

Disciplina: Programação Cliente-servidor

Aula 1: HTTP e HTML 5

Apresentação

É impossível imaginar o mundo atual sem a existência da Internet. Redes sociais, aplicativos bancários, sites informativos, e os mais diversos serviços, são disponibilizados via protocolo HTTP para todo o mundo, derrubando fronteiras e promovendo um acesso muito democrático à informação.

Nós, como desenvolvedores, devemos conhecer este protocolo e as tecnologias para criação de páginas, como HTML e CSS, que durante muito tempo apresentaram diversas inconsistências entre navegadores distintos, mas que, atualmente, sob a normatização da W3C e uso do DOM, converge para um ambiente com baixíssimo índice de incompatibilidade.

Objetivos

- Identificar as características gerais do HTTP;
- Explicar a sintaxe básica do HTML 5 e CSS;
- Descrever as características do Document Object Model (DOM).



Protocolos

Em uma rede de computadores, podemos observar diferentes tipos de informação transitando a todo o tempo, como arquivos, vídeos, páginas e e-mails. Para que estas transferências sejam possíveis, é necessário uniformizar a comunicação entre as partes envolvidas. É neste ponto que aparecem os protocolos.

Um protocolo é, basicamente, um conjunto de regras para a forma de como se dará a comunicação entre dois dispositivos.

Vale lembrar que o contexto atual envolve não só computadores, mas também celulares, componentes IoT, Smart TVs, entre diversos outros.

TCP/IP



Os primeiros protocolos que devemos ter em mente são o **TCP (Transmission Control Protocol)** e o **IP (Internet Protocol)**, pois toda a grande rede da Internet se baseia na combinação destes dois, ou seja, no **TCP/IP, responsáveis pela emissão e recepção de dados entre máquinas da rede**.

Logo acima do TCP/IP, outros protocolos serão necessários para especificar serviços, determinando o formato dos dados a serem transmitidos.

Exemplo

Entre os diversos protocolos existentes, podemos destacar alguns:

- FTP – transmissão de arquivos;
- SMTP e POP – envio e recepção de e-mails;
- SSL – comunicação criptografada;
- HTTP – protocolo de hipertexto, ou hipermídia.

Protocolo HTTP



Em nossos estudos, o protocolo HTTP é de particular interesse, pois será utilizado como base para a implementação de um sistema para a World Wide Web, também denominada simplesmente Web ou WWW, que trata de todo o grande conjunto de documentos de hipermídia disponíveis e interligados existentes na Internet.

DNS



As máquinas da Internet são identificadas por conjuntos de números que vão de 0 a 255, sendo 4 números no caso do IPV4, e seria muito difícil para o ser humano gravar endereços como 127.54.12.208 e 234.57.125.201.

Logo, foi criado um sistema de identificação baseado em nomes, com servidores específicos para a tradução destes nomes para endereços numéricos, os quais são chamados de **servidores DNS (Domain Name System)**.

Com isso, quando digitamos um endereço, como www.estacio.br, este nome é traduzido para o IP da máquina, permitindo encontrar o site.

Devemos considerar, portanto, um domínio como a relação entre o nome utilizado para navegação e o IP da máquina servidora que detém os recursos desejados, tais como páginas e arquivos. Todo site está hospedado em algum domínio, de forma a viabilizar seu acesso através da Internet.

Atenção! Aqui existe uma videoaula, acesso pelo conteúdo online

Servidores HTTP

Para que determinado protocolo de serviço seja suportado, é necessário que exista um programa capaz de “escutar” a rede e prover este serviço.

Estes programas executam em diferentes servidores da rede para que os usuários possam acessar seus serviços através de programas denominados “clientes”, o que permite definir a comunicação como cliente-servidor.

Como poderíamos identificar de forma imediata o tipo de informação solicitada?

A resposta é simples: através do uso de portas diferentes.

Um programa servidor deve escutar uma ou mais portas de comunicação, que definirão o protocolo que será utilizado e, conseqüentemente, o tipo de dado que deverá ser transmitido, como as páginas HTML através do HTTP (porta 80), ou o fornecimento de arquivos via FTP (porta 21).

Existem vários programas servidores (ou simplesmente servidores) que fornecem suporte ao HTTP, como Apache e IIS.

Contudo, para o **ambiente de programação Java, é utilizado o Tomcat como Web Server**, podendo ser executado de forma direta, ou incorporado a outros servidores, como GlassFish, WebSphere, BEA Weblogic e JBoss, estes últimos denominados **Application Servers**.

Web Server

primeiro fornece nativamente apenas o suporte a tecnologias Web, como HTML, JSP e Servlet



Application Server

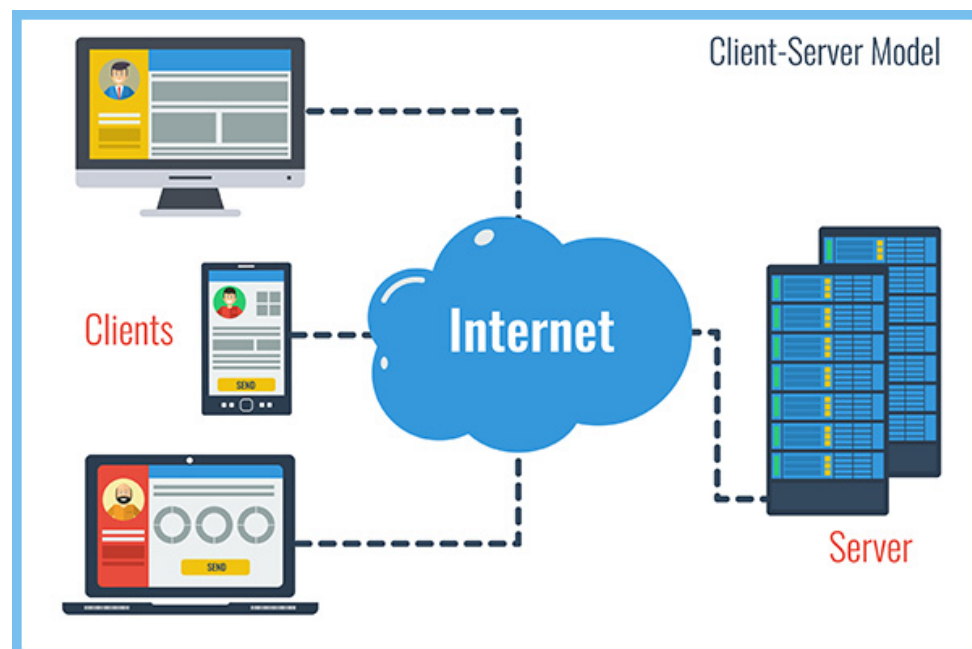
acrescenta suporte a tecnologias corporativas baseadas em objetos distribuídos, a exemplo dos EJBs (Enterprise Java Beans).

Cientes HTTP

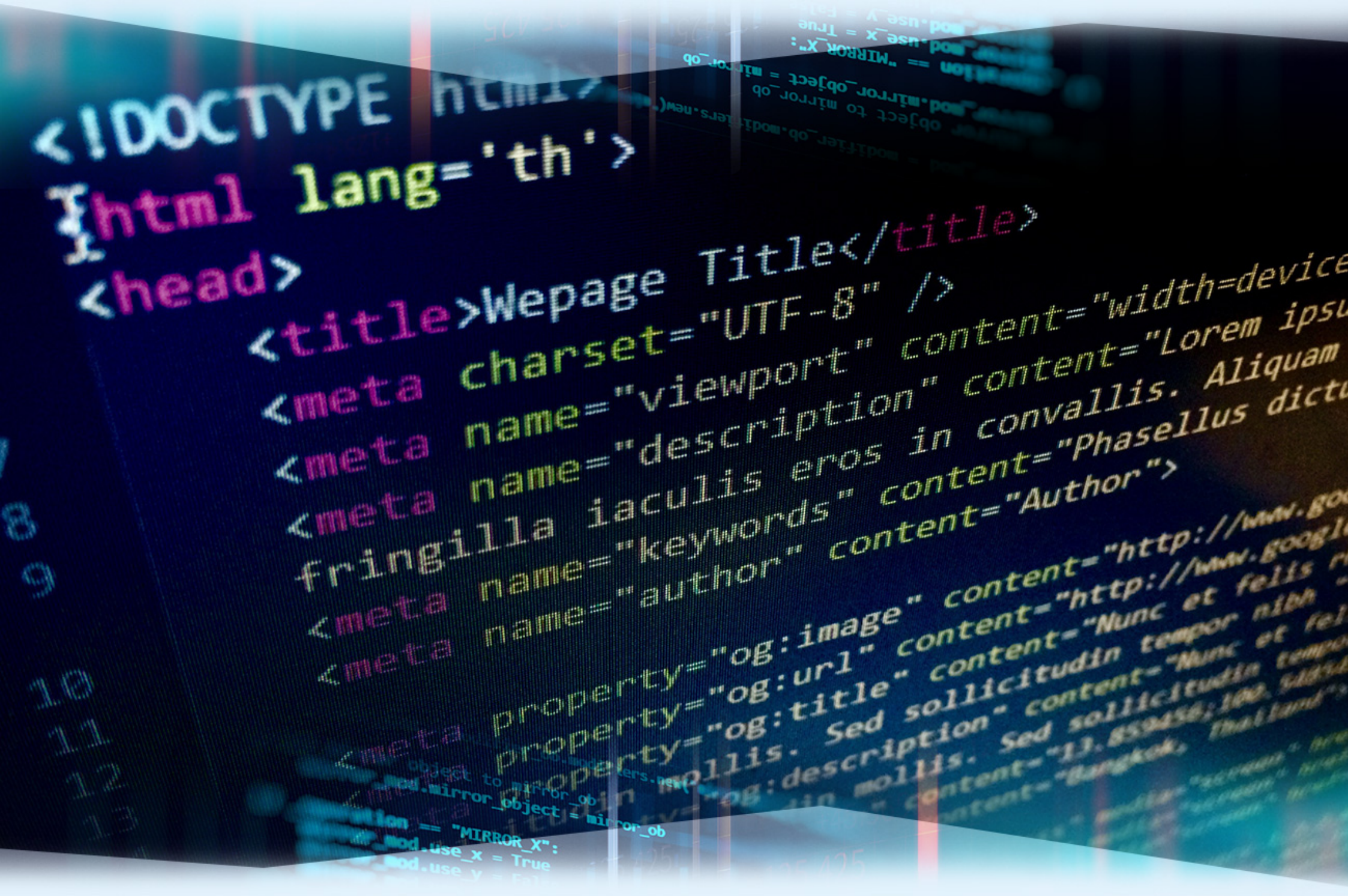
Um cliente HTTP é tipicamente denominado “navegador”, sendo comum o seu uso no nosso dia a dia, a exemplo do Google Chrome, Opera, FireFox, Microsoft Edge, além de navegadores específicos dos diversos dispositivos móveis.

A linguagem HTML foi adotada como padrão para a criação de páginas na Internet, e um navegador deve ser capaz de interpretar estas páginas, exibindo o conteúdo com as formatações corretas e permitindo a navegação entre as mesmas através de hiperlinks.

É importante notarmos que diferentes dispositivos apresentam dimensões e recursos variados, o que pode trazer problemas quanto à exibição da informação. Fato este que acabou trazendo à tona um termo que se tornou comum no design de páginas: a responsividade.



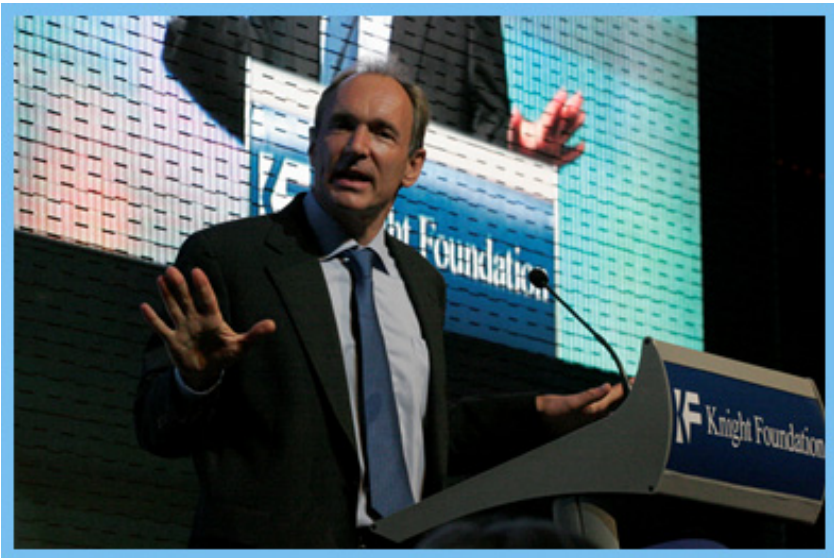
Responsividade é a capacidade de exibir a mesma informação em diferentes dispositivos, adequando o formato da exibição aos mesmos, ou seja, uma interface responsiva pode ser acessada por diferentes plataformas sem deformações que inviabilizem sua utilização.



Linguagem HTML 5

1991

Foi formalizada a linguagem HTML, do inglês Hypertext Markup Language, baseada nas propostas de Tim Berners-Lee, físico britânico que visava obter um meio de divulgação de suas pesquisas entre seus colegas.



1996

Inicialmente controlada pela IETF (Internet Engineering Task Force), passou a ter suas especificações mantidas pela W3C (World Wide Web Consortium) a partir de 1996, sendo a recomendação HTML 4.01 publicada em 1999.



2000

A W3C passa a recomendar a sua nova especificação XHTML, a qual traz uma sintaxe mais rígida para o HTML, baseando-se nas regras do XML (Extended Markup Language).

2008

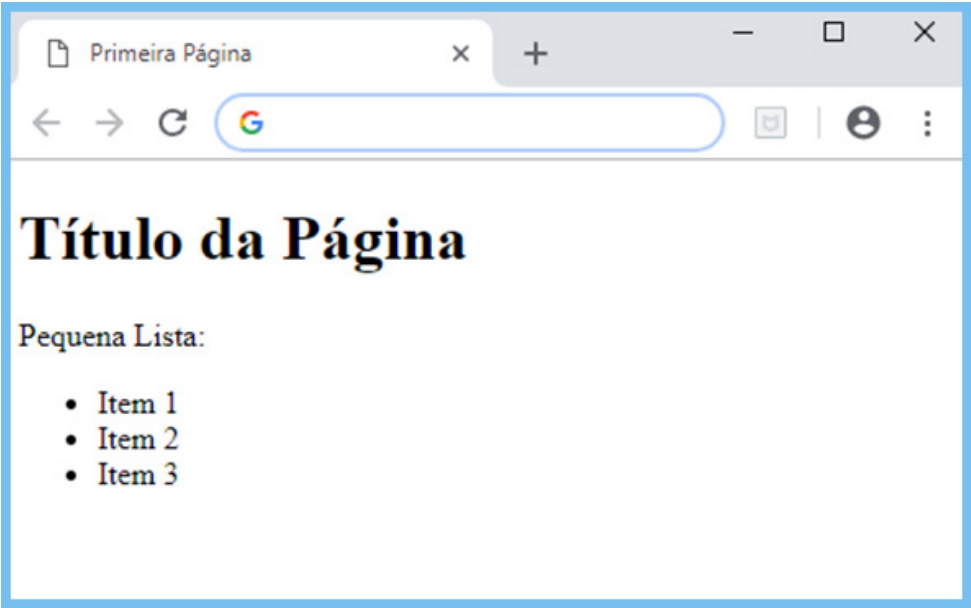
Finalmente, a partir da primeira publicação pela W3C, surge a especificação do HTML 5, oficializada em **2014**, e tendo como característica principal um **novo mecanismo de extensibilidade que permite combinar diferentes sintaxes baseadas em XML dentro de uma mesma página.**

Como nas versões iniciais, o HTML 5 é uma linguagem de marcação baseada em etiquetas (tags) que devem controlar diversos aspectos do texto.

Nas versões iniciais do HTML ocorria o controle tipográfico e estrutural, mas no **HTML 5 a preocupação é apenas estrutural**, sendo delegado para o CSS (Cascade Style Sheet) o controle de características tipográficas.

Exemplo

```
<!DOCTYPE html>
<html>
  <head>
    <title>Primeira Página</title>
    meta charset="UTF-8"/>
  </head>
  <body> <!-- PARTE VISUAL -->
    <h1>Título da Página</h1>
    <p>
      Pequena Lista:
      <ul>
        <li>Item 1</li>
        <li>Item 2</li>
        <li>Item 3</li>
      </ul>
    </p>
  </body>
</html>
```



Na primeira linha, podemos observar o comando `<!DOCTYPE html>`, que não faz parte da sintaxe HTML, mas do XML, definindo um domínio de utilização para tags. No caso, esta linha indica que teremos, em seguida, um documento HTML.

Todo documento é iniciado com a tag **<html>**, que serve para delimitar o início e final do documento, englobando as tags , responsável por informações gerais de pré-carga, e , correspondente ao conteúdo visual da página.

Diversas tags aceitam a definição de atributos específicos, como **<meta charset="UTF-8"/>**, e estes atributos devem ter seus valores colocados entre aspas ou apóstrofes.

Também devemos observar a possibilidade de adicionar comentários, os quais são iniciados com `<!--` e finalizados com `-->`.

Conjunto de tags

A seção **<head>** aceita um pequeno conjunto de tags, sendo as principais apresentadas a seguir:

Tag	Utilização
<title>	Define o título da página, normalmente apresentado no topo da janela
<meta>	Permite a inclusão de informações e controle de características do documento, como o uso de UTF-8 para acentuação da língua portuguesa
<link>	Efetua a ligação da página com um recurso externo, como folha de estilo CSS ou biblioteca JavaScript

Quanto à seção **<body>**, por cuidar da parte estrutural da visualização, pode conter um grande conjunto de tags, algumas delas voltadas para o formato de títulos, áreas de escrita e listas, outras para a inclusão de imagens e elementos diversos, além de comandos de navegação e formulários, entre diversas outras.

Podemos observar algumas destas tags a seguir:

Tag	Utilização
<h1> <h2> <h3> <h4> <h5> <h6> <h7>	Definem elementos de texto utilizados como títulos em diferentes tamanhos de fonte, sendo <h1> para o maior e <h7> para o menor.
 <i> <u> 	Negrito, itálico e outros efeitos de texto. São consideradas obsoletas, sendo recomendado o uso de folhas de estilo CSS.
<p> <div>	Definem trechos de texto, sendo que <p> ressalta o parágrafo e <div> é muito utilizado para a criação de camadas.
 	Iniciam a criação de listas de elementos, sendo numeradas com .
	Define um item de lista.
<hr/>	Linha divisória na horizontal.
 	Quebra de linha.

“index.html” – a página inicial

Quando construímos um site, estamos criando um conjunto de páginas, em que pode ocorrer a navegação entre as mesmas.

Todo site tem uma página inicial, que, nos casos mais comuns, recebe o nome “**index.html**”.

A partir da página inicial, podemos navegar para outras páginas do mesmo domínio, ou até para outros sites, acessar recursos específicos, como vídeo e áudio, entre outras diversas formas de ligação com elementos externos.

TEXTO DO LINK

Os **hiperlinks** serão nosso ferramental básico de navegação entre páginas, sendo criados com a tag **<a>**, ou âncora (do inglês *anchor*), sempre apontando para alguma URL (*Universal Resource Locator*), que pode ser definida como uma identificação de algo localizado na rede, como um site, uma página, um trecho da página, ou qualquer outro elemento identificado de forma unívoca.

Exemplo

```
<!DOCTYPE html>
<html>
  <head>
    <title>Alguns Links</title>
    <meta charset="UTF-8"/>
  </head>
  <body>
    <h1>Alguns exemplos de links</h1>
    <p>
      <ul>
        <li><a href="//www.estacio.br">Site da Estacio</a>
        <li><a href="//portal.estacio.br/graduacao.aspx">Cursos de Graduação</a></li>
      </ul>
    </p>
  </body>
</html>
```

Tabela

Outro elemento interessante do HTML padrão é a tabela.

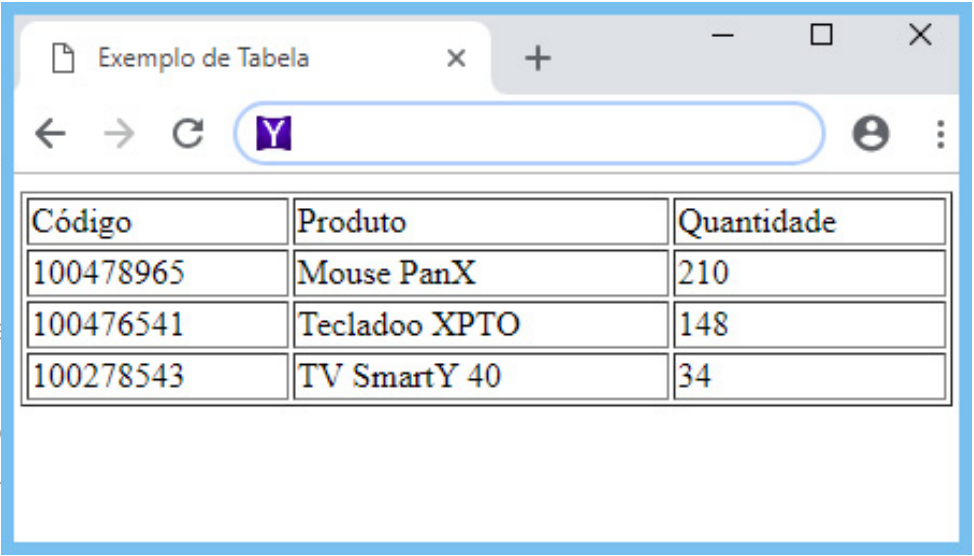
Uma tabela exibe dados organizados em termos de linhas e colunas a partir de três tags:

Tag	Utilização
<table>	Define uma tabela e suas características gerais, como espessura de borda e espaçamento entre colunas.
<tr>	Inicia uma linha da tabela.
<td>	Define uma célula na linha criada anteriormente.

Exemplo

Podemos observar a criação de uma tabela simples no código a seguir:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Exemplo de Tabela</title>
    <meta charset="UTF-8"/>
  </head>
  <body>
    <table border="1" width="100%">
      <tr><td>Código</td><td>Produto</td><td>Quantidade</td></tr>
      <tr><td>100478965</td><td>Mouse PanX</td><td>210</td></tr>
      <tr><td>100476541</td><td>Teclado XPTO</td><td>148</td></tr>
      <tr><td>100278543</td><td>TV SmartY 40</td><td>34</td></tr>
    </table>
  </body>
</html>
```



Código	Produto	Quantidade
100478965	Mouse PanX	210
100476541	Teclado XPTO	148
100278543	TV SmartY 40	34

Atenção

Hoje em dia ouvimos falar bastante do **padrão Tableless**, erroneamente interpretado como “sem tabelas”. Na verdade, este padrão estipula que as tabelas não devem ser utilizadas para organizar visualmente o conteúdo da página, o que passou a ser feito com camadas, mas permite que elas sejam utilizadas livremente para a exposição de dados “tabulares”, como planilhas.

Dica

Editores de Código HTML

Embora não seja imprescindível, com certeza vamos preferir utilizar um editor de texto que ressalte os comandos HTML quando estivermos criando nossas páginas.

Uma sugestão gratuita e de fácil utilização é o [NotePad++ <https://notepad-plus-plus.org/download/>](https://notepad-plus-plus.org/download/).

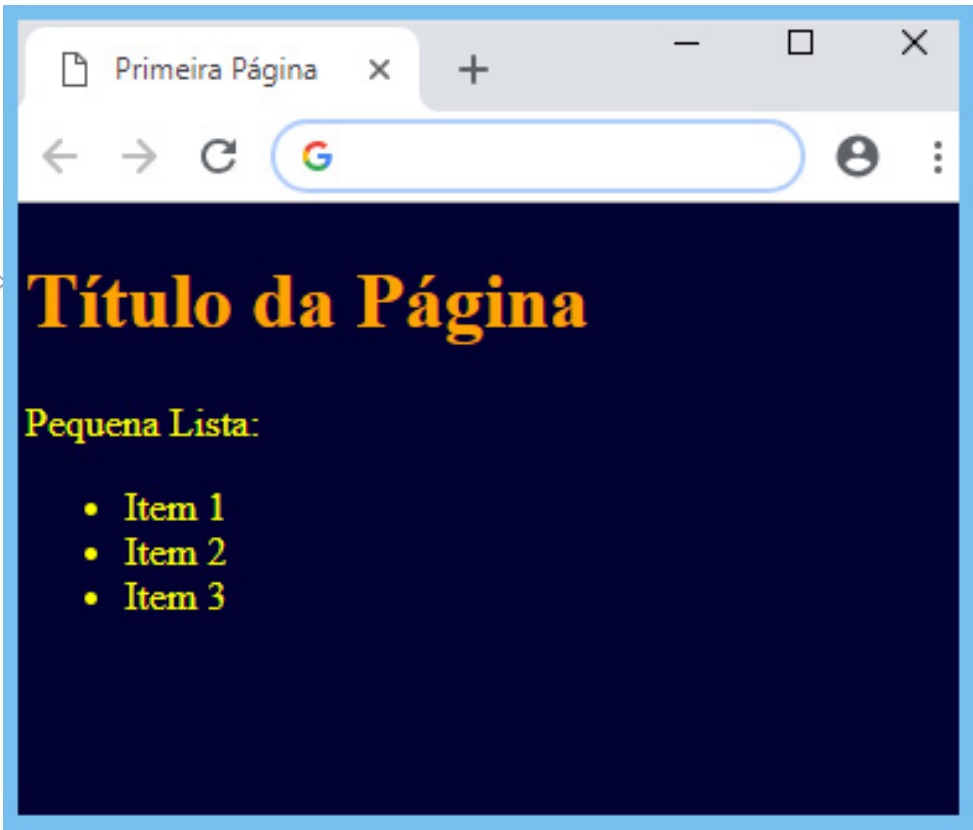
CSS – Cascade Style Sheet

TEAgora que nossos textos já podem ser estruturados com o uso de HTML, devemos nos preocupar com a formatação de fontes e demais elementos, o que era feito antigamente através de tags, mas que atualmente é viabilizado pelo uso das **folhas de estilo (CSS)**.

Exemplo

Podemos observar a criação de uma tabela simples no código a seguir:


```
<!DOCTYPE html>
<html>
  <head>
    <title>Primeira Página</title>
    <meta charset="UTF-8"/>
    <style>
      h1 {color: orange}
      body {background-color: #000032; color:yellow}</style>
    </head>
    <body>
      <h1>Título da Página</h1>
      <p>
        Pequena Lista:
        <ul>
          <li>Item 1</li>
          <li>Item 2</li>
          <li>Item 37</li>
        </ul>
      </p>
    </body>
  </html>
```



É fácil entender como o CSS possibilita modificar completamente o aspecto original de uma página. Elementos como fontes e cores, alinhamentos, figuras de fundo, entre vários outros, podem ser configurados, sempre seguindo a sintaxe **propriedade: valor**.

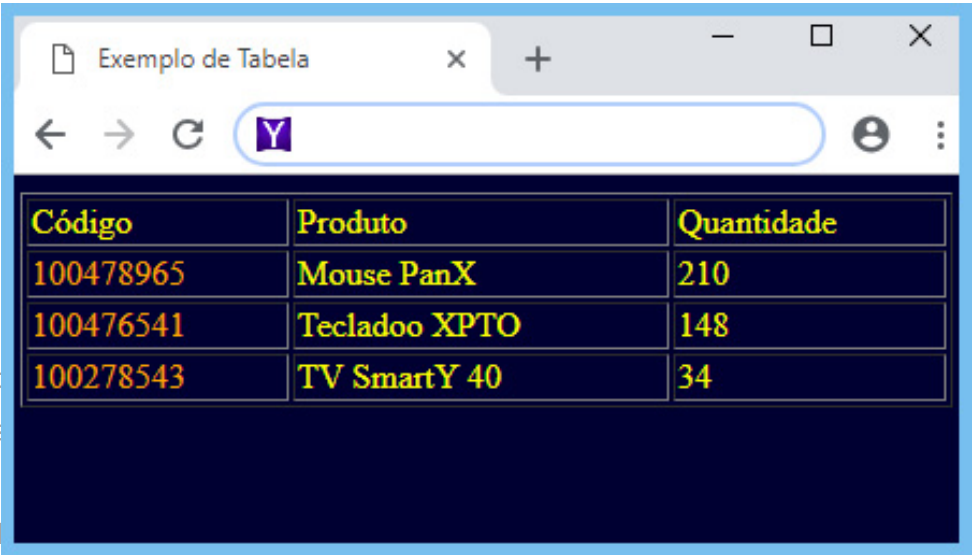
Outro elemento a se considerar é o elemento ao qual será aplicado o estilo, também chamado de **seletor**, podendo ser, inicialmente, uma tag, uma classe ou um Id (identificador).

No exemplo anterior, a tag **<h1>** foi formatada com a fonte de cor laranja, enquanto **<body>** recebe fundo de tonalidade azul (RGB=#000032) e fonte amarela.

Exemplo

Para criarmos classes devemos utilizar um ponto antes do nome, e aplicar às tags de interesse através do atributo class, como podemos observar a seguir:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Exemplo de Tabela</title>
    <meta charset="UTF-8"/>
    <style>
      body {background-color: #000032; color:yellow}
      .enfoque {color:orange}
    </style>
  </head>
  <body>
    <table border="1" width="100%">
      <tr><td>Código</td><td>Produto</td><td>Quantidade</td></tr>
      <tr><td class="enfoque">100478965</td><td>Mouse PanX</td><td>210</td></tr>
      <tr><td class="enfoque">100476541</td><td>Tecladoo XPTO</td><td>148</td></tr>
      <tr><td class="enfoque">100278543</td><td>TV SmartY 40</td><td>34</td></tr>
    </table>
  </body>
</html>
```



Devemos fazer algo similar para o uso de identificadores (Ids), porém utilizaremos o símbolo de **hash (#)**, e aplicaremos a elementos que devem ser unicamente identificados na página. **Isso quer dizer que não devo ter duas tags com o mesmo Id.**

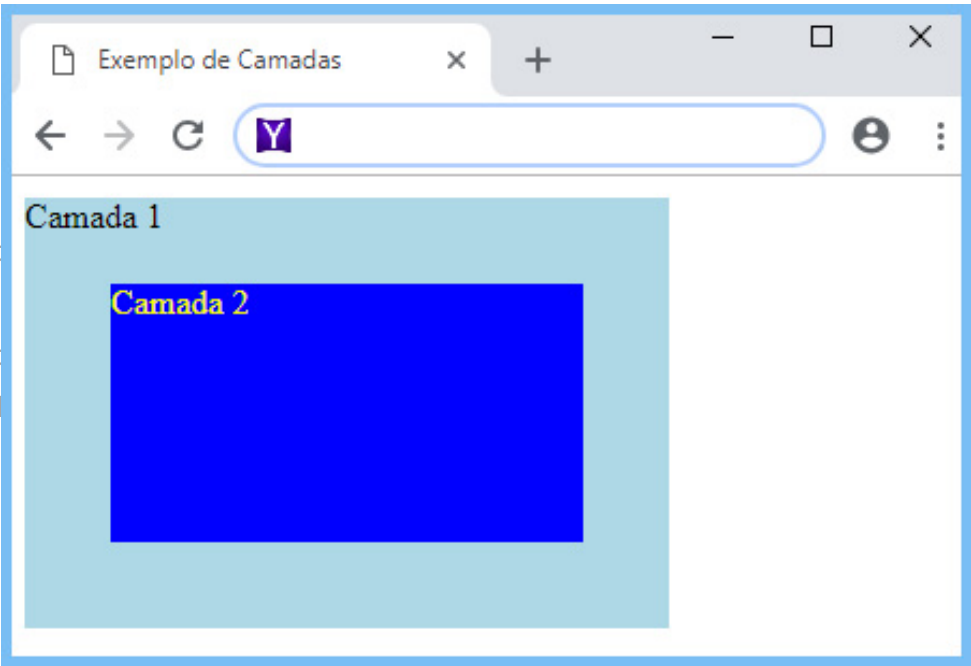
O uso de Ids é muito comum na construção de **camadas**, as quais servem como padrão para organização dos elementos de uma página, segundo o atual padrão vigente do Tableless.

Exemplo

Nós podemos observar a criação de duas camadas no exemplo:

- C1, a 10 pixels do topo e 10 pixels da lateral esquerda, com altura de 200 pixels e largura de 300 pixels, adotando azul claro como cor de fundo; e
- C2, a 50 pixels do topo e 50 pixels da lateral esquerda, com altura de 120 pixels e largura de 220 pixels, utilizando azul como cor de fundo e amarelo como cor de fonte.

```
<!DOCTYPE html>
<html>
  <head>
    <title<Exemplo de Camadas>/title>
    <meta charset="UTF-8"/>
    <style>
      #C1{position:absolute; top:10px; left:10px; width:200px; height:200px; background-color:lightblue}
      #C2{position:absolute; top:50px; left:50px; width:120px; height:120px; background-color:blue; color:yellow}
    </style>
  </head>
  <body>
    <div id="C1">Camada 1</div>
    <div id="C2">Camada 2</div>
  </body>
</html>
```



Atenção! Aqui existe uma videoaula, acesso pelo conteúdo online

Formulários

Quando efetuamos o cadastro em algum site de compras pela Web, o que estamos fazendo é o preenchimento de um formulário HTML.

Estas informações são enviadas para o servidor de duas formas, denominadas “**métodos**”:



GET

Os dados são enviados através da própria URL, ficando expostos na chamada.



POST

Os dados são enviados em background, o que evita a exposição, mas não determina segurança, já que os pacotes podem ser capturados por ferramentas do tipo Sniffer.

Para que possamos transmitir os dados com maior segurança devemos utilizar **SSL** (Secure Socket Layer) com um certificado digital de qualidade, o que fará com que os dados transitem de forma criptografada.

Quando fazemos uma chamada **https** estamos utilizando um ambiente deste tipo.

Todo formulário é iniciado com a tag **<form>**, que terá como principais atributos o método de envio (**method**) e o destino para a informação (**action**).

Dentro do formulário, podemos incluir os componentes de entrada de dados, como:

- Caixas de texto;
- Botões de rádio;
- Caixas de marcação.

A grande maioria definida através da tag **<input>**, cujos atributos principais são:

- **type**, referente ao tipo de componente;
- **name**, utilizado para identificação do dado na chegada ao lado servidor;
- **value**, para determinar o valor de envio ou o valor de preenchimento inicial.

Tipos utilizados pela tag <input>

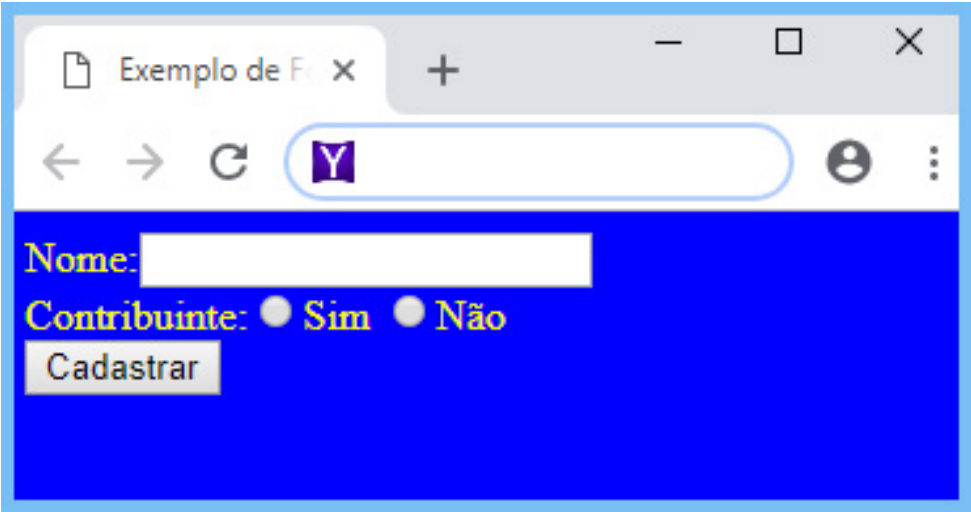
Os tipos que a tag **<input>** pode utilizar são explicados no quadro abaixo:

Tipo	Utilização
text	Cria uma caixa de texto, podendo ter um valor inicial definido através de value.
hidden	Quando queremos guardar um valor para envio, mas sem ficar visível.
radio	Opções mutuamente exclusivas. Quando marcamos um, todos com o mesmo atributo name serão desmarcados. O que ficar marcado, ao final, enviará seu atributo value como dado para o servidor.
checkbox	Quando queremos a possibilidade de marcar várias opções. Todos que forem marcados enviarão seus atributos value como dados para o servidor
submit	Cria um botão de envio para o formulário
reset	Cria um botão que reinicia (limpa) o formulário

Exemplo

Podemos observar, a seguir, um formulário simples, com o uso das tags <form> e <input>.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Exemplo de Formulário</title>
    <meta charset="UTF-8"/>
    <style>
      body {background-color:blue; color:yellow}
    </style>
  </head>
  <body>
    <form action="//servidor/app.aspx" method="get">
      Nome:<input name="N1" type="text"/><br/>
      Contribuinte:
      <input type="radio" name="CONTR" value="S"/>Sim
      <input type="radio" name="CONTR" value="N"/>Não
      <br/>
      <input type="submit" value="Cadastrar"/>
    </form>
  </body>
</html>
```



Neste exemplo, se escrevermos o nome XPTO e selecionarmos a opção “Sim”, ao clicarmos em **Cadastrar**, verificaremos a seguinte URL na barra de navegação:

`//servidor/app.aspx?N1=XPTO&CONTR=S`

Atenção

O trecho ressaltado após a interrogação corresponde ao que chamamos de **Query String**, onde estarão os dados, no formato nome=valor, separados pelo sinal **&**.
Caso modificássemos o método para **“post”**, a informação enviada não ficaria visível para o usuário, mas vale ressaltar que estaria transitando em um pacote sem criptografia na rede para o caso do HTTP comum.

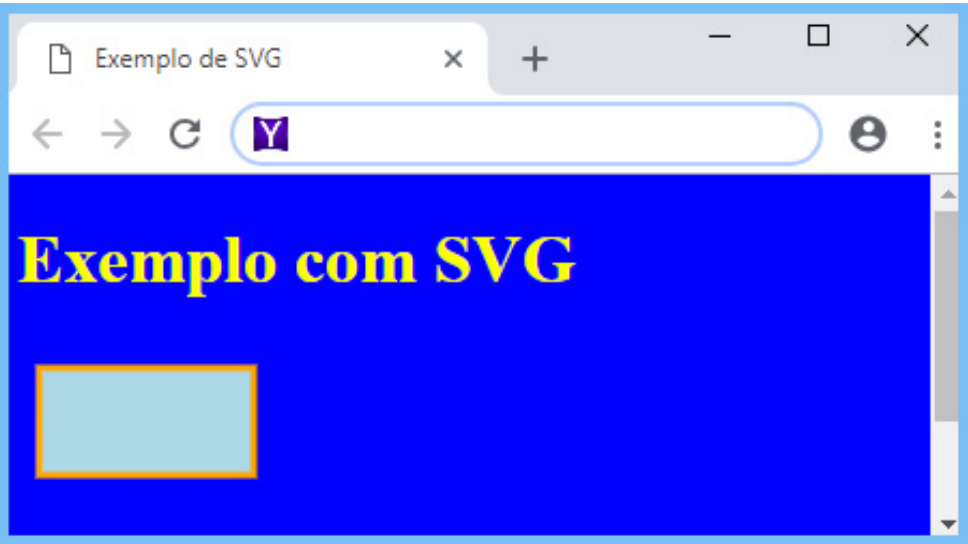
Sintaxes Diversas no HTML 5

Uma grande evolução do HTML 5 em relação às versões anteriores é a possibilidade de utilizar outras linguagens, como SVG e VRML, em meio ao código HTML, o que permite expandir a funcionalidade do ambiente básico.

Exemplo

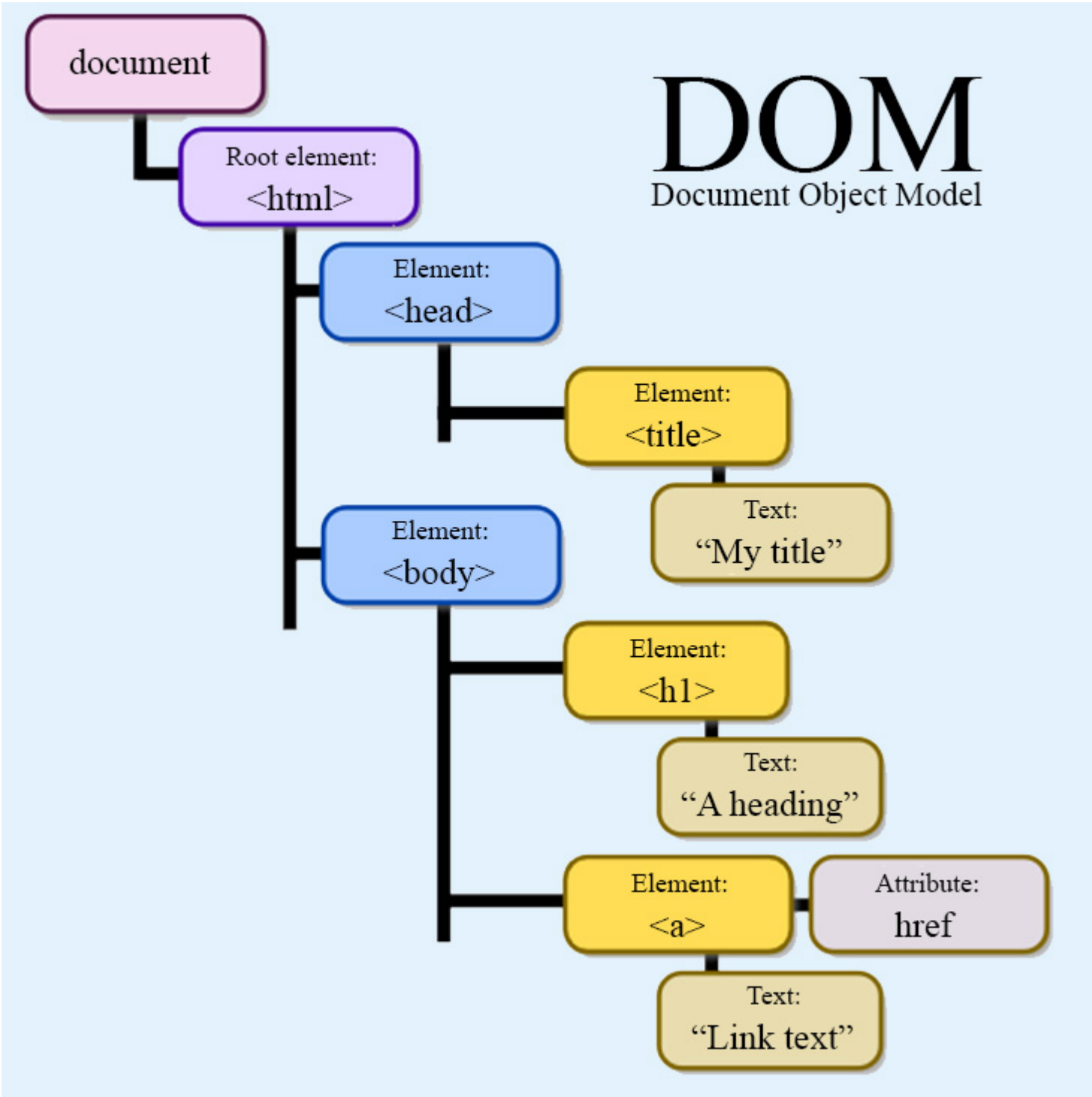
Podemos observar, a seguir, um exemplo de utilização de SVG em uma página HTML5.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Exemplo de SVG</title>
    <meta charset="UTF-8"/>
    <style>
      body {background-color:blue; color:yellow}
    </style>
  </head>
  <body>
    <h1>Exemplo com SVG</h1>
    <svg>
      <rect width="100" height="50" x="10" y="10" fill="lightblue" stroke="orange" />
    </svg>
  </body>
</html>
```



DOM

O Document Object Model, ou **DOM**, é uma API que nos permite acessar os elementos de um documento HTML ou XML, proporcionando uma visualização hierárquica do mesmo no formato de árvore, baseado na forma como as tags são aninhadas.



Com o uso de DOM ficou muito mais fácil criarmos páginas compatíveis com navegadores diversos, principalmente no que se refere à integração da página HTML com o código JavaScript.

Após acessar o componente da página, os valores de seus atributos podem ser consultados e modificados com grande facilidade e de forma padronizada para os diversos ambientes.

O processo mais comum para acessar este componente é através de seu Id, com o uso do comando a seguir, que será estudado quando tratarmos da linguagem JavaScript:

```
document.getElementById( Id )
```

Como as tags são organizadas hierarquicamente, a partir de um nó raiz, devemos considerar a existência de diferentes tipos de nós ao longo desta árvore, como elementos, ou tags, atributos e textos.

Os principais tipos de nós são apresentados na tabela:

Tipo de Nó	Propriedade nodeName	Propriedade nodeValue
Element	Nome do Elemento (ou tag)	null
Attr	Nome do atributo	Valor do atributo
Text	#text	Conteúdo do nó
CDATASection	#cdata-section	Conteúdo do nó
Comment	#comment	Texto do comentário

Através da definição estrutural do DOM e manuseio das informações fornecidas pelos nós da árvore constituída, podemos acessar, modificar e até acrescentar elementos à página HTML, o que permite grande dinamismo e possibilidade de modificações parciais em um documento durante a visualização do mesmo, viabilizando um dos pilares do AJAX, tecnologia que será tratada posteriormente nesta disciplina.

Atenção! Aqui existe uma videoaula, acesso pelo conteúdo online

Atividade

1 - Os Web Servers são programas servidores que respondem ao HTTP, sendo que alguns deles oferecem suporte a tecnologias de objetos distribuídos e são normalmente classificados como Application Servers.

Dentre os servidores apresentados a seguir, qual deles NÃO pode ser considerado nativamente como um Application Server?

- a) BEA Weblogic
 - b) Tomcat
 - c) WebSphere
 - d) GlassFish
 - e) JBoss
-

2 - Hoje em dia a formatação tipográfica das páginas não é mais feita no código HTML, que assumiu um papel estruturante com relação ao conteúdo da página, e as folhas de estilo CSS passaram a ser a tecnologia adotada para esta formatação.

Qual seria o comando CSS para formatar o fundo de uma camada com id “centro” na cor amarela?

- a) .centro { background-color: yellow }
 - b) #centro { color: yellow }
 - c) *centro { background: yellow }
 - d) #centro { background-color: yellow }
 - e) .centro{ color:yellow }
-

3 - A utilização de DOM viabiliza a interpretação do conteúdo XML ou HTML no formato de árvore (hierárquico), onde cada nó da árvore apresenta diversas informações, as quais podem ser expressas em termos das propriedades nodeName e nodeValue.

Para um nó do tipo Element, qual a informação fornecida por nodeValue?

- a) Nome do Atributo
 - b) Conteúdo do Nó
 - c) Valor Nulo
 - d) Texto de Comentário
 - e) Valor do Atributo
-

Referências

CASSATI, J. P. **Programação cliente em sistemas web**. Rio de Janeiro: Estácio, 2016.

DEITEL, P; DEITEL, H. **Ajax, Rich Internet Applications e Desenvolvimento Web para Programadores**. São Paulo: Pearson Education, 2009.

PLOTZE, R. **Tecnologias Web**. Rio de Janeiro: Estácio, 2016.

SANTOS, F. **Tecnologias para Internet II**. Rio de Janeiro: Estácio, 2017.

Próxima aula

- Características do JavaScript;
- Sintaxe básica do JavaScript;
- Criação de funções e bibliotecas com uso de JavaScript.

Explore mais

- [RFC do protocolo HTTP <https://tools.ietf.org/html/rfc2616>](https://tools.ietf.org/html/rfc2616)
- [Introdução ao HTML 5 <https://www.w3schools.com/html/html5_intro.asp>](https://www.w3schools.com/html/html5_intro.asp)
- [Tutorial de CSS <https://www.w3schools.com/css/>](https://www.w3schools.com/css/)