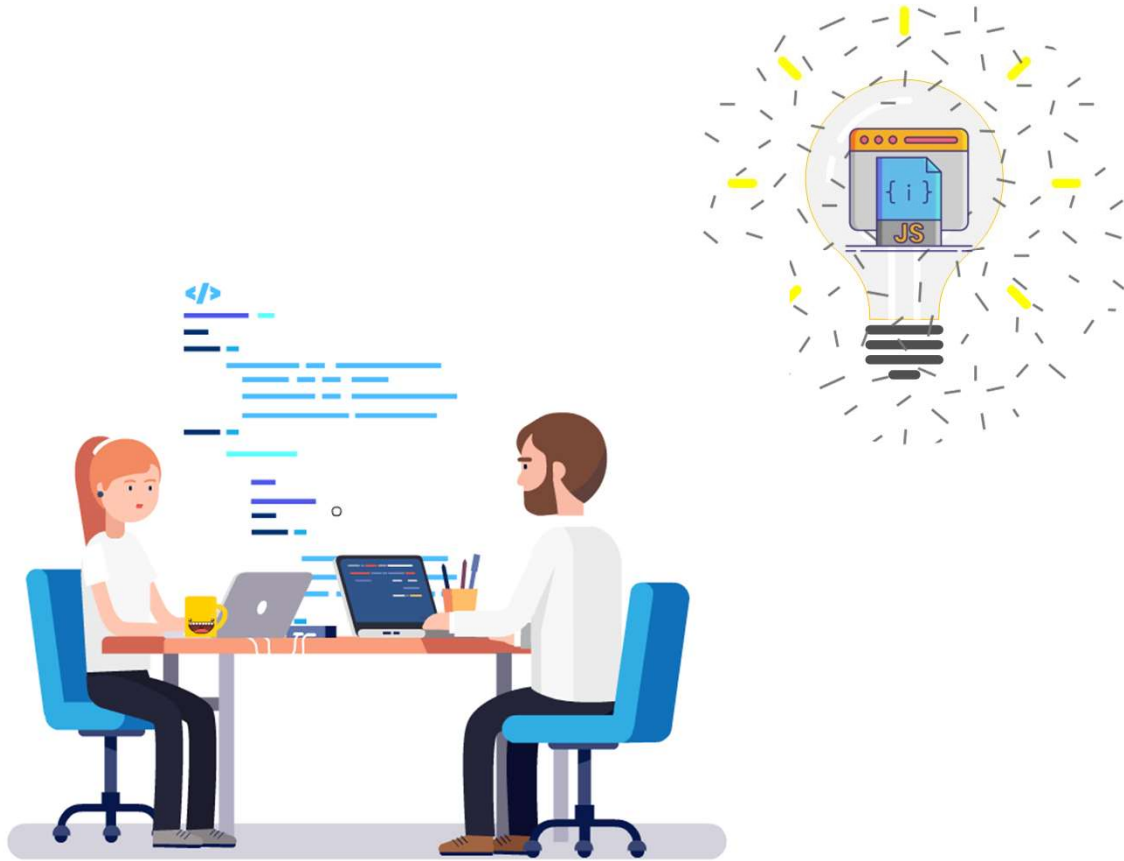


JAVASCRIPT

```
def fib(n):  
    if n == 0:  
        return 0  
    elif n == 1:  
        return 1  
    else:  
        return fib(n-1) + fib(n-2)
```

Diagram illustrating the recursive calculation of the Fibonacci sequence (fib(n)). The diagram shows the function call stack and the return values for different values of n, highlighting the recursive nature of the function.

www.penjee.com



AGENDA

Parte I - JavaScript Básica

Parte II. Estrutura léxica

Parte III. Tipos, valores e variáveis

Parte IV. Expressões e operadores

Parte V. Instruções

Parte VI. Objetos

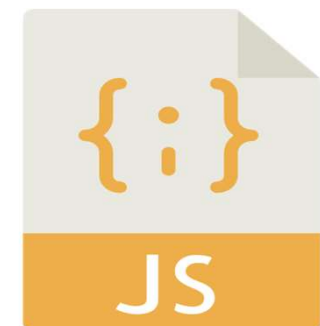
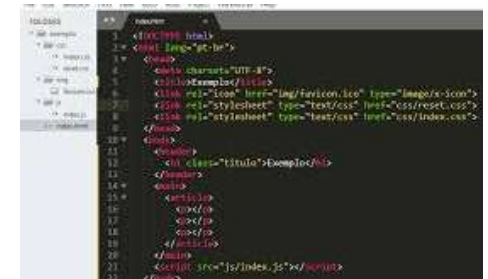
Parte VII. Arrays

Parte VIII. Funções

Parte IX. Classes e módulos

Parte X. JavaScript do lado do servidor

Parte XI. APIs de HTML5



ORIGEM JAVASCRIPT

JavaScript: nomes e versões

JavaScript foi criada na Netscape na fase inicial da Web e, tecnicamente, “JavaScript” é marca registrada, licenciada pela Sun Microsystems (agora Oracle), usada para descrever a implementação da linguagem pelo Netscape (agora Mozilla). A Netscape enviou a linguagem para a ECMA – European Computer Manufacturer’s Association – para padronização e, devido a questões relacionadas à marca registrada, a versão padronizada manteve o nome estranho “ECMAScript”. Pelos mesmos motivos ligados à marca registrada, a versão da Microsoft da linguagem é formalmente conhecida como “JScript”. Na prática, quase todo mundo chama a linguagem de JavaScript. Este livro usa o nome “ECMAScript” apenas para se referir ao padrão da linguagem.

Na última década, todos os navegadores Web implementaram a versão 3 do padrão ECMAScript e não havia necessidade de se pensar em números de versão: o padrão da linguagem era estável e as implementações dos navegadores eram, na maioria, interoperáveis. Recentemente, uma importante nova versão da linguagem foi definida como ECMAScript versão 5 e, quando este livro estava sendo produzido, os navegadores estavam começando a implementá-la. Este livro aborda todos os novos recursos da ECMAScript 5, assim como todos os recursos consagrados da ECMAScript 3. Às vezes, você vai ver essas versões da linguagem abreviadas como ES3 e ES5, assim como às vezes vai ver o nome JavaScript abreviado como JS.

Quando falamos da linguagem em si, os únicos números de versão relevantes são ECMAScript versões 3 ou 5. (A versão 4 da ECMAScript esteve em desenvolvimento por anos, mas se mostrou ambiciosa demais e nunca foi lançada.) Contudo, às vezes você também vai ver um número de versão de JavaScript, como JavaScript 1.5 ou JavaScript 1.8. Esses são números da versão do Mozilla: a versão 1.5 é basicamente a ECMAScript 3 e as versões posteriores incluem extensões não padronizadas da linguagem (consulte o Capítulo 11). Por fim, também existem números de versão vinculados a interpretadores ou “engines” de JavaScript específicos. O Google chama seu interpretador JavaScript de V8, por exemplo, e quando este livro estava sendo produzido a versão corrente do mecanismo V8 era a 3.0.

JAVASCRIPT

É uma linguagem de programação da Web, usada pela maioria dos navegadores modernos, através PC's, consoles de jogos, tablets e smartphones – incluem interpretadores JavaScript, tornando-a a linguagem de programação mais onipresente da história. JavaScript faz parte da tríade de tecnologias que todos os desenvolvedores Web devem conhecer: HTML, para especificar o conteúdo de páginas Web; CSS, para especificar a apresentação dessas páginas; e JavaScript, para especificar o comportamento delas.



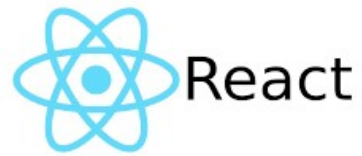
POPULARIDADE (TIOBE)

Mar 2021	Mar 2020	Change	Programming Language	Ratings
1	2	▲	C	15.33%
2	1	▼	Java	10.45%
3	3		Python	10.31%
4	4		C++	6.52%
5	5		C#	4.97%
6	6		Visual Basic	4.85%
7	7		JavaScript	2.11%
8	8		PHP	2.07%
9	12	▲	Assembly language	1.97%
10	9	▼	SQL	1.87%

Language Types				
<div><div> Web</div><div> Mobile</div><div> Enterprise</div><div> Embedded</div></div>				
Rank	Language	Type		Score
1	Python▼		 	100.0
2	Java▼		 	95.3
3	C▼		 	94.6
4	C++▼		 	87.0
5	JavaScript▼			79.5
6	R▼			78.6
7	Arduino▼			73.2
8	Go▼			73.1
9	Swift▼			70.5
10	Matlab▼			68.4
11	Ruby▼			66.8

JAVASCRIPT

- JavaScript é
 - Uma linguagem de script interpretada
 - Orientada a objetos (baseada em protótipos)
 - Dinâmica
 - Fracamente tipada
- Navegadores suportam scripts que rodam código no lado-cliente
- JavaScript é o nome “comum” de versões da linguagem
- O JavaScript tem como objetivo principal ser usado para escrever funções que são embarcadas ou incluídas em páginas HTML e que interagem com o Modelo de Objeto de Documentos (DOM) da página. Alguns exemplos deste uso são:
 - abrir uma nova janela (dialog) com controle programático sobre seu tamanho, posição e atributos;
 - validar valores de um formulário para garantir que são aceitáveis antes de serem enviados ao servidor;
 - mudar imagens à medida que o mouse se movimenta sobre elas.



Vue.js

ONDE PODE SER COLOCADO

- JavaScript no documento:

- Código inline

```
<script type="text-javascript">  
Código  
</script>
```

- Código em arquivo externo

```
<script type="text-javascript" src="arquivo.js">  
</script>
```


ONDE PODE SER COLOCADO

- Elemento SCRIPT:
 - Pode aparecer múltiplas vezes dentro dos elementos HEAD e BODY
 - No HEAD usualmente colocam-se funções
 - No BODY usualmente colocam-se código e chamada a funções que geram conteúdo dinamicamente
 - O script pode ser definido dentro do conteúdo do elemento ou através de referência via atributo *src*
 - A linguagem de script definida via atributo *type*
- Elemento NOSCRIPT:
 - Deve ser avaliado no caso de scripts não suportados ou desabilitado no navegador
 - Conteúdo do elemento é utilizado ao invés do elemento SCRIPT

EXEMPLO

```
<!DOCTYPE html >
<html>
<head>
  <title>Título</title>
</head>
<body>
<script type="text/javascript">
  document.write("<p>Alô Mundo!</p>");
</script>
<noscript>
  <p>Por favor, habilite o JavaScript em seu navegador.</p>
</noscript>
</body>
</html>
```

COMUNICAÇÃO NAVEGADORES

- Para escrever código que se comunica com os elementos dos navegadores, JavaScript faz uso de diversas APIs
 - Algumas padronizadas pelo W3C
 - DOM – Document Object Model
 - Permite manipular elementos, conteúdos e estilos de documentos
 - XMLHttpRequest
 - Permite adicionar conteúdo adicional sem a necessidade de carregar um novo documento
 - Elemento básico para o AJAX

EXEMPLO DE DO <HEAD> E DENTRO DO <BODY>

```
<html>
<head>
  <script language="javascript">

    function Carregar()
    {
      document.getElementById("texto").innerHTML = "Pronto...";
    }

  </script>
</head>
<body>
  <a href="#" onclick="Carregar()">Próxima página</a>
  <div id="texto"></div>
</body>
</html>
```

LINKS ÚTEIS

- Maiores informações sobre JavaScript:
 - <http://www.ecma-international.org/publications/standards/Ecma-262.htm>
 - <http://www.w3schools.com/js/>

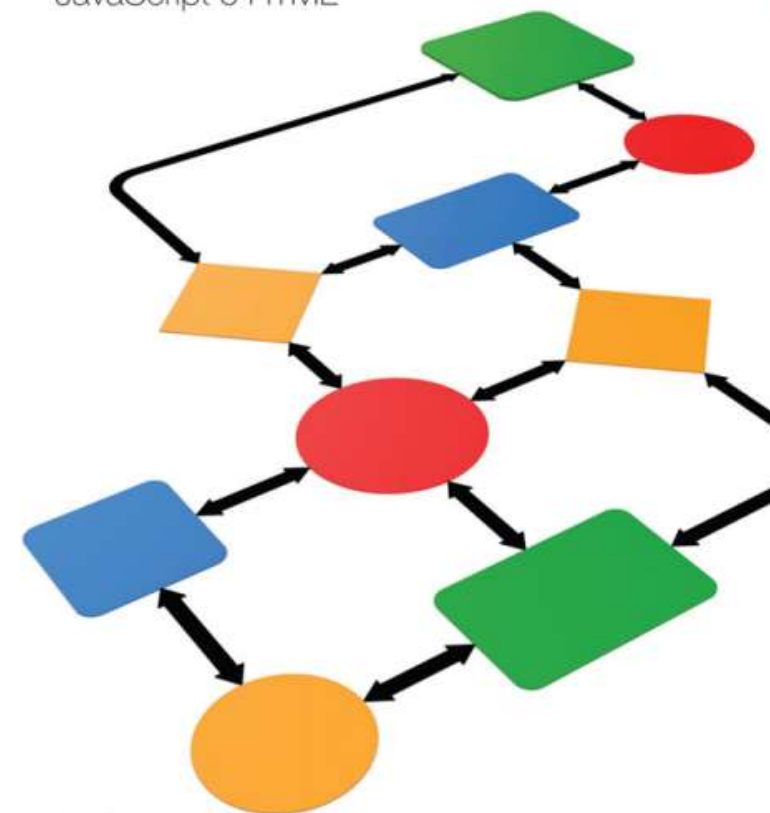


LIVROS RECOMENDADOS



Lógica de Programação

Crie seus primeiros programas usando JavaScript e HTML



Casa do Código

PAULO SILVA
ADRIANO ALMEIDA