o seguinte código:

```
- function primeirafuncao() {  
  alert('bem vindos');  
}  
  
- function escondeTitulo() {  
  document.getElementById('titulo').style.display="none";  
}  
  
- function apareceTitulo() {  
  document.getElementById('titulo').style.display="block";  
}
```

Exemplo: Manipulando uma lâmpada



Integrando Eventos e Funções no JavaScript

```
function acendeLampada() {  
    document.getElementById("luz").src="imagens/lampada-acesa.jpg";  
}  
  
function apagaLampada() {  
    document.getElementById("luz").src = "imagens/lampada-apagada.jpg";  
}  
  
function quebraLampada() {  
    document.getElementById("luz").src = "imagens/lampada-quebrada.jpg";  
}  
}
```

```
<body>  
    <h1>Acenda a Lampada</h1>  
      
  
</body>
```

Dúvidas?



Referências

- Disponível em <https://becode.com.br/javascript-para-iniciantes-origens-o-que-e-para-que-serve/> Acesso em outubro de 2017.
- MORRISON, Michael. **Use a Cabeça:JavaScript**. Rio de Janeiro: Alta Books, 2008.
- W3Schools. **JavaScript Tutorial**. Disponível online em [<http://www.w3schools.com/js/default.asp>]