

[< VOLTAR](#)

Interface com usuário (HTML, CSS e JavaScript)

Capacitar o aluno em relação à tecnologia de interface com o usuário, além de realizar um resgate dos conceitos de páginas bem formatadas com o uso de HTML (HyperText Markup Language), CSS (Cascading Style Sheets) e JavaScript.

NESTE TÓPICO

- › Tecnologia de interface com o usuário
- › Estrutura de documentos HTML (ou XHTML)
- › CSS
- › Exemplo de CSS inline
- › Exemplo de CSS incorporado ou interno



Tecnologia de interface com o usuário

Para que possamos trabalhar com o desenvolvimento de sites dinâmicos, é importante relembrar alguns fundamentos importantes em relação aos sites estáticos, desenvolvidos mais especificamente com HTML, CSS e JavaScript.

Deve-se lembrar que serão utilizados HTML ou XHTML para a criação das páginas de interface homem/máquina, para interação entre os usuários e nossas aplicações desenvolvidas e hospedadas em servidores remotos.

Conforme o W3Schools (2012a), o *XHTML*¹ é um *HTML*² escrito de forma mais rigorosa, que adota os conceitos do XML³, sendo, portanto, uma recomendação do W3C desde janeiro de 2000, com suporte a todos os principais navegadores.

Com o *CSS*⁴ é possível agregar a formatação desejada ao site de forma a garantir uma manutenção rápida e fácil quando criados arquivos HTML com o devido planejamento.

Com o JavaScript, muitos trechos de código podem ser executados no lado cliente para evitar tráfego de informação desnecessária entre cliente e servidor.

Com estas três tecnologias juntas, é possível garantir uma interface amigável, leve e bem estruturada.

Estrutura de documentos HTML (ou XHTML)

Vejamos a seguir a estrutura básica necessária para um documento HTML/XHTML:

```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2.     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3. <html xmlns="http://www.w3.org/1999/xhtml">
4.     <head>
5.         <title>Title of document</title>
6.     </head>
7.     <body>
8.         .....
9.     </body>
10. </html>
```

O exemplo já está mostrando uma linha com !DOCTYPE que informa ao W3C que o documento é um padrão XHTML Transacional.

CSS

Serve para definir como elementos HTML devem ser exibidos. Podemos aplicar a formatação diretamente em uma tag específica, através do CSS inline. As tags dentro de uma página denominamos CSS incorporado ou interno. Ainda pode-se armazenar toda formatação desejada em arquivos CSS, denominando-o de CSS externo.

Segundo o W3Schools (2012b), normalmente os estilos são guardados em arquivos externos. Desta maneira, as folhas de estilo permitem que sejam alteradas, alterando a aparência e o layout de todas as páginas de um site, editando apenas um único arquivo, que é o de formatação CSS vinculado a todas as páginas de um site.



Exemplo de CSS inline

Para utilização de um CSS inline, utiliza-se o **atributo style** na tag desejada. Por exemplo:

```
1. <div style="color:blue; font-size:22px">Desenvolvimento Web</div>
```

O resultado:

Desenvolvimento Web

Exemplo de CSS incorporado ou interno

Para a utilização de um CSS incorporado ou interno, utiliza-se a tag **<style>** no cabeçalho do HTML. Por exemplo:

```
1. <html>
2.   <head>
3.     <title></title>
4.     <style type="text/css">
5.       body {
6.         color: green;
7.         font-size: 22px;
8.       }
9.       span {
10.        font-size:48px;
11.        background-color: red;
12.        font-weight: bold
13.      }
14.    </style>
15.  </head>
16.  <body>
17.    <div style="color:blue; font-size:22px">
18.      <span>D</span>esenvolvimento <span>W</span>eb
19.    </div>
20.    <div>Com... <span>S</span>ervlet e ...</div>
21.    <div><span>J</span>ava<span>S</span>erver <span>P</span>age</div>
22.  </body>
23. </html>
```

O resultado:



Observe que no cabeçalho do HTML (entre <head> e </head>) foi colocada **tag style**, que define que ali será colocado um trecho de formatação CSS, que incidirá em toda página. Este é o CSS interno ou incorporado.

No corpo do HTML (entre <body> e </body>), estão sendo utilizadas tags que são formatadas pelo CSS interno e também com CSS inline através do atributo style.

Exemplo de CSS externo

Neste caso, a única coisa que muda é que o CSS passa a ser criado separadamente do HTML e deve ser vinculado à página através da tag **link**.

Primeiramente, vamos criar o arquivo (**formata.css**) com o seu devido conteúdo:

```
1.  body {
2.      color: green;
3.      font-size: 22px;
4.  }
5.  span {
6.      font-size:48px;
7.      background-color: red;
8.      font-weight: bold;
9.      text-decoration: underline;
10. }
```

A seguir, o arquivo HTML que vincula o CSS anterior, através da **tag link**. Observe o atributo href.: neste caso, ele aponta para um arquivo CSS de nome **"formata.css"**, que está dentro do diretório "Formatação".

```
1.  <html>
2.      <head>
3.          <title></title>
4.          <link rel="stylesheet" type="text/css" href="Formatacao/formata.css" />
5.      </head>
6.      <body>
7.          <div style="color:blue; font-size:22px">
8.              <span>D</span>esenvolvimento <span>W</span>eb
9.          </div>
10.         <div>Com... <span>S</span>ervlet e ...</div>
11.         <div><span>J</span>ava<span>S</span>erver <span>P</span>age</div>
12.     </body>
13. </html>
```

Ao exibir o HTML no navegador, o resultado deve ser o mesmo do CSS interno exemplificado, apenas com uma alteração: os textos exibidos com **estão com atributo de formatação sublinhado (*underline*)**.



JavaScript

Conforme o W3Schools (2012c), JavaScript é uma linguagem padrão utilizada em páginas web, considerada a mais popular do mundo. Além disso, é utilizada em aplicações desktop e em aplicações desenvolvidas para telefonia móvel e servidores de internet.

O JavaScript é de grande valia no desenvolvimento de interfaces na web. Algumas validações em um site, por exemplo, podem ser feitas através de pequenos códigos em JavaScript. Eles podem ser colocados no corpo do HTML sempre que se desejar e serão executados no carregamento da página. Podem ser colocados em funções no cabeçalho ou no corpo do HTML e chamados através de algum evento de tags do HTML ou, ainda, podem ser colocados em arquivos externos criados especificamente para isso, com a extensão **js**.

Exemplo de JavaScript

Trata-se de um exemplo didático, obviamente. O objetivo é mostrar a interatividade entre HTML, CSS e JavaScript. Para isso, foi construído um HTML que aponta para um JavaScript externo. Vejamos primeiramente o código JavaScript (arquivo **programa.js** na pasta **js**):

```

1. function exibeData() {
2.     document.getElementById("demo").innerHTML = Date();
3. }
4.
5. function exhibir(par1)
6. {
7.     if (par1 == false) {
8.         document.getElementById("minhaDiv").style.display = "none";
9.         document.getElementById("btnEsconde").style.display = "none";
10.        document.getElementById("btnMostra").style.display = "inline";
11.    }
12.    else {
13.        document.getElementById("minhaDiv").style.display = "inline";
14.        document.getElementById("btnEsconde").style.display = "inline";
15.        document.getElementById("btnMostra").style.display = "none";
16.    }
17. }

```

Agora vejamos o arquivo HTML que utilizará as duas funções colocadas no arquivo (**programa.js**):

```

1. <!DOCTYPE html>
2. <html>
3.     <head>
4.         <title>Teste de JavaScript (EXTERNO)</title>
5.         <style type="text/css">
6.             #minhaDiv { position:absolute; top:0px; left:0px;
7.                         width:200px; background-color:Red; text-align:center;
8.                     }
9.             #demo { background-color: azure; font-size: 22px; }
10.        </style>
11.        <script type="text/javascript" src="js/programa.js"></script>
12.    </head>
13.    <body>
14.        <br />
15.        <p>O programa JavaScript esta em um arquivo externo, chamado
16.        de: "programa.js" na pasta js.</p>
17.        <p>Um exemplo de interatividade entre HTML, CSS e JavaScript.</p>
18.        <p>Os botoes Mostra e Esconde sao exibidos alternadamente.</p>
19.        <p>A data e hora atualiza-se com JavaScript.</p>
20.        <p id="demo"></p>
21.        <button type="button"
22.            onclick="exibeData()">Exibir Data/Hora (js externo)</button>
23.        <div id="minhaDiv" >AAS - Desenvolvimento Web</div>
24.        <button type="button" id="btnEsconde"
25.            onclick="exibir(false)">Esconde</button>
26.        <button type="button" id="btnMostra"
27.            onclick="exibir(true)" style="display:none">Mostra</button>
28.    </body>
29. </html>

```



O resultado:

AAS - Desenvolvimento Web

O programa JavaScript esta em um arquivo externo, chamado de: "programa.js" na pasta js.

Um exemplo de interatividade entre HTML, CSS e JavaScript.

Os botoes Mostra e Esconde sao exibidos alternadamente.

A data e hora atualiza-se com JavaScript.

Sun Sep 09 2012 20:31:25 GMT-0300 (Hora oficial do Brasil)

Exibir Data/Hora (js externo)

Esconde

No exemplo, exibe-se uma mensagem com fundo vermelho no topo da tela com uma `<div>`, alguns parágrafos de texto com `<p>` e dois botões com `<input>`. Embora tenha três botões, um deles inicia com atributo `display:none`, o que deixa a tag invisível.

A linha com fundo azul só aparece depois que clicado no botão para exibir data e hora. O botão "Esconde" tem a função de esconder o trecho com fundo vermelho e exibir o botão que faz o contrário.

Este pequeno trecho de código precisa ser atentamente analisado e entendido porque nele temos diversos conceitos.

VALE A PENA CONFERIR

1. Veja todos os tipos de !DOCTYPE. Disponível em: [www.w3schools.com](http://www.w3schools.com/tags/tag_doctype.asp) (http://www.w3schools.com/tags/tag_doctype.asp). Acesso em: 12 mar. 2013.

2. Outro site muito interessante é do Site do Maujor, de Mauricio Samy Silva e que discute com propriedade o assunto em questão nesta aula. Silva (2012) é autor de vários livros sobre as tecnologias HTML, CSS e JavaScript (entre outras). Disponível em: www.maujor.com (<http://www.maujor.com/index.php>). Acesso em: 12 mar. 2013.



SAIBA MAIS

XHTML¹: Extensible hypertext markup language.

HTML²: Hypertext markup language.

XML³: Xtensible markup language. É projetada para armazenar e transportar dados.

CSS⁴: Cascading style sheets.

Referências

SILVA, Mauricio Samy. *Site do Maujor*: Site sobre CSS e padrões web. Disponível em: [www.maujor.com](http://www.maujor.com/index.php) (<http://www.maujor.com/index.php>). Acesso em 10 set. 2012.

W3SCHOOLS. *XHTML Introduction*. Disponível em: [www.w3schools.com](http://www.w3schools.com/html/html_xhtml.asp) (http://www.w3schools.com/html/html_xhtml.asp). Acesso em 09 set. 2012.

_____. *S Introduction*. Disponível em: [www.w3schools.com](http://www.w3schools.com/css/css_intro.asp) (http://www.w3schools.com/css/css_intro.asp). Acesso em 09 set. 2012.

_____. *JavaScript Introduction*. Disponível em: [www.w3schools.com](http://www.w3schools.com/js/js_intro.asp) (http://www.w3schools.com/js/js_intro.asp). Acesso em: 09 set. 2012.



Avalie este tópico



ANTERIOR

Servlets, JSP e Servidores de conteúdo

Biblioteca

(<https://www.uninove.br/conhec-a->

a-

uninove/biblioteca/sobre-

a-

biblioteca/apresentacao/)

Portal Uninove

(<http://www.uninove.br>)

Mapa do Site



Índice

Ambiente de desenvolvimento

® Todos os direitos reservados

Ajuda?
PRÓXIMO
(<https://www.uninove.br/cursos/>)
idCurso=)

