

JavaScript

Olá Mundo

HTML



JS
JavaScript

parte 3

aprendendo Javascript

agenda

agora que já conhecemos melhor a estrutura do HTML, vamos ensaiar nossos primeiros códigos em Javascript usando o nosso navegador como saída padrão das nossas mensagens.



quais são as saídas padrões ?

comunicando resultados

um código em Javascript rodando no navegador do usuário muito provavelmente terá duas saídas bem claras para se comunicar com um usuário que esteja acessando a página contendo o código.

Uma **primeira** será interagindo com os elementos do HTML, ou seja, modificando o conteúdo das *tags* para exibir algum resultado, ou até mesmo escrevendo diretamente no body do seu documento criando tags dinamicamente permitindo assim modificar seu conteúdo e apresentar informações.

Uma **segunda** possibilidade seria de interagir através de outros formulários como por exemplo um diálogo que informe uma situação específica e que exija uma interação do usuário. Esta interação poderá ser usada para guiar um novo processamento por exemplo.

Uma **terceira** mas menos usada é saída padrão no console do javascript que existe em memória enquanto se executa seus códigos. Os principais navegadores são capazes de exibí-los com facilidade mas como ficam escondidos nas ferramentas de desenvolvimento, não são indicadas para tais fins, mas apenas para rotina do código onde seus programadores já sabem ou tem ciência de que é por lá que devem buscar por algum dado específico.

manipulando o documento html

acessar um seletor e modificá-lo

ao lado uma estrutura html válida para um arquivo que salvamos como index.html em nosso servidor.

se olharmos com atenção, veremos que há duas tags compondo o conteúdo que está entre o **<body>** e **</body>**, região esta onde fica todo o conteúdo que a página vai exibir. Lembrando que o html é a parte semântica da nossa solução. Nada disto é considerado linguagem de programação pois são por natureza estáticas.

a tag **<h1>** como já sabemos vai imprimir no navegador a mensagem Algo vai aparecer a seguir via JS com estilo de título com maior tamanho.

a tag **<p>** que é uma tag de parágrafo, está contida no documento porém não apresentará nenhuma informação, uma vez que está vazia.

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Aprendendo JS</title>
</head>
<body>
  <h1>Algo vai aparecer a seguir via JS</h1>
  <p></p>
</body>
</html>
```


manipulando o documento html

escrevendo direto no html



```
document.write( 'Conteúdo inserido na página via JS.' );
```

a instrução JavaScript acima vai escrever o conteúdo do parâmetro da função write diretamente no body do documento HTML. em primeira instância isto parece interessante, mas na prática não é tão bacana assim, além de que na primeira execução, ele simplesmente substitui todo o conteúdo original da página lida pelo navegador.

a partir da segunda execução ele vai adicionando ao que já está visível.

manipulando o documento html

acessar um seletor e modificá-lo



```
let paragrafo = document.getElementsByTagName('p')[0];  
paragrafo.innerText = 'Conteúdo inserido via Javascript na página HTML modificando a tag <p>';
```

através do acesso do **DOM**, Document Object Model da página html que escrevemos anteriormente, vamos acessar a tag **<p>** que está sem nenhum conteúdo e portanto sem utilidade alguma. ao acessar o elemento, vamos modificá-lo dinamicamente para conter um conteúdo e prontamente ser exibido no navegador.

acima o código em JavaScript para pegar o elemento através do DOM usando o método **getElementsByTagName** e depois através da propriedade **innerText** informarmos qual é o conteúdo que queremos exibir no navegador.

um detalhe muito importante. o método **getElementsByTagName** sempre irá retornar um **HTMLCollection** portanto vamos ter que dizer qual dos elementos do array nós vamos interagir. ainda não estudamos `arrays` com JS, mas é importante saber que arrays iniciam na posição **0 (zero)**. Mesmo a página só tendo uma tag **<p>**, iremos precisar apontar para a primeira posição como está no código acima.

usando o browser para dar saída

um alerta não muito interessante



```
alert('Olá, eu posso me comunicar por aqui também. Agora clique para fechar');
```

através da instrução acima, escrita em JavaScript, um comando será enviado ao navegador em uso para que seja exibido a mensagem e solicitado que seja clicado um botão para fechar o aviso. normalmente o botão padrão com esta instrução é o **OK** e o formato vai depender de cada navegador.

não é possível customizar esta janela que vai aparecer, não pelo menos até o momento em que escrevo este slide, mas há algumas variações deste mesmo comando que veremos a seguir. *esta função não retorna nenhuma informação.*

usando o browser para dar saída

pode retornar true ou false



```
confirm( 'Você deseja fechar o seu navegador?' );
```

já a instrução acima, escrita em JavaScript, vai mostrar a mensagem no mesmo formato da função `alert`, porém vai apresentar dois botões para que o usuário possa escolher. normalmente um botão é o **OK** e o outro **Cancel**, mas vai depender de cada navegador.

não é possível customizar o nome dos botões, mas é possível recuperar a escolha do usuário para tomar alguma decisão. Se ele clicar no botão **OK** receberemos `true` como retorno, ou `false` para o **Cancel**.

usando o browser para dar saída

coletando um dado



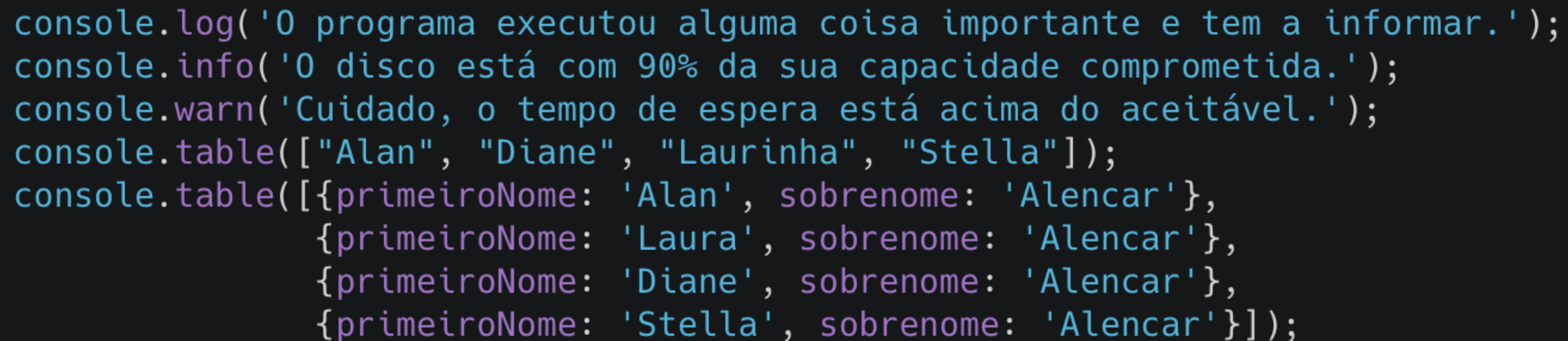
```
prompt( 'Qual o nome da sua cidade natal?', 'Digite aqui' );
```

com a função `prompt`, podemos solicitar uma coleta de dado para o usuário. agora perceba que a função tem dois argumentos. o primeiro é a mensagem ou pergunta e a segunda, opcional é um valor padrão para a resposta ou uma instrução (placeholder). a instrução exibirá ainda dois botões para que o usuário possa escolher. normalmente um botão é o **OK** e o outro **Cancel**, mas vai depender de cada navegador.

se o botão **OK** foi clicado, o valor do preenchimento poderá ser recuperado para tomar alguma decisão. Se ele clicar no botão **Cancel** receberemos `null` como retorno.

usando o browser para dar saída

saída na console do navegador



```
console.log('O programa executou alguma coisa importante e tem a informar.');
```

```
console.info('O disco está com 90% da sua capacidade comprometida.');
```

```
console.warn('Cuidado, o tempo de espera está acima do aceitável.');
```

```
console.table(["Alan", "Diane", "Laurinha", "Stella"]);
```

```
console.table([  
  {primeiroNome: 'Alan', sobrenome: 'Alencar'},  
  {primeiroNome: 'Laura', sobrenome: 'Alencar'},  
  {primeiroNome: 'Diane', sobrenome: 'Alencar'},  
  {primeiroNome: 'Stella', sobrenome: 'Alencar'}]);
```

todas as funções acima vão dar saída de informações na console do navegador, cada qual com sua missão. a primeira é para logs da aplicação, a segunda informações do desenvolvedor para documentar um estado, a terceira uma **warning** e dois formatos para exibir dados estruturados em **tabelas**.

para visualizar, precisa acessar as ferramentas do desenvolvedor, normalmente no atalho **f12**.

como escrever código JS no HTML

inserindo códigos no html



```
<body>

  <h1>Aprendendo JS</h1>

  <p>É possível ter código Javascript combinado com as tags do Html no
    documento que você está usando neste momento. Para que isto seja
    possível o código deve estar entre as tags "script" e "/script"
    como visto a seguir.
  </p>

  <!-- HTML5 não necessita da usar type -->
  <script type="text/javascript">
    console.info('Esta mensagem está sendo exibida na console via JS');
  </script>

</body>
```

é possível deixar nosso código javascript dentro do mesmo arquivo HTML que vai renderizar nossa página e é bem simples.

basta escrever todo o código entre as tags **<script>** e **</script>** como no exemplo ao lado. Uma boa prática é sempre colocar este código no final de toda a informação do documento. Depois falaremos mais sobre isto.

também é possível e altamente recomendado que seu código JS seja escrito em um arquivo separado com a extensão .JS e depois carregado pelo navegador. Veremos mais adiante como fazer isto.



VAMOS PRATICAR

programming

importante recordar

sobre Javascript

1. lembrando que JavaScript é ***Case Sensitive***.
2. as linhas de código podem ou não terminar com ; (ponto e vírgula), mas é recomendado usá-lo.
3. é ***fracamente*** tipado.
4. tem conceito de escopo ***global*** e ***local*** para as suas variáveis.
5. é fácil de depurar ***erros*** usando a console do javascript presente nos principais navegadores.
6. um código JavaScript com erro vai ***afetar por completo*** o comportamento da sua página web.

revendo a prática

princípio básico

1. vimos que um código JavaScript dentro de uma página HTML deve estar entre as tags **<script>** e **</script>** para poder ser interpretado pelo navegador.
2. aprendemos como abrir o **console** do browser.
3. aprendemos como inserir conteúdo na página com **document.write()**.
4. aprendemos a recuperar o conteúdo de uma tag HTML pelo sua Tag Name e alterar seu valor com **innerHTML** e **innerText**.
5. aprendemos a mandar alertas visuais no navegador com o método **alert()**.
6. aprendemos como analisar os nossos **erros** no console do browser.

código

<script>



```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Aprendendo JS</title>
</head>
<body>
  <h1>Vai aparecer algo abaixo</h1>
  <p></p>
  <script>
    let paragrafo = document.getElementsByTagName('p')[0];
    paragrafo.innerText = 'Conteúdo inserido na tag <p> via JavaScript';
  </script>
</body>
</html>
```

código

<script>

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Aprendendo JS</title>
</head>
<body>
  <h1>Vai aparecer algo abaixo</h1>
  <p></p>
  <script>
    document.write('Conteúdo inserido no body via JavaScript');
  </script>
</body>
</html>
```


código

<script>

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Aprendendo JS</title>
</head>
<body>
  <h1>Vai aparecer algo abaixo</h1>
  <p></p>
  <script>
    let resposta = confirm('Você deseja aprender JS:');
    if (resposta) {
      let paragrafo = document.getElementsByTagName('p')[0];
      paragrafo.innerText = '😄 Que ótimo, você optou por aprender JS. Escolha correta.';
    } else {
      let paragrafo = document.getElementsByTagName('p')[0];
      paragrafo.innerText = '😡 não gostei disto.';
    }
  </script>
</body>
</html>
```

código

<script> v2

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Aprendendo JS</title>
</head>
<body>
  <h1>Vai aparecer algo abaixo</h1>
  <p></p>
  <script>
    let resposta = confirm('Você deseja aprender JS:');
    let paragrafo = document.getElementsByTagName('p')[0];
    if (resposta) {
      paragrafo.innerText = '👍 Que ótimo, você optou por aprender JS. Escolha correta.';
    } else {
      paragrafo.innerText = '😡 não gostei disto.';
    }
  </script>
</body>
</html>
```

código

<script>

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Aprendendo JS</title>
</head>
<body>
  <h1>Vai aparecer algo abaixo</h1>
  <p></p>
  <script>
    let nome = prompt('Qual o seu nome?', 'Digite aqui');
    let paragrafo = document.getElementsByTagName('p')[0];
    if (nome != null) {
      paragrafo.innerText = 'Olá ' + nome + ', vamos aprender JS juntos.';
    } else {
      paragrafo.innerText = 'É tão tímido ou antisocial a ponto de não informar o nome?';
    }
  </script>
</body>
</html>
```

código

<script>

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Aprendendo JS</title>
</head>
<body>
  <h1>Vai aparecer algo abaixo</h1>
  <p></p>
  <script>
    console.log('Script JS iniciado com sucesso.');  

    console.info('Se você não estiver acessando a console do desenvolvedor não vai ler isto.');  

    console.warn('Já estamos quase terminando a aula de hoje.');  

    console.table([{ aluno: 'Alan', status: 'Animado'},  

                     { aluno: 'Marcelo', status: 'Gostando muito'},  

                     { aluno: 'Francieli', status: 'Aprendendo'}]);
  </script>
</body>
</html>
```