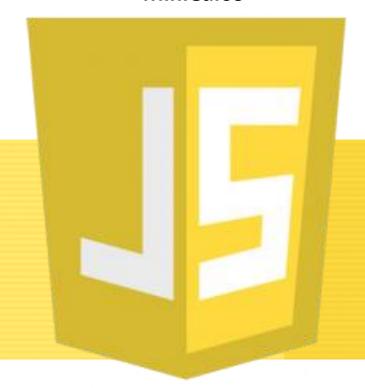
MiniCurso



JavaScript

Thassio Ramon Vieira Lucena



Roteiro

Sintaxe

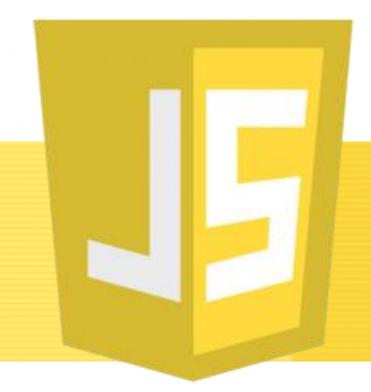
- Declarações
- Estrutura de dados e Tipos

Tratamento de Eventos

- EventosComuns
- 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:

```
// comentário de uma linha
/* isto é um comentário longo
de múltiplas linhas.
/* */
/* 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;

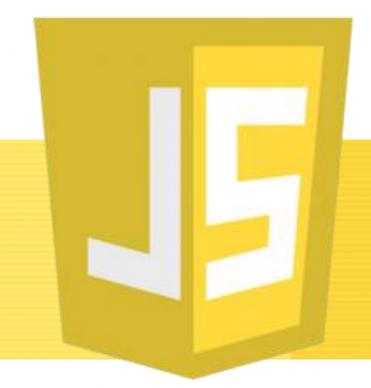


Estrutura de dados e Tipos

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 nāo,
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></td></tr><tr><td>onchange</td><td>Quando uma seleção ou campo de texto tem seu conteúdo modificado</td><td><input type=text>, <textarea>, <select></td></tr><tr><td>onfocus</td><td>Quando um componente de formulário ou janela se
torna ativa</td><td><textarea>, <body>, <form>, <input>, <select>, <option></td></tr><tr><td>onblur</td><td>Quando um componente de formulário ou janela se
torna inativa</td><td><textarea>, <body>, <form>, <input>, <select>, <option></td></tr><tr><td>onmouseover</td><td>Quando o mouse está sobre um link</td><td><a>, <area></td></tr><tr><td>onmouseout</td><td>Quando o mouse deixa um link</td><td><a>, <area></td></tr><tr><td>onsubmit</td><td>Antes de enviar um formulário</td><td><form></td></tr><tr><td>onreset</td><td>Antes de limpar um formulário</td><td><form></td></tr><tr><td>onload</td><td>Após carregar uma página, janela ou frame</td><td><body></td></tr><tr><td>onunload</td><td>Ao deixar uma página, janela ou frame</td><td>
body></td></tr><tr><td>onerror</td><td>Quando um erro ocorre durante a carga de uma
imagem ou página</td><td>, <body></td></tr><tr><td>onabort</td><td>Quando a carga de uma imagem é abortada</td><td></td></tr></tbody></table></textarea>



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!" );
}
</ body >
</ body >
</ html >
```



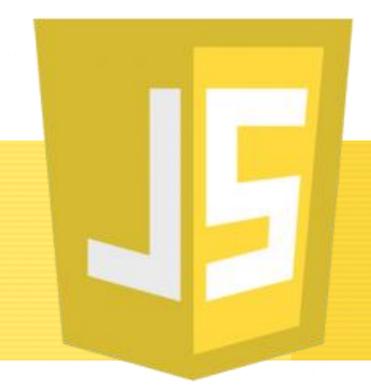
Principais Eventos

onchange

Ocorre quando o valor de um elemento foi alterado.

```
<! DOCTYPE html>
< html >
< corpo >
 Selecione um carro novo na lista. 
< 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 >
 Quando você seleciona um carro novo, uma função é acionada, que exibe
o valor do carro selecionado. 

< script >
function myFunction () {
   var x = documento . getElementById ( "mySelect" ). valor ;
   documento . getElementById ( "demo" ). innerHTML = "Você selecionou:" +
х;
                         JavaScript thassioramon@gmail.com
</ 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:

```
<ht.ml>
    <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>
        <hl id ="titulo">Bem Vindos ao minicurso de JavaScript</hl>
    </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:

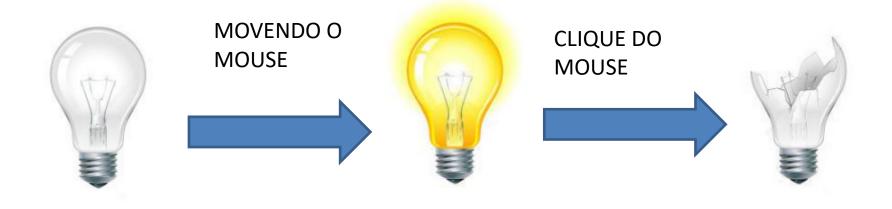
```
function primeirafuncao() {
    alert('bem vindos');
}

function escondeTitulo() {
    document.getElementById('titulo').style.display="none";
}

function apareceTitulo() {
    document.getElementById('titulo').style.display="block";
}
```

15

Exemplo: Manipulando uma lâmpada JavaScript





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";
}
```

Dúvidas?





Referências

- Disponível em https://becode.com.br/javascript-parainiciantes-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]