



Linguagem de marcação de hypertexto - HTML

Os conceitos de HTML, linguagem de marcação e estrutura básica de páginas web.

Prof. Alexandre de Oliveira Paixão

Propósito

Computador com um editor de texto que dê suporte à marcação HTML. Para o sistema operacional Windows, é indicado o Notepad++; para o Linux, o Nano Editor.

Objetivos

- Reconhecer a estrutura de uma página web.
- Empregar as tags na HTML.
- Empregar as tags complementares às tags básicas em uma página web.
- Aplicar um formulário em uma página web.

Introdução

Assista ao vídeo e entenda o objetivo principal de uma linguagem HTML, que é estruturar o conteúdo de um documento. Este conteúdo pode ser composto de textos, figuras, tabelas etc. Para realizar a estrutura, que pode ser definida como a organização dos elementos de conteúdo, a HTML faz uso de um sistema hierárquico de elementos chamados de tags.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Princípio da HTML

O objetivo principal de uma linguagem de marcação de hipertexto, mais especificamente, da HTML, que será usada como linguagem padrão e alvo de nosso estudo, é estruturar o conteúdo de um documento. Esse conteúdo pode ser composto de textos, figuras, tabelas etc.

Uma estrutura, por sua vez, pode ser definida como a organização dos elementos de conteúdo. Para realizar essa função, a HTML faz uso de um sistema hierárquico de elementos chamados de tags.

A linguagem HTML

Assista ao vídeo e entenda o que é HTML, a origem dessa linguagem, as suas versões e os tipos de documentos.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

O que é HTML?

A linguagem de marcação de hipertexto (HTML) é uma linguagem de marcação formada por elementos conectados (textos, imagens, vídeos etc.), cujo principal objetivo é organizar o conteúdo e estabelecer a estrutura básica de uma página web.

Como tudo começou

A especificação da HTML foi criada a partir da junção de dois padrões: o SGML e o HyTime.

O SGML (*standard generalized markup language*) é um padrão ISO que especifica as regras para a criação de linguagens de marcação que sejam independentes de plataforma.

Já o **HyTime** é um padrão desenvolvido pela ISO com o objetivo de possibilitar que hiperdocumentos de hipermídia pudessem ser descritos em função de sua estrutura e conteúdo e, consequentemente, utilizados por quaisquer sistemas conformantes (NEWCOMB; KIPP; NEWCOMB, 1991).

HyTime

Hypermedia/Time-Based Structuring Language.

Inicialmente, a especificação da HTML e a sua evolução foram mantidas pelo IETF (Internet Engineering Task Force), desde sua primeira publicação formal, em 1993. Entretanto, a partir de 1996, passou a ser mantida pelo **World Wide Web Consortium (W3C)**, além de ter ganhado o status de norma internacional (ISO/IEC) em 2000.

World Wide Web Consortium (W3C)

É um consórcio internacional de organizações filiadas e que conta com equipe em tempo integral responsável por, junto com o público em geral, desenvolver padrões para a web.

Evolução da HTML: versões

Ao longo dos anos, a HTML tem passado por uma série de aperfeiçoamentos. Esse processo é liderado pelo W3C. Desde a sua criação, a HTML obteve diferentes versões, tais como: HTML, HTML 2.0, HTML 3.2, HTML 4.01, XHTML, HTML 5, HTML 5.1 e HTML 5.2.

Veja o histórico de versões da HTML!

HTML

1991

Tim Berners-Lee cria a HTML.

HTML 2

1995

Grupo de Trabalho HTML define a HTML 2.0.

HTML 3.2

1997

W3C publica a Recomendação HTML 3.2.

HTML 4.01

1999

W3C publica a Recomendação HTML 4.01.

XHTML

2000

W3C publica a Recomendação XHTML 1.0.

HTML5

2014

W3C publica a Recomendação HTML5.

HTML 5.1

2016 / 2017

HTML 5.2

2017

W3C publica a Recomendação HTML5.2.

- W3C candidata a Recomendação HTML 5.1.
- W3C publica a Recomendação HTML 5.1 2ª edição.

Cada versão lançada da HTML representa algum tipo de evolução em relação à versão anterior: formalização de regras e características; correção de problemas de compatibilidade; criação de novas tags; suporte a novos recursos; entre outras mudanças.

Definição de tipos de documento (DTD)

O ponto inicial da estrutura de uma página web é o tipo de documento a ser utilizado. Inicialmente, a HTML continha apenas tags para **marcações simples**, uma vez que ainda não existia um dispositivo com interface gráfica no qual pudesse ser exibida.

Com isso, as primeiras tags eram relacionadas a elementos como títulos, parágrafos, quebras de linha etc. Entretanto, com a sua rápida expansão e sobretudo pelo lançamento do primeiro navegador, o Mosaic, surgiu a necessidade de criação de novas tags, para **marcações mais complexas**.

Essa necessidade se tornou contínua, tendo como principais pilares os seguintes eventos: a rápida expansão da web e a construção de novos navegadores.

O fato de que cada navegador implementava seus próprios padrões, incluindo tags que só podiam ser interpretadas por eles mesmos, tornou necessária a adoção de padrões reconhecidos por todos. Por isso, o W3C criou a definição de tipos de documentos (DTD) ou *document type definition*, uma lista de elementos, atributos e requisitos relacionados a determinada linguagem ou variação da linguagem.

Utilizando doctypes: document types

A importância dos document types na criação de páginas HTML diz respeito ao conjunto de tags que podem ser usadas e que serão ou não **renderizadas** a partir do tipo utilizado.

Renderizadas

É o processo realizado pelo navegador que consiste em ler e interpretar as tags HTML e apresentá-las adequadamente.

Antes da HTML5, existia um conceito chamado **concordância de documento**.

Embora relevante apenas nas versões anteriores da HTML, é importante abordá-lo, uma vez que muitas páginas web ainda não foram portadas para a HTML5. Isso significa que cada página precisa ser concordante com a especificação W3C oficial para a qual foi escrita.

A especificação da HTML 4, por exemplo, previa três tipos de doctypes: strict, transitional e frameset. Vamos conhecer cada um desses tipos?



Exemplos de tags na HTML.

Strict

É responsável por determinar uma série de restrições e exigências: obrigatoriedade de separação entre a estrutura e a apresentação; limitação nos elementos de apresentação disponíveis - como tags de fonte e atributos de alinhamento -; ausência de suporte a tags obsoletas etc.

Transitional

É mais maleável em relação aos atributos de apresentação, além de possibilitar a utilização de tags obsoletas.

Frameset

É caracterizada pela tag obsoleta da HTML4, usada para combinar múltiplos frames e exibi-los em uma única página.

Veja a seguir o fragmento do código inicial de uma página HTML 4.01:

Na declaração, são definidas as seguintes informações sobre a página HTML:

- Tipo de Documento: HTML.
- Acesso à DTD: Public – logo, a DTD a que se refere está disponível para uso de forma pública.
- Onde se encontra o DTD: local em que foi criado e está hospedado no W3C.
- URL da especificação: o endereço <http://www.w3.org/TR/html4/loose.dtd> leva à especificação em questão.

Portanto, o doctype não é uma tag HTML, mas, sim, uma **instrução**. É uma declaração que serve para informar ao navegador qual a versão da HTML usada em um arquivo HTML. A partir da HTML5, quando a HTML deixou de ser baseada na SGML, a declaração do doctype foi simplificada, como visto a seguir.

Comparando as declarações do doctype entre a HTML5 e a HTML4, notamos algumas diferenças. A principal diferença é que a HTML5, ao contrário das versões anteriores, não se baseia na SGML, portanto não requer referência a um DTD.

Atividade 1

Questão 1

Com relação aos tipos de documento, doctypes, marque a afirmativa correta.

- A O doctype não é importante nem obrigatório. Portanto, sequer precisa ser declarado. Sendo assim, o que de fato importa são as tags utilizadas no documento.
- B A especificação W3C (contendo as tags suportadas, entre outros padrões) para cada versão de HTML perde sua importância a partir do lançamento de uma nova versão.
- C A combinação entre a definição de um doctype e a utilização das tags suportadas por ele permite a correta renderização de uma página web.
- D Nunca será possível renderizar uma página web que contenha tags definidas como obsoletas pelo W3C.
- E A principal semelhança entre a HTML5 e a HTML4 é o fato de que ambas requerem o uso de referências a um doctype, pelo motivo de serem baseadas na SGML.



A alternativa C está correta.

O doctype é uma instrução obrigatória e permite que seja informado ao navegador ou dispositivo o tipo de documento a ser carregado, permitindo assim a correta renderização da página.

Estrutura de uma página web

Quando falamos de estrutura de uma página HTML, estamos nos referindo aos elementos – ou tags – que irão compor o documento em questão. Entre esses elementos, alguns são considerados obrigatórios para a correta renderização da página nos navegadores. Além desses, há outros elementos que possuem papéis distintos e são usados para diferentes fins, como criação de tabelas, listas, inserção de links etc. Esse conjunto de elementos, obrigatórios e básicos, forma a estrutura de uma página web escrita com HTML.

Assista ao vídeo e entenda quais são os elementos da estrutura de uma página web, como os elementos obrigatórios, a raiz, o cabeçalho, o corpo, os elementos básicos e o funcionamento da HTML.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Elementos obrigatórios

Como vimos, a estruturação de uma página web tem início na definição do doctype a ser utilizado. Em seguida, é necessário inserir um conjunto de tags que, com o doctype, formam a sua estrutura básica, apresentada a seguir.

python

Vamos entender o que significa cada tag?

Raiz

A tag inicial, após a declaração do doctype, é a `<html>`. Esse elemento é a **raiz do documento**. Logo, lembrando que o código HTML é composto por uma estrutura hierárquica de tags, podemos dizer que a `<html>` é a raiz da árvore do documento. Outra característica importante é que uma tag deve ser aberta e fechada.

A declaração é a abertura, e o seu fechamento ocorre com a declaração `</html>`. Com isso, observando o código anterior, é possível perceber que as demais tags estão contidas dentro da tag raiz, `<html>`.



Comentário

Cabe destacar outro elemento importante referente à tag raiz: o atributo `lang`. Esse atributo global (por ser declarado na tag raiz) deve ser utilizado para definir o idioma da página – podendo também ser aplicado a outras tags, definindo o idioma delas. Veja alguns exemplos de sua declaração: `<html lang="en-US">` `<html lang="en-GB">` `<html lang="pt">`

Cabeçalho

A tag é chamada de cabeçalho do documento. Em uma analogia ao corpo humano, devemos vê-la como a “cabeça”, uma vez que contém as tags que serão usadas para manter todo o documento funcionando e em ordem.

Vejam agora as tags que fazem parte do cabeçalho e entenderemos para que servem. Vamos lá?

1

`<title>`

É o título do documento, sendo visualizado na barra de título do navegador.

2

`<meta>`

Engloba uma série de informações – comumente chamadas de metainformações – como a descrição da página, palavras-chave etc.

3

`<script>`

É responsável pela inclusão e/ou definição de scripts relacionados ao documento.

4

`<link>`

É responsável pela inclusão de folhas de estilo (externas) relacionadas ao documento. Também possibilita a inclusão de favicons (pequenos ícones que aparecem na barra de endereços do navegador).

5

`<style>`

É responsável pelo vínculo de folhas de estilo ao documento – quando elas são declaradas diretamente no documento.

Corpo

Logo abaixo do cabeçalho, deve ser inserida a tag , responsável pela estruturação do documento, sobretudo de seu conteúdo, e também pela apresentação. Apesar disso, recomenda-se que a apresentação do documento seja feita por meio de folhas de estilo (CSS).

Quando navegamos em um website, todo o conteúdo que vemos – textos, imagens e demais elementos – está contido na tag .

Elementos básicos

Embora as páginas web sejam diferentes umas das outras, elas tendem a compartilhar uma mesma estrutura. Isso significa que, além da estrutura apresentada anteriormente, uma página web também é composta por seções, sendo as mais comuns:

1

Cabeçalho

É a área inicial, logo na parte superior da página. Normalmente, nesta seção vemos título, logomarca ou algo do tipo.

2

Barra de navegação

É uma área que pode ser horizontal e/ou vertical, e contém os links para navegação pelas seções/páginas do site.

3 Conteúdo

É o conteúdo principal da página, o qual pode ser dividido em subseções.

4

Barra lateral

É possível conter tanto links de navegação quanto informações adicionais que forneçam complemento ao conteúdo principal.

5

Rodapé

É a área final da página, na qual normalmente são dispostos links, avisos de direitos autorais, informações complementares sobre o proprietário do site etc.

Funcionamento da HTML

Após concluído, o documento estruturado, composto por tags, deve ser salvo com a extensão “html” ou “htm”. Assim, ele estará pronto para ser lido por um dispositivo, como um navegador web – em um computador pessoal ou em um smartphone. Quando uma página web é exibida no navegador, o que vemos é o seu conteúdo: texto, imagens etc., e não o código ou as tags. Esse processo consiste no que chamamos de **interpretação e renderização**.

Atividade 2

Questão 1

Assinale a alternativa que contém o conjunto de tags que forma a estrutura considerada obrigatória em uma página HTML.

A <html> e <body>

B <html>, <body>, <header> e <footer>

C <DocType>, <html> e <body>

D <DocType>, <html>, <head>, <body> e <footer>

E <DocType>, <html>, <head> e <body>



A alternativa E está correta.

A estrutura considerada obrigatória, para a correta renderização de uma página web nos navegadores, é composta pelas tags <DocType>, <html>, <head> e <body>.

Criando a primeira página HTML

Após termos visto a estrutura de uma página HTML, chegou a hora de colocarmos a mão na massa para vermos, na prática, como funciona um documento HTML.

Assista ao vídeo e entenda o roteiro da atividade descrita logo a seguir. Observe, ainda, os aspectos importantes e implícitos ao conhecimento necessário para a criação de uma página HTML.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Roteiro de prática

Nesta prática, seguindo o roteiro, você codificará sua primeira página HTML. Para isso, além dos conhecimentos vistos anteriormente, será necessário usar as seguintes ferramentas: um editor de textos – pode ser o próprio Bloco de Notas do Windows ou o Nano Editor do Linux, ou então algo um pouquinho mais avançado, como o software (livre/gratuito) Notepad++, e também de um navegador – Google Chrome, Firefox, MS Edge etc.

Siga o passo a passo:

1. Com o seu editor de textos aberto, insira a estrutura básica demonstrada no código da imagem “Estrutura básica de uma página web”. Feito isso, salve seu documento com a extensão “html” ou “htm”. Agora, basta clicar duas vezes sobre o documento criado para que ele seja exibido no navegador.
2. Agora que você já viu o seu primeiro código HTML no navegador – na verdade, você viu uma página em branco – avance um pouco: lembre o que aprendeu até aqui, insira uma tag Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec congue nisl a magna tempus dignissim. Curabitur eget porta lacus, eu bibendum leo. Mauris enim lorem, convallis eget sodales ullamcorper, semper eu purus. Aenean vitae varius ex. Maecenas sagittis diam quis tincidunt blandit. Donec id egestas nisl. Nullam sollicitudin arcu quis risus dignissim eleifend. Morbi ipsum augue, aliquet eget ante ac, feugiat interdum nisl.

Faça você mesmo!

Questão 1

Na estrutura de elementos que compõem uma página web, há uma hierarquia na qual alguns elementos devem estar contidos dentro de outros elementos “pai”. No código visto anteriormente, temos como exemplo disso a tag <title>, posicionada como filho da tag <head>. Com base nessa informação, crie uma cópia da página HTML codificada anteriormente e remova as tags <head> e </head>. Com isso, a tag <title>...</title> ficará sem um pai. Salve esse novo documento e o abra no navegador. Sobre o resultado obtido, é correto afirmar que:

A

A página web é exibida corretamente no navegador, mesmo que um elemento obrigatório, como a tag <head></head> seja omitido.

B

Ao removermos um elemento obrigatório de uma página web, esta só é exibida corretamente no navegador devido ao cache que tais ferramentas possuem.

C Apenas o navegador Google Chrome é capaz de exibir corretamente uma página web que não possui um elemento obrigatório.

D A versão mais atual do HTML dispensou o uso de elementos como o doctype, head, entre outros.

E Além da tag <head>, a tag <title> também poderia ser removida do documento HTML e ainda assim seu valor ser exibido no navegador.</p>



A alternativa A está correta.

A renderização correta de uma página no navegador, mesmo sem a presença de um elemento obrigatório, como no caso da tag <head>, é possível graças aos mecanismos de renderização dos navegadores, que conseguem interpretar o valor de algumas tags – em nosso caso a <title>, mesmo que o documento não esteja formatado adequadamente.</p>

Princípio da HTML

A HTML, desde a sua criação, no início de 1990, vem passando por uma série de transformações, em uma constante evolução. A versão inicial da HTML se baseou em uma outra linguagem, o SGML, herdando desta algumas de suas primeiras tags, como as de título (h1, h2, etc.), parágrafo (p), entre outras.

Como principal diferença entre essas linguagens, vemos, na HTML, já em sua primeira versão, um dos princípios do funcionamento da web: a interligação entre documentos – sendo essa relação feita com o uso da tag a partir do seu atributo “href”.

Ao longo dos anos, novas tags foram adicionadas ao HTML, assim como outras foram descontinuadas. A seguir, veremos mais sobre algumas das tags básicas da HTML.

Usando tags HTML

Assista ao vídeo e entenda como declarar tags, a composição de uma tags, tags textuais e semânticas e tags de formatação.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Aprofundando o conceito de tags

Como já aprendemos, a HTML é uma linguagem de marcação que utiliza tags para estruturar o seu conteúdo. Mas o que são tags?

Tags são palavras, escritas entre os caracteres de menor “<” e maior “>” e que servem para informar ao navegador a estrutura e o significado do conteúdo inserido em uma página web.

Um dos exercícios propostos anteriormente consistia em inserir na estrutura básica de tags da HTML – mais precisamente dentro da tag – um texto qualquer e, em seguida, salvar e carregar essa página no navegador. Como pôde ser visto, o navegador exibiu o texto, que provavelmente era curto.

Agora, tente imaginar o site de um portal de notícias com todo o conteúdo amontoado dentro da tag . Embora exibido pelo navegador, tudo seria uma grande bagunça, uma confusão! Agora, imagine que seja possível dizer ao navegador que determinado conteúdo é o título de uma seção, sendo procedido de um parágrafo que diga respeito a tal seção. Você ainda pode ir além, inserindo uma imagem e vários outros elementos. Para isso servem as tags!

Como declarar tags

Antes de conhecermos mais a fundo as tags, a sua composição e os seus tipos, vamos voltar à sua criação.

Primeiro, devemos ter em mente que existe um conjunto já definido de tags, padronizado pelo W3C. Logo, não é possível criarmos as nossas próprias tags HTML. Outro ponto importante é o modo como uma tag é escrita: como já sabemos, entre os sinais de menor e maior. Lembre-se, a tag deve,

obrigatoriamente, ser fechada! Para isso, repetimos a declaração da tag e utilizamos uma barra "/" antes do seu nome.

Observe os exemplos:

Título de uma seção

```
<h1> Título de uma  
seção </h1>
```

Parágrafo contendo um texto

```
<p> Parágrafo contendo um texto </p>
```

Com base no primeiro exemplo, dizemos que a tag foi iniciada ao inserirmos o `<h1>` e fechada em `</h1>`.

Existem algumas tags que fogem a essa regra. Por exemplo, a tag de quebra de linha:

. Repare que, nesse caso, a tag é fechada com a utilização da "/" logo antes do sinal de maior.



Atenção

A tag de quebra de linha tornou-se obsoleta a partir da HTML5.

Tipos e composição das tags

As tags podem ser divididas nos seguintes tipos de acordo com as suas funções: estruturais, textuais e semânticas. Outra característica importante é o fato de que as tags também podem ter atributos, a respeito dos quais falaremos a seguir.

Atributos

Servem para que algumas características sejam adicionadas a um elemento, a uma tag. São compostos por um nome e por um valor.

Exemplo

```

```

Essa tag é utilizada para a inserção de imagens no documento. Temos dois exemplos de atributos em sua declaração: "src" e "alt", que são nomes de atributo; e "imagem.jpg" e "minha imagem" são seus valores, respectivamente.

O atributo "src" define o endereço e o nome da imagem. Já o atributo "alt" define um texto alternativo a ser exibido no navegador caso a imagem não seja carregada.

Há uma enorme variedade de atributos, assim como de relacionamentos entre esses atributos e as tags. O site do W3C contém a lista completa de atributos e combinações. Por ora, cabe destacar dois atributos de extrema importância no desenvolvimento web:

ID

Utilizado para definir um identificador, que deve ser único, para uma tag em um documento.

Class

Utilizado para definir uma classe à qual uma ou mais tags pertencem.

Com base nesses dois tipos de identificação, é possível, por exemplo, fazer referência a um ou mais atributos para inserirmos estilização visual nas páginas, por meio de folhas de estilo ou eventos e interação, usando o Javascript.

Tipos de tags: textuais e semânticas

Agora, vamos conhecer os tipos de tag textuais e semânticas.

- **Tags textuais:** São responsáveis por organizar o conteúdo da página, ou seja, textos, mídias e links, por exemplo. Algumas das principais tags textuais são: `<p>`, `<h1>`, ..., `<h6>`, `` e `<a>`. Essas tags e suas funções serão descritas a seguir.
- **Tags semânticas:** A partir da HTML5, foram inseridas tags com a função semântica de organizar a estrutura de conteúdo de uma página.

Logo, voltando ao exemplo de seções básicas de uma página, nossa página ficaria da seguinte maneira:



Conteúdo interativo

Acesse a versão digital para ver mais detalhes da imagem abaixo.

```
1  <!DOCTYPE html>
2
3  <html>
4
5      <head>
6          <title>Título da Página</title>
7      </head>
8
9      <body>
10
11          <header>
12              Cabeçalho da Página
13          </header>
14
15          <nav>
16              Barra de Navegação
17          </nav>
18
19          <main>
20              Conteúdo da Página
21
22              <aside>
23                  Barra Lateral
24              </aside>
25          </main>
26
27          <footer>
28              Rodapé
29          </footer>
30
31      </body>
32
33  </html>
```

Tags estruturais básicas de uma página web.

Ao analisar a imagem, é possível perceber que existem tags específicas para cada seção do conteúdo. Essa é uma característica importante da HTML, chamada de semântica, que, nesse contexto, pode ser considerada a correta utilização de uma tag HTML de acordo com o seu conteúdo ou a sua finalidade.

Em muitos casos, ao visualizarmos uma página no navegador, é indiferente o fato de as tags serem utilizadas para guardar determinado conteúdo. Entretanto, recomenda-se usá-las de acordo com sua função. Esse uso cria uma organização no documento que facilita tanto para você, que o escreveu, quanto para outras pessoas que venham a editar o mesmo documento. Além disso, muitos dispositivos fazem uso dessa marcação para uma correta interpretação do conteúdo ali contido.

Vamos agora conhecer outras tags e suas respectivas funções!

1

`<article>`

Inclui um bloco de conteúdo que deve ser usado quando se deseja inserir um artigo, como um post de um blog, por exemplo.

2

`<section>`

Define uma seção no documento. É normalmente utilizado para agrupar seções. Por exemplo: uma `<section>` poderia conter vários `<articles>`.

3

`<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>` e `<h6>`

Apresenta a inserção de títulos ao longo da página. É uma boa prática que cada seção ou cada bloco de texto seja precedido por um título. Trata-se de uma tag hierárquica, logo, no título mais importante, deve ser usada a tag `<h1>`; no subtítulo seguinte, `<h2>`, e assim por diante. Além disso, uma página poderá conter várias `<h1>`, `<h2>`... `<h6>`. A exceção aqui fica por conta da tag `<h1>`, uma vez que é recomendado utilizar apenas uma em uma mesma página.

4

`<p>`

Aponta a inserção de parágrafos de texto.

5

`<pre>`

Aponta a inserção de um texto pré-formatado.

6

`<div>`

Aponta o agrupamento de algum tipo de conteúdo que não tenha nenhuma semântica específica ou que não se encaixe bem dentro de uma tag semântica, embora não seja considerada semântica.

7

``

Apresenta semelhança com a `<div>`. Entretanto, enquanto a `<div>` é um elemento não semântico no bloco (quando usada, quebra o conteúdo em uma seção), a `` é embutida (não quebra o conteúdo, apenas o agrupa).

8

`<a>`

Aponta a inserção de links.

9

`
`

Aponta a inserção de uma quebra de linha.

10

`<hr>`

Aponta a inserção de uma linha horizontal no documento. Normalmente, é utilizada quando se pretende alterar a temática de um texto.

Tags de formatação `` e `<bold>` vs `` e ``

A última versão HTML ainda dá suporte a algumas tags direcionadas à formatação visual de conteúdo. Embora possam ser substituídas por CSS, quatro dessas tags merecem atenção especial:

1

``

Aplica o efeito de negrito em um texto.

2

``

Aplica o efeito de negrito em um texto e o marca como importante.

3

`<i>`

Aplica o efeito de itálico em um texto.

4

``

Aplica o efeito de itálico e dá ênfase a um texto.

O destaque que tais elementos merecem diz respeito à função semântica que as tags **e exercem**.

É interessante notar que ambas produzem o mesmo efeito visual das tags `` e ``, respectivamente, em um texto, ou seja, marcam em negrito e/ou itálico. Entretanto, há uma diferença importante entre essas tags que vai além da visualização do texto no navegador pela maioria dos usuários. A função semântica é perceptível em dispositivos de leitura de tela, que transformam o texto em áudio, e normalmente são utilizados por pessoas com deficiência visual.

Tags obsoletas

A cada nova versão da HTML, novas tags são criadas, assim como antigas são descontinuadas. As tags descontinuadas, chamadas de obsoletas, embora ainda possam ter suporte em boa parte dos navegadores, devem ser evitadas pelos seguintes motivos:

- Porque, provavelmente, foram substituídas por novas tags, com melhor semântica.*
- Por causa do risco de desconfigurarem o conteúdo da página, uma vez que os browsers podem deixar de suportá-las a qualquer momento.*

Veja alguns exemplos de tags obsoletas!

1

`<applet>`

Identifica a inclusão de um applet Java.

2

`<center>`

Centraliza horizontalmente o conteúdo de um bloco.

3

`<dir>`

Apresenta container para lista de diretórios ou arquivos.

4

``

Determina características relacionadas a fontes de um determinado elemento.

5

`<image>`

Apresenta relação ancestral com a tag ``, que era usada nas primeiras versões HTML para a inserção de imagens.

Atualmente consideradas obsoletas, as tags

e ainda são usadas em muitas páginas HTML. Ambas se enquadram no conceito de que não é função da HTML cuidar da apresentação. Logo, as duas foram substituídas por propriedades CSS.

Atividade

Questão 1

Em termos de semântica, na utilização de tags, assinale a alternativa correta.

A O conceito de semântica pode ser resumido como a escolha e utilização de tags de acordo com o tipo de conteúdo ou função.

B Na prática, o que importa não é a amarração entre uma tag e seu conteúdo, e sim o resultado final, ou seja, independente das tags usadas, o importante é o que será exibido no navegador.

C A criação de novas tags com teor semântico na HTML5, como <header> e <footer>, teve como finalidade apenas dar um novo nome às tags já existentes e amplamente utilizadas, já que a maioria das páginas web possuía um cabeçalho e um rodapé.

D Um código HTML não pode ser dito semântico caso não faça uso de todas as tags definidas na HTML5.

E Tags ditas semânticas têm por característica organizar textos, mídias e links da página HTML.



A alternativa A está correta.

A HTML semântica tem como objetivo principal descrever o significado do conteúdo contido em um documento HTML a fim de torná-lo mais claro para os dispositivos que processam o documento e também para o programador que o criou ou precisará modificá-lo.

Utilizando TAGs

Como já vimos, uma página web é composta por uma série de tags, cada uma exercendo um papel específico. Para fixarmos toda a teoria vista até aqui, vamos realizar algumas atividades, nas quais usaremos algumas das tags HTML estudadas.

Aplicando as tags na prática

Assista ao vídeo e confira a proposta da confecção de uma página HTML com as tags: tags doctype, html, head, title e bod. Veja também um exemplo e como abrir a página no navegador.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Roteiro de prática

Siga os passos necessários para realização dessa prática. Para isso, você precisará de um editor de textos – pode ser o próprio Bloco de Notas do Windows ou o Nano Editor do Linux, ou então algo um pouquinho mais avançado, como o software (livre/gratuito) Notepad++ e de um navegador – Google Chrome, Firefox, MS Edge etc.

- Em seu editor de textos, crie a estrutura básica de uma página HTML (tags doctype, html, head, title e body). Salve o documento com a extensão "html" ou "htm".
- Dentro da tag body, insira um título principal, seguido de um parágrafo. Aqui você deverá utilizar as tags HTML adequadas para tais funções.
- Por fim, insira uma linha de separação horizontal no documento, procedida de um subtítulo e de um novo parágrafo de texto. Ao final, salve o documento e visualize o resultado no navegador.

Como resolução, teremos o seguinte código:

```
python
```

Primeiro título

```

    Lorem ipsum dolor sit amet, consectetur adipiscing elit.
    Donec congue nisl a magna tempus dignissim.
    Curabitur eget porta lacus, eu bibendum leo.
    Mauris enim lorem, convallis eget sodales ullamcorper, semper eu
purus.
    Aenean vitae varius ex. Maecenas sagittis diam quis tincidunt
blandit.
    Donec id egestas nisl. Nullam sollicitudin arcu quis risus
dignissim eleifend.
    Morbi ipsum augue, aliquet eget ante ac, feugiat interdum nisl.
```

Subtítulo

```

    Lorem ipsum dolor sit amet, consectetur adipiscing elit.
    Donec congue nisl a magna tempus dignissim.
```

Faça você mesmo!

Questão 1

Dentre as várias tags HTML existentes, algumas exercem papel voltado para a formatação visual de conteúdo. Mesmo sabendo que tal formatação deve ser sempre realizada pelo CSS, algumas tags continuam sendo importantes e devem ser utilizadas em ocasiões especiais – sobretudo no que tange a questões de acessibilidade. Nesse sentido, observe o texto:

Atenção, ao clicar nesse link você concordará com os termos e condições de utilização do serviço contratado.

Note que uma das palavras foi destacada no texto. Qual o código HTML necessário para reproduzir o texto?

A `<p>Atenção, ao clicar nesse link você concordará com os termos e condições de utilização do serviço contratado.</p>`

B `<p>Atenção, ao clicar nesse link você concordará com os termos e condições de utilização do serviço contratado.</p>`

C `<p><i>Atenção</i>, ao clicar nesse link você concordará com os termos e condições de utilização do serviço contratado.</p>`

D `Atenção, ao clicar nesse link você concordará com os termos e condições de utilização do serviço contratado.`

E `<p>Atenção, ao clicar nesse link você concordará com os termos e condições de utilização do serviço contratado.</p>`



A alternativa E está correta.

Nas alternativas, duas delas, em termos visuais, poderiam ser utilizadas para exibição do texto alvo. Entretanto, considerando o aspecto de acessibilidade mencionado no enunciado, a alternativa "e" contém o código correto, uma vez que a tag ``, nas ferramentas de leitura de tela, dá a ênfase necessária ao texto nela contido, diferentemente da tag ``. Além disso, cabe ressaltar que devemos, sempre que possível, utilizar a tag `<p>` para conter os textos em uma página web.

Listas, tabelas e mídias em HTML

Você já deve conhecer as tags HTML básicas e ter criado páginas web simples usando esses elementos. No HTML, existem diversas outras tags mais avançadas que podem ser usadas para diferentes fins, como exibição de listas e tabelas ou/e inserção de vídeos e áudios. E é justamente esse o tópico que trataremos a seguir.

Aplicando as tags avançadas

Assista ao vídeo e confira o conceito de lista, tabelas e a estrutura sintática das tags relacionadas a esses objetos. Veja também como inserir mídias na estrutura da página.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Listas na HTML

O HTML fornece suporte para a representação visual de três tipos de listas: as ordenadas, as não ordenadas e as de definição. Vamos conhecer cada uma dessas listas a seguir.

Ordenadas

Usadas quando desejamos listar dados com a necessidade de representar a sua ordenação de forma numérica ou alfabética.

Não ordenadas

Usadas quando não há necessidade de listar ordenadamente.

De definição

Usadas quando precisamos listar itens e atribuímos uma descrição a eles.

Outra característica importante das listas se deve ao fato de que a sua marcação HTML é composta por uma tag de container, ou tag “pai”, e por seus itens, ou “filhos”. Além disso, a lista de definição possui ainda um terceiro item, que é justamente o utilizado para descrevê-lo. Veja na tabela cada tipo de lista:

Tipo	Tag container	Tag item	Tag descrição
Ordenadas			--
Não ordenadas			--

Tipo	Tag container	Tag item	Tag descrição
Definição	<dl>	<dt>	<dd>

Tabela: Tags utilizadas para a representação de Listas.
Alexandre de Oliveira Paixão.

Agora acompanhe exemplos de como as tags são utilizadas e seu resultado na tela:

Lista ordenada

<pre> Primeiro Item da lista Segundo Item da lista Terceiro Item da lista </pre>	Resultado: <ol style="list-style-type: none"> 1. Primeiro Item da lista 2. Segundo Item da lista 3. Terceiro Item da lista
--	--

Lista não ordenada

<pre> Primeiro Item da lista não ordenada Segundo Item da lista não ordenada Terceiro Item da lista não ordenada </pre>	Resultado: <ul style="list-style-type: none"> • Primeiro Item da lista não ordenada • Segundo Item da lista não ordenada • Terceiro Item da lista não ordenada
---	--

Lista de definição

<pre><dl> <dt> Primeiro Item da lista não ordenada </dt> <dd> Descrição do primeiro elemento </dd> <dt> Segundo Item da lista não ordenada </dt> <dd> Descrição do segundo elemento </dd> <dt> Terceiro Item da lista não ordenada </dt> <dd> Descrição do terceiro elemento </dd> </dl></pre>	Resultado: <p>Primeiro Item da lista não ordenada Descrição do primeiro elemento</p> <p>Segundo Item da lista não ordenada Descrição do segundo elemento</p> <p>Terceiro Item da lista não ordenada Descrição do terceiro elemento</p>
--	--

Cabe destacar, em relação às tags demonstradas anteriormente, que tanto o símbolo de ordenação (a numeração 1,2,3) quanto o símbolo visual (o bullet) podem ser alterados com a utilização de CSS.



Atenção

A HTML permite que sejam criadas listas dentro de listas – as chamadas listas aninhadas (nested lists).

Tabelas na HTML

Quando precisamos lidar com dados tabulares em uma página web, utilizamos tabelas. As tabelas usadas nesse documento são exemplos do tipo de dado e também da apresentação obtida ao utilizarmos tais elementos na HTML.

Estrutura das tabelas

A marcação HTML relacionada às tabelas contém outras tags além da principal, a saber: `<table>`. A tag principal define o container geral.

De forma hierárquica, temos as tags para a definição de linhas `<tr>` e colunas `<td>`. Com essas três tags é possível representarmos uma tabela simples. Entretanto, há tags adicionais que podem ser usadas e que ajudam a melhor organizar o conteúdo.

O esquema seguinte apresenta as tags de marcação, e suas respectivas funções, relacionadas às tabelas:

1

`<table>`

Representa container principal da tabela.

2

`<tr>`

Representa as linhas, sendo composta pelas tags relacionadas às colunas.

3

`<td>`

Representa as colunas e precisa ser inserida dentro da tag de linha.

4

`<th>`

Representa colunas e é usada para exibir o título de uma coluna, possuindo, neste sentido, função semântica. Assim como a tag `<td>`, deve estar contida em uma tag de linha. Esse elemento apresenta estilos próprios, sendo renderizado de forma diferente das colunas comuns.

5

`<thead>`

Armazena o cabeçalho da tabela, sendo composto por linhas e colunas. Esse elemento, a exemplo do que vimos anteriormente, tem função semântica em termos de estruturação de conteúdo.

6

`<tfoot>`

Armazena o rodapé da tabela, tendo também função semântica em termos de estruturação.

As tabelas, normalmente, são organizadas de maneira uniforme: linhas sobre linhas, colunas após colunas, célula ao lado de célula. Logo, as colunas costumam ter a mesma largura, assim como as linhas a mesma altura. Entretanto, há situações nas quais é preciso mudar um pouco essa organização. Por exemplo, pode ser necessário mesclar duas colunas ou mesmo duas ou mais linhas.

Para isso, fazemos uso de alguns atributos, que têm função de destaque ao tratarmos das tags relacionadas às tabelas. São eles:

Linha

Rowspan

Coluna

Colspan

Esses atributos têm a função de expandir as linhas ou colunas. Sua declaração é acompanhada de um número que indica a quantidade de células a serem utilizadas para expansão da linha ou coluna. Ao final deste módulo, teremos a oportunidade de praticarmos um pouco mais e veremos como esses atributos funcionam.

As tabelas e a semântica

Desde a criação do código HTML, as tabelas foram o elemento mais utilizado fora de sua função semântica. Isso se deve ao fato de que a estrutura básica de uma página HTML lembra muito a estrutura de uma tabela: cabeçalho, rodapé, seções (linhas) etc. Logo, foi prática comum ao longo de muitos anos a codificação de páginas web completas fazendo-se uso de tabelas. Além de ir de contra a semântica, tal uso traz consigo outros problemas, como o peso das páginas e a redução da acessibilidade, entre outros.

Mídias na HTML: vídeo e áudio

O suporte à multimídia na HTML vem melhorando ao longo dos anos e versões. Com o advento da HTML5, tornou-se possível, de forma muito simples, incorporar vídeo e áudio em uma página web. Para isso, são usadas as tags `<video>` e `<audio>`, veja:

python

Como visto, os atributos também têm suma importância ao fazermos uso das tags de vídeo e áudio. No exemplo acima, temos, inicialmente, o atributo “src” – que informa o endereço, podendo ser de um site ou mesmo de um arquivo local – no seu computador ou no servidor em que a página web fica hospedada.

Os atributos “controls”, “autoplay” e “loop”, que são específicos para esse tipo de mídia em questão, fornecem suporte ao controle do conteúdo (vídeo ou áudio) incorporado pelo navegador, além de definirem alguns comportamentos, como o “autoplay” e o “loop”, que são autoexplicativos.

Os três atributos aqui descritos são apenas uma prévia, uma vez que há alguns outros disponíveis. Outros componentes importantes, contidos na especificação do HTML5, são os eventos que permitem o controle de mídia embutida com a utilização de Javascript. São os chamados “Media Events”.

Atividade

Questão 1

As linhas e colunas de uma tabela HTML podem ser modificadas (mescladas) para permitir uma organização não uniforme. Sobre as tabelas, é correto afirmar que:

- A Para modificar a organização de uma tabela, basta inserir novas tabelas aninhadas.
- B Os atributos “colspan” e “rowspan” são utilizados para modificar a organização de colunas e linhas em uma tabela.
- C A organização de linhas e colunas de uma tabela pode ser modificada com a utilização de tags HTML como as listas, por exemplo.
- D As tags HTML colspan e rowspan, quando inseridas dentro das tags, permitem modificar a organização das linhas e colunas da tabela, mesclando o seu conteúdo.
- E A modificação da organização de linhas de uma tabela se dá por meio da deleção de linhas existentes e inserção de novas linhas.



A alternativa B está correta.

Os atributos têm papel importante dentro da HTML. Combinados às tags, eles informam como elas devem se comportar.

Criando tabelas HTML

As tabelas são um recurso importante dentro do HTML, apesar de, por muitas vezes, controverso. Como já mencionado, esse elemento já foi utilizado para fins de agrupamento de conteúdo, embora sua real finalidade seja a de exibir dados tabulares. Para melhor fixar os elementos e a estrutura de uma tabela, vamos realizar algumas atividades práticas.

Confira, neste vídeo, a atividade proposta no roteiro a partir de exemplos e seus respectivos resultados.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Roteiro de prática

Como vimos, um documento HTML pode conter tanto tabelas simples quanto mais complexas. Tendo como base o arquivo HTML utilizado nas práticas anteriores, chegou a hora de codificarmos. No editor de texto, comece criando uma nova seção no seu HTML e insira:

Como resolução, teremos o seguinte código:

```
<!-- Tabela Simples -->
<h2>Tabela Simples</h2>
<table width="100%" border="1" >
  <tr>
    <td>Alex</td>
    <td>HTML</td>
    <td>10</td>
  </tr>
  <tr>
    <td>Ana</td>
    <td>HTML</td>
    <td>10</td>
  </tr>
</table>

<!-- Tabela Com Título -->
<h2>Tabela Com Título</h2>
<table width="100%" border="1" >
  <tr>
    <th>Aluno</th>
    <th>Disciplina</th>
    <th>Nota</th>
  </tr>
  <tr>
    <td>Alex</td>
    <td>HTML</td>
    <td>10</td>
  </tr>
  <tr>
    <td>Ana</td>
    <td>HTML</td>
    <td>10</td>
  </tr>
</table>

<!-- Tabela Com Cabeçalho e Rodapé -->
<h2>Tabela Com Cabeçalho e Rodapé</h2>
<table width="100%" border="1" >
  <thead>
    <tr>
      <th>Mês</th>
      <th>Horas/Aula</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Junho</td>
      <td>40</td>
    </tr>
    <tr>
      <td>Julho</td>
      <td>30</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td colspan="2">Total</td>
      <td>70</td>
    </tr>
  </tfoot>
</table>

<!-- Tabela Com Colspan -->
<h2>Tabela Simples</h2>
<table width="100%" border="1" >
  <tr>
    <td>Aluno</td>
    <td colspan="2">Disciplina</td>
  </tr>
  <tr>
    <td>Alex</td>
    <td>HTML</td>
    <td>Ambiente Web</td>
  </tr>
</table>

<!-- Tabela Com Colspan -->
<h2>Tabela Simples</h2>
<table width="100%" border="1" >
  <tr>
    <td>Aluno</td>
    <td rowspan="2">Disciplina</td>
  </tr>
  <tr>
    <td>Alex</td>
  </tr>
  <tr>
    <td>HTML</td>
    <td>Ambiente Web</td>
  </tr>
</table>
```

Faça você mesmo!

Questão 1

Uma tabela HTML é composta pela combinação de linhas e colunas. Observe o código abaixo e assinale a alternativa correta.

```
<table>
<tr><td>Col 1</td><td>Col 4</td></tr>
```

```
<tr><td>Col 2</td><td>Col 5</td></tr>
<tr><td>Col 3</td><td>Col 6</td></tr>
</table>
```

A A tabela possui 3 linhas com 2 colunas cada.

B A tabela possui 2 linhas com 3 colunas cada.

C A tabela possui 3 linhas com 3 colunas cada.

D A tabela possui 3 linhas com 5 colunas cada.

E A tabela possui apenas linhas, sem colunas.



A alternativa A está correta.

Na tabela demonstrada no enunciado, há três linhas com duas colunas cada.

Formulários HTML

Tudo o que vimos até este ponto nos permitiu criar um documento HTML estruturado contendo textos, listas, tabelas, vídeo e áudio. Neste módulo, abordaremos o formulário. Esse elemento é um dos mais utilizados para prover interação entre usuários e uma página web ou até mesmo um aplicativo mobile.



Exemplo

Usamos formulários quando visitamos um site e desejamos solicitar uma informação adicional ou um serviço, nos cadastramos para receber e-mails com ofertas ou realizamos uma reclamação.

Ao longo deste módulo, veremos os elementos básicos necessários para a construção de formulários HTML.

Assista ao vídeo e entenda os principais componentes de um formulário e as tags associadas, com exemplos da sua aplicação.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Estrutura do formulário

A exemplo do que vimos com as tabelas, o formulário é composto por uma tag principal, um container, e várias tags “filhas”. Fazem parte de sua estrutura tags como as de campos de texto, de uma ou mais linhas, campos de seleção e botões.

Além disso, para maior clareza, também usamos tags para informar a função dos campos do formulário. São as chamadas label. Veja agora uma lista as tags comumente usadas em um formulário. Vamos lá?

1

`<form>`

Container principal do formulário.

2

`<input>`

Campo do formulário. Como há diversos tipos de campos, fazemos uso do atributo “type” para informar o tipo a ser utilizado – conforme veremos mais adiante.

3

`<textarea>`

Campo de texto de múltiplas linhas.

4

`<select>` e `<option>`

Campos de seleção, onde o container é definido pela tag `<select>` e os itens pela tag `<option>`.

5

`“button”`

Campo de botão. Permite que uma ação seja executada no formulário – enviar o formulário; limpar os dados etc.

6

`“label”`

Campo usado para definir um título, uma legenda, que descreva para que serve cada campo do formulário.

Conhecendo melhor os elementos e atributos do formulário

Na próxima imagem, temos o fragmento HTML correspondente a um formulário. É possível observar as tags já mencionadas, assim como algumas tipificações nos campos e novos atributos, que serão abordadas a seguir:

```

<form action="/pagina-processa-dados-do-form" method="post">
... <fieldset>
...   <legend>Dados Gerais</legend>
...   <label for="nome">Nome:</label>
...   <input type="text" minlength="3" id="nome" />
...
...   <label for="data_nascimento">Data de Nascimento:</label>
...   <input type="text" id="data_nascimento" />
...
...   <label for="cpf">CPF:</label>
...   <input type="cpf" minlength="11" id="cpf" />
... </fieldset>
... <fieldset>
...   <legend>Endereço</legend>
...
...   <label for="tipo_endereco">Tipo:</label>
...   <select id="tipo_endereco">
...     <option value="">Selecione</option>
...     <option value="">Rua</option>
...     <option value="">Estrada</option>
...     <option value="">Rodovia</option>
...     <option value="">Outros</option>
...   </select>
...
...   <label for="logradouro_endereco">Logradouro:</label>
...   <input type="text" id="logradouro_endereco" />
...
...   <label for="numero_endereco">Número:</label>
...   <input type="text" id="numero_endereco" />
...
...   <label for="complemento_endereco">Complemento:</label>
...   <input type="text" id="complemento_endereco" />
... </fieldset>
... <fieldset>
...   <legend>Fale Conosco</legend>
...
...   <label for="msg">Mensagem:</label>
...   <textarea type="text" id="msg"></textarea>
... </fieldset>
... <fieldset>
...   <button type="submit">Enviar sua mensagem</button>
...   <button type="reset">Limpar o Formulário</button>
... </fieldset>
</form>

```

É fácil identificar para o que serve cada campo do formulário anterior ao lermos o conteúdo da tag `<label>`. Além disso, a tag `<fieldset>` cria seções dentro do formulário, ajudando a separar os campos no código e a visualizar a página no navegador.

Isso fica ainda mais claro quando vinculamos ao `<fieldset>` a tag `<legend>`.

Como vimos, é necessária uma atenção especial aos atributos quando tratamos de formulários. Para melhor visualização e entendimento, todos os atributos contidos no código mencionado são descritos a seguir:

`<form>`

action

Define a URL (endereço) para a qual os dados do formulário serão enviados.

`<form>`

method

Define o método HTTP (POST ou GET) para envio dos dados.

`<label>`

for

Vincula a tag `<label>` ao campo ao qual ela se refere. Esse vínculo faz com que seja possível clicar na label para ativar o campo relacionado.

`<input>`

minlength, maxlength

Define a quantidade de caracteres mínima e máxima, respectivamente, permitida em um campo.

`<input>` e `<button>`

type

Define o tipo do campo e, sobretudo, como ele se comporta.

`<option>`

value

Define o valor do campo, e também pode ser utilizado na tag `<input>`. No caso da `<option>`, os seus valores possíveis são previamente definidos quando a página é codificada. Já na `<input>`, embora também possa ser previamente definido, normalmente é o usuário quem define o seu valor.

Atributos do formulário: o atributo type

Este atributo, dada a sua importância, precisa ser visto de forma aprofundada. Como já dito, além de definir o tipo do campo, também determina como este se comporta. No código apresentado na tabela, é utilizado apenas o tipo "text", que, no caso da tag "input", corresponde ao seu valor padrão.

Ao longo de vários anos, havia apenas esses tipos disponíveis na HTML. Com isso, algumas necessidades – fossem de inserção de tipos de dados específicos fossem de validação de valores – não podiam ser supridas apenas com a utilização de tags, sendo necessário combinar códigos Javascript e CSS. Por exemplo: um campo para seleção de data.

Além disso, novos tipos de dados, com características específicas, ganharam importância ao longo dos anos. Podemos citar, como exemplo, o e-mail. Embora seja um campo de texto, ele possui um padrão de composição próprio, como o uso de @, entre outras características.

Novos atributos e tipos

Pensando nas deficiências citadas, a HTML5 definiu novos tipos de entrada e também novos atributos relacionados a formulários. Destacamos alguns desses novos atributos e tipos, veja!

Atributos

- **placeholder** Função: Exibir um texto no campo de input. Comentário: Utilizado para dar uma dica ao usuário sobre o dado a ser inserido.
- **required** Função: Determinar se um campo é obrigatório. Comentário: Utilizado na validação dos dados de um formulário.
- **autofocus** Função: Fixar o foco no campo. Comentário: Utilizado quando desejamos que, ao carregar o formulário, um determinado campo seja focado.
- **pattern** Função: Validar o valor de um campo com base em uma expressão regular. (as RegEX – expressões regulares – podem ser consideradas um recurso de linguagem de programação para a análise e manipulação de texto) Comentário: O campo de tel. é um bom exemplo de utilização desse atributo. Com ele podemos, por exemplo, determinar a quantidade de caracteres e o formato esperado para um campo.

Tipos

- **tel** Função: Inserir números de telefone. Comentário: Para uma melhor usabilidade, deve ser usado em conjunto com o atributo "pattern".
- **datetime** Função: Inserir datas com o fuso horário em UTC a partir de um calendário.
- **date** Função: Inserir datas sem fuso horário a partir de um calendário.
- **number** Função: Inserir números. Comentário: Cria um componente diferente do input normal, em que, por meio de setas, os números podem ser incrementados ou decrementados.

Atividade 1

Questão 1

Sobre a composição e o comportamento do formulário HTML, é correto afirmar:

A Apenas os formulários validados podem ser enviados/submetidos.

B Para submeter um formulário, é necessária a utilização de uma linguagem que permita a interação em páginas web, como a Javascript.

C Formulários que não possuam uma tag “label” para identificar cada um dos seus campos retornam um erro ao serem submetidos.

D O formulário HTML, por ser um elemento de interação, na maioria das vezes envia os dados nele contidos para um servidor web, responsável por receber e tratar os dados.

E Por não possuir um atributo que defina o endereço/URL para onde seus dados serão enviados, todo formulário HTML é processado na mesma página HTML em que está inserido.



A alternativa D está correta.

Os formulários são compostos por tags que permitem a entrada e seleção de dados. Essa funcionalidade concede a este elemento a característica de interatividade, permitindo assim que, a partir de ações do usuário, dados sejam enviados, informações sejam cadastradas ou comportamentos e conteúdo da própria página sejam modificados – como, por exemplo, acontece quando filtramos o resultado de uma pesquisa de produtos em um site de e-commerce. Logo, podemos dizer que há dois comportamentos comuns na submissão de um formulário: o de enviar seus dados para serem processados remotamente e o de alterar as informações contidas na página (como no caso do filtro já mencionado).

Validação de dados em formulários

Uma boa prática, quando utilizamos formulários HTML, é validarmos os dados submetidos. Isso nos garante que serão cumpridos alguns requisitos preestabelecidos, como a quantidade de caracteres ou o formato do dado, por exemplo. O processo de validação, assim como toda a HTML, tem mudado ao longo dos anos. Se, anteriormente, tal processo era restrito ao uso de outra linguagem, o Javascript, hoje já é possível, na HTML5, usarmos alguns de seus atributos para tal finalidade.

Assista ao vídeo e entenda como funciona a validação aplicando a técnica para validar um código de rastreamento do correio brasileiro, que é composto por duas letras, seguidas por nove algarismos e letras BR.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Para explicarmos a criação de um formulário HTML, apresentamos tags e atributos que nos permitem montar um formulário já funcional, pronto para ser preenchido e submetido. Entretanto, há outro aspecto relacionado a esses elementos que precisa ser discutido: a validação de dados.

A validade dos dados concorre com a utilização das tags corretas que permitam a **melhor experiência possível aos usuários**. Podemos dizer que, ao pensarmos na estrutura do formulário, nas tags e nos atributos, estamos pensando em quem vai preencher o formulário.



Exemplo

Imagine que podemos permitir a inserção de uma data de nascimento com a digitação de valores ou a partir da seleção em um elemento do tipo calendário. Agora pense em quem vai receber as informações provenientes do formulário. Em um campo de texto simples, sem nenhum tipo de padrão, serão recebidas as mais variadas combinações de dados. Por exemplo: 01 de janeiro de 1980; 01/01/1980; 01011980 etc. Imaginando esse cenário, é possível entender a importância da validação das informações.

Como funciona a validação?

A validação é um processo que pode e deve ocorrer tanto no **lado cliente** – no navegador – quanto no **lado servidor**. Pensando no lado cliente, essa ação ocorre assim que o formulário é submetido, antes que os dados sejam recebidos pelo servidor de destino.

Há pouco tempo, para validar um formulário, era necessário usar o Javascript. Entretanto, na HTML5 é possível fazê-lo de forma nativa, sem o uso de linguagens de programação.

Tipos de validação

Na HTML5, há dois tipos possíveis de validação:

Tipagem

Um tipo verifica se o dado inserido em um campo é consistente com o seu tipo e/ou padrão (pattern).

Obrigatoriedade

Outro tipo verifica se um campo obrigatório foi preenchido.

Vejamos o exemplo!



Exemplo

Quanto à primeira validação, podemos citar novamente o elemento input do tipo “e-mail”. Um endereço de e-mail é um dado que possui uma formatação predefinida: precisa ter o “@”; precisa ter a extensão de domínio “.com / .com.br” etc. Logo, declarar uma tag input com o type “e-mail” faz com que, naturalmente, seja validado o seu conteúdo.

Algo semelhante acontece com a utilização do atributo “pattern”, sendo que, nesse caso, é você quem define o que um campo precisa conter para ser considerado válido. Por exemplo: determinar o formato desejado para um campo que receba um número de telefone.

Você pode definir que ele contenha o DDD, com dois caracteres numéricos, seguido por dois conjuntos de números – um contendo 5 e outro contendo 3 ou 4 caracteres – o que geraria este código: pattern=“

[0-9]{5}-[0-9]{3,4}\$”

Além da validação pelo tipo de dado, há também a validação de campos obrigatórios. Logo, quando precisamos que determinado campo não fique em branco, usamos o atributo “required”.

O comportamento padrão da HTML5 é validar os dados quando ocorre o envio do formulário. Entretanto, é possível desativar essa funcionalidade. Para isso, deve ser atribuído o atributo “novalidate” à tag <form>.

Atividade 2

Questão 1

Sobre a validação de formulários na HTML5, assinale a afirmativa verdadeira.

- ☐ A A HTML5 é tão flexível que permite ao usuário, o visitante de uma página, decidir quando ele deseja ou não que o formulário que está preenchendo seja validado.
- ☐ B A partir da HTML5, para facilitar o trabalho do desenvolvedor, novas tags foram criadas para que a validação aconteça apenas no lado servidor.
- ☒ C A validação de formulários na HTML5 é habilitada por padrão. Entretanto, é possível desabilitar este recurso com a utilização de um atributo na tag <form>.
- ☐ D Como a validação de formulários na HTML5 é habilitada por padrão, automaticamente todos os campos do formulário serão marcados como obrigatórios e, portanto, deverão ser preenchidos.
- ☐ E A validação de um formulário não tem como ser realizada de maneira nativa no HTML5, devendo ser feita através de alguma linguagem de programação, como a linguagem Javascript.



A alternativa C está correta.

Uma das grandes novidades da HTML5 foi implementar, nativamente, a validação de dados em um formulário, tarefa que, nas versões anteriores, só era possível, no lado cliente, com a utilização de Javascript.

Inserindo um formulário numa página web

O formulário HTML é uma das principais formas de interação em uma página web. Com esse elemento, podemos inserir dados em um cadastro, filtrarmos buscas, entre outras ações. Para realizar tais tarefas, o formulário é composto por uma estrutura de tags e atributos. Vamos agora colocar em prática o que aprendemos até aqui para criarmos um formulário HTML.

Assista ao vídeo e confira a atividade em que se cria a estrutura básica da página HTML e o container do formulário, contendo os campos para inserção de nome, e-mail, telefone de contato, calendário, data de nascimento e o campo para inserção de texto livre.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Roteiro de prática

A criação de um formulário precisa seguir algumas regras, como inserir o elemento container e os demais elementos, e também atributos, de acordo com a sua funcionalidade. Aqui, seguindo os passos abaixo, você deverá criar um formulário de contato.

- ***Utilizando um editor, crie a estrutura básica da página HTML e o container do formulário.***
- ***O formulário deverá conter campos para inserção de: nome, e-mail e telefone de contato.***
- ***O usuário deverá poder selecionar, em um calendário, a sua data de nascimento.***
- ***O formulário deverá conter um campo para inserção de texto livre.***
- ***Indique ao usuário o que ele deverá preencher em cada campo.***
- ***Ao final, salve sua página e a teste no navegador. Lembre-se de incluir também o botão de submissão do formulário.***

Como resolução, teremos o seguinte código:

python

Nome :

Email:

Telefone:

Data de Nascimento:

Mensagem:

Faça você mesmo!

Questão 1

Observe o seguinte fragmento de código:

```
<form>
  <label for="nome">Nome: </label>
  <input type="text" name="nome" required>
  <br>
  <label for="website">Website: </label>
  <input type="url" name="website" required pattern="https?://.+">
  <br>
  <label for="tel">Idade:</label>
  <input type="number" size="6" name="idade" min="18" max="99" value="21">
  <br>
  <input type="submit" value="Enviar">
</form>
```

Repare que estão sendo usados alguns atributos para validação dos dados. A partir de sua análise, assinale a alternativa abaixo que representa as validações realizadas no formulário.

A

Está sendo validado se todos os campos foram preenchidos, exceto o botão de enviar.

B

Está sendo validado se o campo nome é do tipo texto; se o campo website é do tipo URL e se o campo idade é do tipo número. Além disso, está sendo validado se todos os campos foram preenchidos.

C

Além de validar se os campos nome e website foram preenchidos e garantir que o campo idade tenha um valor padrão, está sendo validado o formato inicial do valor inserido no campo website (a partir de um padrão predefinido) e se a idade é maior ou menor que valores predeterminados, além de ser do tipo número.

D

Está sendo validado o preenchimento dos campos nome e website foram preenchidos. Além disso, a definição do tipo URL no campo website está garantindo que sejam inseridas apenas URLs válidas. Por fim, está sendo validado se a idade é maior ou menor que valores predeterminados, além de ser do tipo número.



A alternativa C está correta.

Para validarmos os dados de um formulário, é necessário fazermos uso de alguns atributos, como o "required" ou o "pattern". Enquanto o primeiro garante que o campo seja preenchido, o segundo determina um padrão para o valor a ser inserido. Além disso, podemos fazer uso de outros atributos como min ou max para garantir que um campo tenha valores dentro de uma faixa predeterminada.

Considerações finais

O que você aprendeu neste conteúdo?

- **A história por trás da HTML e como estruturar uma página web.**
- **Como utilizar as tags HTML e as tags para fins específicos, como listas e tabelas.**
- **Como utilizar um formulário HTML, validando seus dados.**

Agora você está pronto para aplicar seus conhecimentos de HTML, e pode construir páginas e desenvolver aplicações web.

Explore +

Confira as indicações que separamos especialmente para você!

- **Leia o artigo Estrutura de documentos e sites no site MDN.**
- **Para conhecer as referências das tags HTML, leia o artigo HTML Element Reference no site da W3Schools.**
- **Faça o Test your skills: Form validation, encontrado no site do MDN, para avaliar os seus conhecimentos sobre a validação de formulários.**

Referências

FREEMAN, E.; ROBSON, E. **Use a Cabeça!** Programação em HTML5: desenvolvendo aplicativos para web com Javascript. Rio de Janeiro: Alta Books, 2014.

HTML Tutorial. W3Schools. Consultado na internet em: 05 jun. 2023.

NEWCOMB, S.; KIPP, N.; NEWCOMB, V. **The "HyTime" Hypermedia/Time-Based Document Structuring Language.** Communications of the ACM, n. 11, v. 34, p. 67-83, 1991.