



Linguagem de marcação de hipertexto - HTML

Prof. Alexandre de Oliveira Paixão

Prof. Kleber de Aguiar

Descrição

Apresentação dos conceitos de HTML, linguagem de marcação e estrutura básica de páginas Web.

Propósito

Explicar a função e utilização da linguagem de marcação HTML.

Preparação

Leia sobre o conceito básico de HTML e instale no computador um navegador web/browser e um editor de texto que dê suporte à marcação HTML. No Sistema Operacional Windows é indicado o Notepad++. No Linux, o Nano Editor.

Objetivos

Módulo 1

Estrutura de uma página Web

Descrever a estrutura de uma página Web.

Módulo 2

Tags HTML básicas

Descrever o uso das tags na HTML.

Módulo 3

Tags HTML complexas

Demonstrar as tags complementares às tags básicas em uma página Web.

Módulo 4

Formulários em páginas Web

Aplicar um formulário em uma página Web.



Introdução

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

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.

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



Hipertexto

Texto ao qual podem ser agregados outros conjuntos de informação, como textos, multimídias etc., e cujo acesso é realizado através de referências denominadas hiperligações.



1 - Estrutura de uma página Web

Ao final deste módulo, você será capaz de descrever a estrutura de uma página Web.

Visão geral

O princípio da HTML

A especificação da HTML foi criada a partir da junção de dois padrões: o **SGML** e o HyTime. O primeiro, SGML (Standard Generalized Markup Language), é, na verdade, 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 hipermídia pudessem ser descritos em função de sua estrutura e conteúdo e, consequentemente, utilizados por quaisquer sistemas conformantes (NEWCOMB *et al.*, 1991).

Inicialmente, a especificação da HTML e 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 também o status de norma internacional (ISO/IEC) em 2000.

HyTime

Hypermedia/Time-Based Structuring Language.

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.

A evolução da HTML: versões

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

Veja na tabela a seguir o histórico de **versões da HTML**:

Ano	Versão	Histórico
1991	HTML	Tim Berners-Lee cria a HTML

Ano	Versão	Histórico
1995	HTML 2	Grupo de Trabalho HTML define a HTML 2.0
1997	HTML 3.2	W3C publica a Recomendação HTML 3.2
1999	HTML 4.01	W3C publica a Recomendação HTML 4.01
2000	XHTML	W3C publica a Recomendação XHTML 1.0
2014	HTML5	W3C publica a Recomendação HTML5
2016 2017	HTML 5.1	W3C candidata a Recomendação HTML 5.1 W3C publica a Recomendação HTML 5.1 2ª edição
2017	HTML 5.2	W3C publica a Recomendação HTML5.2

Tabela: Histórico de versões da HTML.
Alexandre de Oliveira Paixão.

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 etc.

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 onde 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**.

Tal necessidade se tornou contínua, tendo como principais pilares estes eventos já mencionados: 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. Por isso, o W3C criou o DTD – Definição de Tipos de Documentos (Document Type Definition), que é 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.

Antes da HTML5, existia um conceito chamado **concordância de documento**. Embora relevante apenas nas versões anteriores da HTML, ainda é importante abordá-lo, uma vez que muitas páginas Web ainda não foram portadas para a HTML5. Tal conceito significa que cada página precisa ser concordante com a especificação W3C oficial para a qual ela foi escrita.



Exemplos de tags na HTML.

Renderizadas

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

A especificação da HTML 4, por exemplo, previa três tipos de Doctypes: **Strict**, **Transitional** e **Frameset**. Vamos conhecer cada uma delas?

Strict



Determinava 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

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

Frameset

Tag obsoleta da HTML4 usada para combinar múltiplos frames e exibi-los como numa única página.

Vemos o fragmento do código inicial de uma página HTML 4.01:

```
<!DOCTYPE HTML PUBLIC "-//W3C/DTD HTML 4.01 Transitional//EM" http://www.w3.org/TR/html4/loose.dtd>
```

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: 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:

```
<!DOCTYPE html>
```

Comparando as declarações do Doctype entre a HTML5 e a HTML4, notamos algumas diferenças.

A principal é que a HTML5, ao contrário das versões anteriores, **não se baseia na SGML** e, portanto, **não requer referência a um DTD**.



Utilizando Doctypes – Document Types

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



Estrutura de uma página Web

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, em conjunto com o Doctype, formam a sua estrutura básica. Tal estrutura é demonstrada a seguir:

HTML



```
1 <!doctype html>
2 <html>
3   <head>
4   </head>
5   <body>
6   </body>
7 </html>
```

Estrutura básica de uma página Web.

Vamos entender o que significa cada tag?

Raiz

A tag inicial, após a declaração do Doctype, é a `<html>`. Tal elemento é a **raiz do documento**. Logo, e 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 `<html>` é a abertura, e o seu fechamento se dá com a declaração `</html>`. Com isso, observando a imagem anterior, é possível perceber que as demais tags estão contidas dentro da tag raiz, `<html>`.

Comentário

Antecipando um conteúdo que será visto mais adiante, cabe destacar outro elemento importante referente à tag raiz: o atributo `lang`. Este, propositalmente, não foi inserido na Figura Estrutura básica de uma página Web, uma vez que ainda não tratamos dos conceitos de atributo. Por ora, cabe dizer que este 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. Alguns exemplos de sua declaração:

```
<html lang="en-US" >
```

```
<html lang="en-GB" >
```

```
<html lang="pt" >
```

Cabeçalho

A tag **`<head>`** é chamada de **cabeçalho do documento**. Fazendo 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.

Na tabela a seguir visualizamos as tags que fazem parte do cabeçalho:

Tag	Para que serve
<code><title></code>	É o título do documento, sendo visualizada na barra de título do navegador.
<code><meta></code>	Engloba uma série de informações – comumente chamadas de metainformações – como a descrição da página, palavras-chave etc.
<code><script></code>	É responsável pela inclusão e/ou definição de scripts relacionados ao documento.
<code><link></code>	É responsável pela inclusão de folhas de estilo (externas) relacionadas ao documento. Também possibilita a inclusão de favicons (pequenos ícones que

Tag	Para que serve
	aparecem na barra de endereços do navegador).
<style>	Assim como a anterior, é responsável pelo vínculo de folhas de estilo ao documento – quando elas são declaradas diretamente no documento.

Tabela: Lista de tags contidas no cabeçalho da página Web.

Corpo

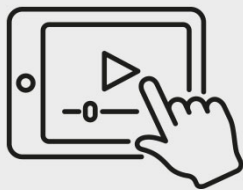
Logo abaixo do cabeçalho deve ser inserida a tag **<body>**. Esta é responsável pela **estruturação do documento**, sobretudo de seu conteúdo e também apresentação – embora seja fortemente recomendado que a apresentação do documento, como será visto no módulo específico, seja feita por meio de folhas de estilo (CSS).

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



Estrutura de uma Página Web – Elementos obrigatórios

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



Elementos básicos

Embora as páginas Web sejam diferentes umas das outras, elas tendem a compartilhar uma mesma estrutura. Isto é, além da estrutura apresentada acima, uma **página Web** é, na maioria das vezes, **composta por seções**, sendo as mais comuns:

Cabeçalho

Uma área inicial, logo na parte superior da página. Normalmente vemos um título, uma logomarca ou algo do tipo.

Barra de navegação

Esta área pode ser tanto horizontal quanto vertical – e, em alguns casos, os dois. Contém os links para navegação pelas seções/páginas do site.

Conteúdo

O conteúdo principal da página, que pode ser dividido em subseções.

Barra Lateral

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

Rodapé

A área final da página, onde normalmente são dispostos links, avisos de direitos autorais, informações complementares sobre o proprietário do site etc.



Estrutura de uma Página Web – Elementos Básicos

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



Funcionamento da HTML

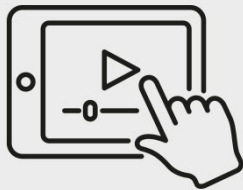
Após concluído, o documento estruturado, composto por tags, deve ser salvo com a extensão “html” ou “htm”. Assim, ele estará pronto para ser lido por um dispositivo, como um navegador Web – em um computador pessoal ou em um smartphone. Ou seja, quando uma página Web é exibida no navegador, o que vemos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



Vamos praticar?

Agora que já vimos um pouco de teoria sobre a estrutura de uma página Web, chegou a hora de praticar. Vamos lá!

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



Falta pouco para atingir seus objetivos.

Vamos praticar alguns conceitos?

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 páginas 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.

Parabéns! 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.

Questão 2

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>

Parabéns! A alternativa E está correta.

Conforme visto na figura **Histórico de versões da HTML**, a estrutura considerada obrigatória, para a correta renderização de uma página Web nos navegadores, é composta pelas tags e .



2 - Tags HTML básicas

Ao final deste módulo, você será capaz de descrever o uso das Tags na HTML.

Visão geral

Aprofundando o conceito de tags

Aprendemos que a HTML é uma linguagem de marcação que faz uso de tags para estruturar o seu conteúdo. Mas o que são tags?

Como visto nos exemplos do módulo anterior, 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, consistia em inserir na estrutura básica de tags da HTML – mais precisamente dentro da tag <body> –, um texto qualquer e, a seguir, salvar e carregar esta 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 amontado dentro da tag <body>.

Embora exibido pelo navegador, tudo seria uma grande bagunça, uma confusão. No lugar disso, 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.

Declaração de tags

Como declarar tags

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

O primeiro ponto a se ter em mente é 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 a forma como uma tag é escrita, entre os sinais de menor e maior. Além disso, ela deve, obrigatoriamente, ser fechada. Para isso, repetimos a declaração da tag e utilizamos uma barra "/" antes do seu nome.

Exemplos

```
<h1>Título de uma seção</h1>
```

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

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

Existem algumas tags que fogem à essa regra.

Um exemplo bem conhecido é a 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 em tipos, de acordo com as suas funções: **Estruturais, textuais e semânticas**. Outra característica importante é que elas também podem ter atributos. A seguir, falaremos sobre cada um desses temas.

Os atributos

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

Exemplo

```

```

Esta 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 eles e as tags. Ao longo dos próximos módulos, veremos alguns dos principais, lembrando que o site do W3C contém a lista completa de atributos e combinações. Por ora, cabe ainda 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



Usado 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, através de Folhas de Estilo ou eventos e interação, através de Javascript.

Tipos de tags: **textuais e semânticas**

Até aqui, conhecemos algumas tags associadas à estrutura, dita obrigatória, de uma página. Também vimos que, na maioria dos casos, as páginas Web possuem uma mesma estrutura em termos de conteúdo. A seguir, conheceremos os tipos de tag textuais e semânticos.

Tags textuais

São responsáveis por **organizar o conteúdo da página**, ou seja, textos, mídias e links, por exemplo. Algumas das principais tags textuais, inclusive vistas anteriormente, 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 assim:

```
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 figura, é possível perceber que existem tags específicas para cada seção do conteúdo. Essa é uma característica importante da HTML, chamada de semântica.

Logo, **semântica**, neste contexto, pode ser considerada a correta utilização de uma tag HTML de acordo com o seu conteúdo ou finalidade.

Em muitos casos é indiferente, ao visualizarmos uma página no navegador, as tags utilizadas para guardar determinado conteúdo. Entretanto, é fortemente recomendado usá-las de acordo com sua função.

Isso 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.

Veja outras tags e suas respectivas funções:

Tag	Descrição/Função
<code><article></code>	Inclui um bloco de conteúdo que deve ser usado quando se deseja inserir um artigo, como um post de um blog, por exemplo.
<code><section></code>	Define uma seção no documento. É normalmente utilizado para agrupar seções. Por exemplo: uma <code><section></code> poderia conter vários <code><article></code> .
<code><h1></code> <code><h2></code> <code><h3></code> <code><h4></code> <code><h5></code> <code><h6></code>	Usada para inserir 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 <code><h1></code> , no subtítulo seguinte, <code><h2></code> , e assim por diante. Além disso, uma página poderá conter várias <code><h1></code> , <code><h2></code> ... <code><h6></code> . A exceção aqui fica por conta da tag <code><h1></code> , uma vez que é recomendado utilizar apenas uma numa mesma página.
<code><p></code>	Usada para inserir parágrafos de texto.
<code><pre></code>	Usada para inserir um texto pré-formatado.
<code><div></code>	Embora não seja considerada semântica, essa tag poderá ser usada para agrupar algum tipo de conteúdo que não tenha nenhuma semântica específica ou que não se encaixe bem dentro de uma tag semântica.
<code></code>	É semelhante à <code><div></code> . Entretanto, enquanto a <code><div></code> é um elemento não semântico no bloco (quando usada, quebra o conteúdo em uma seção), a <code></code> é embutida (não quebra o conteúdo, apenas o agrupa).
<code><a></code>	Usada para inserir links.
<code>
</code>	Usada para inserir uma quebra de linha.

Tag	Descrição/Função
<code><hr></code>	Insere uma linha horizontal no documento. Normalmente é utilizada quando se pretende alterar a temática de um texto.

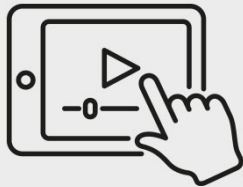
Tabela: Lista de tags estruturais, textuais e semânticas.

Alexandre de Oliveira Paixão.



Tipos de tags: textuais e semânticas

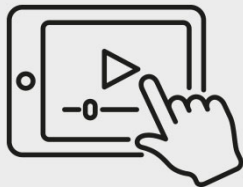
Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



Vamos praticar?

Veja a evolução do que vimos no vídeo anterior.

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



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

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

- ``: aplica o efeito de negrito em um texto;
- ``: aplica o efeito de negrito em um texto e o marca como importante;
- `<i>`: aplica o efeito de itálico em um texto;
- ``: 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 as duas produzem o mesmo efeito visual (que `` e `<i>` respectivamente) em um texto, ou seja, **marcá-lo como negrito e/ou itálico**. Entretanto, há uma diferença importante entre elas, que vai além da visualização do texto no navegador pela maioria dos usuários. A função semântica de ambas 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**. Estas, chamadas de obsoletas, embora ainda possam ter suporte em boa parte dos navegadores, devem ser evitadas pelo seguinte motivo:

- Em primeiro lugar, porque provavelmente foram substituídas por novas tags, com melhor semântica.
- Em segundo lugar, pelo risco de desconfigurarem o conteúdo da página, uma vez que os browsers podem deixar de suportá-las a qualquer momento.

Tag	Função
<code><applet></code>	Identifica a inclusão de um applet Java.
<code><center></code>	Centraliza horizontalmente o conteúdo de um bloco.
<code><dir></code>	Container para lista de diretórios ou arquivos.
<code></code>	Determina características relacionadas a fontes de um determinado elemento.

Tag	Função
<code><image></code>	Ancestral da tag <code></code> , era usada nas primeiras versões HTML para a inserção de imagens.

Tabela: Exemplos de tags obsoletas.

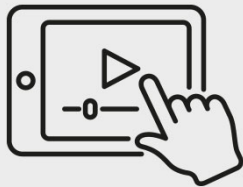
Alexandre de Oliveira Paixão.

Atualmente consideradas obsoletas, as tags `<center>` 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.



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

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



Falta pouco para atingir seus objetivos.

Vamos praticar alguns conceitos?

Questão 1

Marque a alternativa incorreta e que não pode ser usada para definir o que são as tags:

A

São nomes, rótulos, que servem para informar ao browser como uma página deve ser renderizada

B As tags são marcações que, embora possuam a mesma forma de serem declaradas, contêm instruções específicas de acordo com o seu nome.

C São elementos que têm a responsabilidade de organizar e identificar o conteúdo de uma página Web.

D As tags são textos simples inseridos entre o sinal de < e >, e podem tanto ser usadas a partir das especificações W3C ou serem criadas/inventadas por cada pessoa ao construir um documento HTML.

E Servem para informar ao browser a versão do HTML referente à página que está sendo exibida.

Parabéns! A alternativa D está correta.

As tags são elementos que servem para marcar quais informações uma página exibe. Por exemplo: a marcação <h1>Título de um bloco de texto</h1> insere um título em uma página HTML. Estes elementos são definidos, padronizados e mantidos em especificações suportadas pela W3C.

Questão 2

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

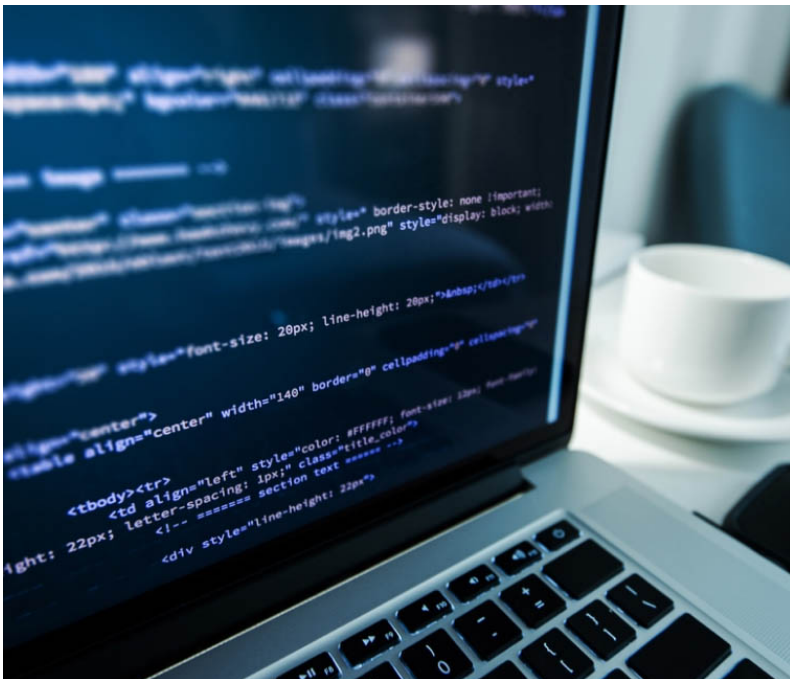
A O conceito de semântica pode ser resumido como a escolha e utilização de tags de acordo com o tipo de conteúdo ou função.

B Na prática, o que importa mesmo não é a amarração entre uma tag e seu conteúdo, mas 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 tem por característica organizar textos, mídias e links da página HTML.

Parabéns! 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.



3 - Tags HTML complexas

Ao final deste módulo, você será capaz de demonstrar as tags complementares às tags básicas em uma página Web.

Listas na HTML

Neste módulo, daremos continuidade à estruturação de conteúdo com a utilização de tags. Novas tags para a representação de novos tipos de conteúdo serão apresentadas: as listas, as tabelas, os vídeos e os áudios.

Listas

O HTML fornece suporte para a representação visual de três tipos de listas: as **ordenadas**, as não **ordenadas** e as de **definição**, veja:

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 é 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 a seguir cada tipo de lista:

Tipo	Tag Container	Tag Item	Tag Descrição
Ordenadas			--
Não Ordenadas			--
Definição	<dl>	<dt>	<dd>

Tabela: Tags utilizadas para a representação de listas.

Veja a seguir exemplos de como as tags são utilizadas e seu resultado na tela:

Lista ordenada

```
<ol>
<li><ol> Primeiro Item da Lista </ol></li>
<li>Segundo Item da Lista </li>
<li>Terceiro Item da Lista </li>
</ol>
```

Resultado:

1. Primeiro Item da lista
2. Segundo Item da lista
3. Terceiro Item da lista

Lista não ordenada

```
<ul>
<li><ol> Primeiro Item da Lista não ordenada </ol></li>
<li>Segundo Item da Lista não ordenada </li>
<li>Terceiro Item da Lista não ordenada </li>
</ul>
```

Resultado:

- Primeiro Item da lista não ordenada
- Segundo Item da lista não ordenada
- Terceiro Item da lista não ordenada

Lista de definição

```
<dl>
<dt><ol> Primeiro Item da Lista não ordenada </ol></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>
```

Resultado:

Primeiro Item da lista não ordenada

Descrição do primeiro elemento

Segundo Item da lista não ordenada

Descrição do segundo elemento

Terceiro Item da lista não ordenada

Descrição do terceiro elemento

Tags e representação visual das listas no navegador.

HTML

```
1 <ol>
2   <li> Primeiro Item da lista </li>
3   <li> Segundo Item da lista </li>
4   <li> Terceiro Item da lista </li>
5 </ol>
```

Cabe destacar que tanto o símbolo de ordenação (a numeração, no exemplo acima), no caso da primeira, quanto o símbolo visual (o bullet – pequeno círculo preto), no caso da segunda, podem ser alterados com a utilização de CSS.

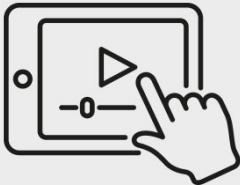
Atenção!

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



Listas ordenada, não ordenada e de definição

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



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, além da tag principal <table>, outras tags. A principal define o container geral.

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

A tabela abaixo apresenta as tags de marcação, e suas respectivas funções, relacionadas às tabelas:

Tag	Função
<table>	Container principal da tabela
<tr>	Representa as linhas, sendo composta pelas tags relacionadas às colunas.

Tag	Função
<td>	Representa as colunas e precisa ser inserida dentro da tag de linha.
<th>	Também 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. Este elemento apresenta estilos próprios, sendo renderizado de forma diferente das colunas comuns.
<thead>	Armazena o cabeçalho da tabela, sendo composto por linhas e colunas. Este elemento, a exemplo do que vimos anteriormente, tem função semântica em termos de estruturação de conteúdo.
<tfoot>	Armazena o rodapé da tabela, tendo também função semântica em termos de estruturação.

Tabela: O elemento Tabela e suas tags.
Alexandre de Oliveira Paixão.

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: **rowspan** e **colspan**. Como o próprio nome indica, estes 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.

Comentário

Ao final deste módulo teremos a oportunidade de praticarmos um pouco mais e veremos como esses atributos funcionam.

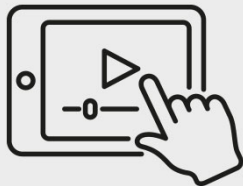
As tabelas e a semântica

Podemos dizer que as tabelas foram, desde a criação do código HTML, o elemento mais utilizado fora de sua função semântica. Isto 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.



Estrutura das Tabelas

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



Aplicação

Como vimos, um documento HTML pode conter tanto tabelas simples quanto mais complexas. Tendo como base o arquivo HTML usado nos módulos anteriores e o conteúdo já visto, chegou a hora de **codificarmos**. No editor de texto, comece criando uma nova seção no seu HTML e insira:

- uma tabela simples;
- uma tabela que contenha título;
- uma tabela com cabeçalho e rodapé;
- uma tabela com linhas e colunas expandidas;
- um exemplo completo;
- salve o arquivo e veja o resultado no navegador.

Veja a demonstração:

```

<!-- Tabela Simples -->
<h2>Tabela Simples</h2>
<table width="100%" border="1">
  <tr>
    <td>Alex</td>
    <td>HTML</td>
    <td>JS</td>
  </tr>
  <tr>
    <td>Ana</td>
    <td>HTML</td>
    <td>JS</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>JS</td>
  </tr>
  <tr>
    <td>Ana</td>
    <td>HTML</td>
    <td>JS</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</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Janeiro</td>
      <td>40</td>
    </tr>
    <tr>
      <td>Fevereiro</td>
      <td>40</td>
    </tr>
  </tbody>
</table>

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

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

```

Demonstração de aplicação no editor.

Mídias na HTML

Mídias: 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:

HTML



```
1 <video src= "http://www.youtube.com/watch?v=20SHvU@PKsM" controle></video>
2 <audio src= "/audios/sample.ogg" controls autoplay loop></audio>
```

Tags de vídeo e áudio no HTML.

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 onde a página Web fica hospedada.

Os atributos `"controls"`, `"autoplay"` e `"loop"`, que são específicos para este 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"`.

Teoria na prática

Ao final deste módulo, vamos inserir novas tags em nosso arquivo HTML. Inclua ao menos uma tag de vídeo e uma de áudio. Além disso, experimente inserir e remover os controles mencionados, sempre salvando e comparando os resultados no navegador.

HTML

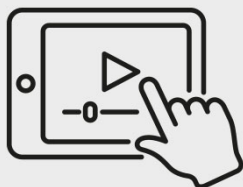


```
1  <!doctype html>
2  <html lang = "pt-BR">
3    <head>
4      <meta charset="utf-8">
5    </head>
6    <body>
7      <video src="small.mp4" controls> < /video>
8      <audio src="sample.ogg" controls autoplay loop> < /audio>
9    </body>
10 </html>
11
```



Vamos praticar?

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



Falta pouco para atingir seus objetivos.

Vamos praticar alguns conceitos?

Questão 1

Qual fragmento de código HTML, entre as opções abaixo, deve ser utilizado para representar a lista apresentada abaixo:

1. Capítulo I

1.1. Primeira Seção

1.2. Segunda Seção

1.3. Terceira Seção

2. Capítulo II

3. Capítulo III

A ``

B ``

C ``

D ``

E ``

Parabéns! A alternativa A está correta.

Os elementos de listas no HTML são bastante flexíveis, além de abrangentes, permitindo a estruturação das mais variadas combinações, incluindo listas aninhadas. Logo, é possível incluir uma lista aninhada – um elemento “filho” de um item de outra lista.

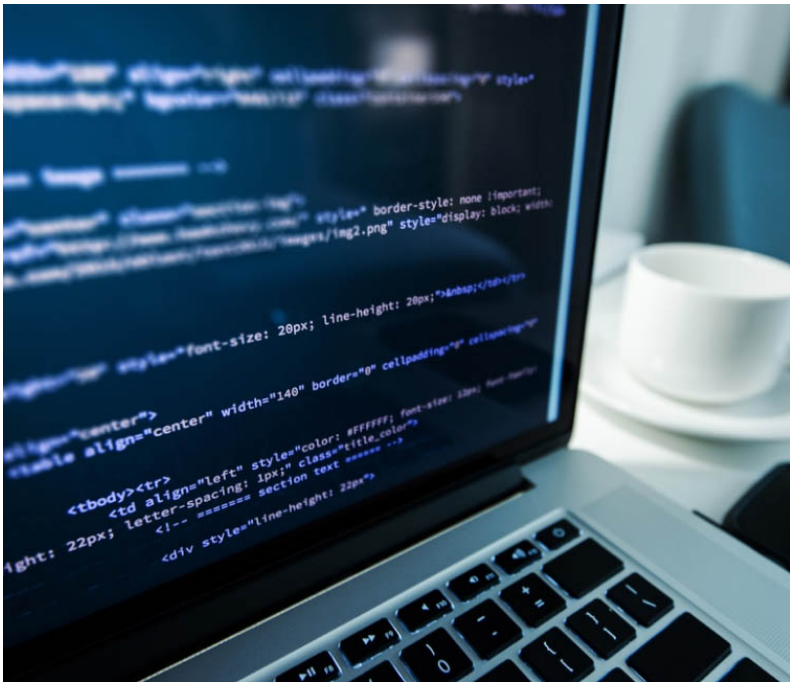
Questão 2

As linhas e colunas de uma tabela HTML podem ser modificadas (mescladas) para permitir uma organização não uniforme. Assinale a opção verdadeira:

- 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 através da deleção de linhas existentes e inserção de novas linhas.

Parabéns! A alternativa B está correta.

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



4 - Formulários em páginas Web

Ao final deste módulo, você será capaz de aplicar um formulário em uma página Web.

Visão geral

Formulários: composição e validação

Tudo o que vimos até esse ponto nos permitiu criar um documento HTML estruturado contendo textos, listas, tabelas, vídeo e áudio. Neste módulo, falaremos sobre o **Formulário**.

Este elemento é um dos mais utilizados para prover interação entre usuários e uma página Web ou até mesmo um aplicativo *mobile*.

Exemplo

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

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

Estrutura básica do formulário

Estrutura inicial do formulário

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

Além disso, para maior clareza, também usamos tags para informar a função dos campos do formulário. São as chamadas “label”. A tabela a seguir lista as tags comumente usadas em um formulário:

Tag	Função
<form>	Container principal do formulário.
<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.
<textarea>	Campo de texto de múltiplas linhas.
<select> e <option>	Campos de seleção, onde o container é definido pela tag <select> e os itens pela tag <option>.
button	Campo de botão. Permite que uma ação seja executada no formulário – enviar o formulário, limpar os dados etc.
label	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**. Nela é possível ver as tags já mencionadas, assim como algumas tipificações nos campos e novos atributos. Falaremos sobre ambos a seguir:

```
<form action="/pagina-processa-dados-do-form" method="post">
  <fieldset>
    <legend>Dados pessoais</legend>
    <label for="nome">Nome</label>
    <input type="text" value="" id="nome" />
    <label for="data_nascimento">Data de Nascimento</label>
    <input type="text" id="data_nascimento" />
    <label for="cpf">CPF</label>
    <input type="text" value="" 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="">Avenida</option>
      <option value="">Bulevar</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>Para Contato</legend>
    <label for="email">Email</label>
    <input type="text" id="email" />
    <label for="telefone">Telefone</label>
    <input type="text" id="telefone" />
  </fieldset>
  <input type="button" value="Enviar" />
  <input type="button" value="Cancelar" />
</form>
```

Formulário HTML – Elementos e atributos.

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 nesta tabela:

Tag	Atributo	Função do atributo
<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.

Tag	Atributo	Função do atributo
<label>	for	Vincula a tag <label> ao campo ao qual ela se refere. Este vínculo faz com que seja possível clicar na label para ativar o campo relacionado.
<input>	minlength, maxlength	Definem a quantidade de caracteres mínima e máxima, respectivamente, permitida em um campo.
<input>, <button>	type	Define o tipo do campo e, sobretudo, como ele se comporta.
<option>	value	Este atributo também pode ser utilizado na tag <input>. Ele define o valor do campo. 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.

Tabela: Formulário HTML – Tags básicas e seus atributos.
Alexandre de Oliveira Paixão.

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**, ele 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.

Alguns outros tipos comuns são:

Password

Mascara o texto com asteriscos.

Hidden



Esconde o campo para não ser exibido no navegador.

Checkbox



Usado para seleção de valor através de click/check.

Radio



Usado para seleção exclusiva de valor – quando é apresentada mais de uma opção, apenas uma poderá ser selecionada, ao contrário do tipo “checkbox”.

Submit



Associada à tag <button>, dispara o evento que envia/submete o formulário.

Reset



Associada à tag <button>, dispara o evento que limpa os valores do formulário.

Button



Uma tag <input> pode ser do tipo “button” – exercendo, assim, a mesma função da tag <button>.

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, conforme veremos mais adiante – **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, e como é comum na evolução da HTML, a HTML5 definiu **novos tipos de entrada** e também **novos atributos relacionados** a formulários. Entre eles, podemos destacar:

Atributos		
Atributo	Função	Comentário
placeholder	Exibir um texto no campo de input.	Utilizado para dar uma dica ao usuário sobre o dado a ser inserido.
required	Determinar se um campo é obrigatório.	Utilizado na validação dos dados de um formulário.
autofocus	Fixar o foco no campo.	Utilizado quando desejamos que, ao carregar o formulário, um determinado campo seja focado.
pattern	Validar o valor de um campo com base em uma expressão regular. (As RegEX – expressões regulares – podem ser consideradas um recurso de linguagem de	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.

Atributos		
Atributo	Função	Comentário
	programação para a análise e manipulação de texto.)	
Tipos		
Tipo	Função	Comentário
tel	Inserir números de telefone.	Para uma melhor usabilidade, deve ser usado em conjunto com o atributo pattern.
datetime	Inserir datas com o fuso horário em UTC a partir de um calendário.	
date	Inserir datas sem fuso horário a partir de um calendário.	
number	Inserir números.	Cria um componente diferente do input normal, em que, por meio de setas, os números podem ser incrementados ou decrementados.

Tabela: Exemplos de novos atributos e tipos de dados definidos na HTML5
Alexandre de Oliveira Paixão.

Uma lista completa dos atributos e tipos pode ser encontrada nas referências mencionadas na seção Explore +.



O Atributo “Type”

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



Validação de dados em formulários

Validação de formulários

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 um outro aspecto relacionado a esses elementos que precisa ser discutido: **A validação de dados**.

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

Exemplo

Imagine que podemos permitir a inserção de uma data de nascimento através da 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.

Até há bem pouco tempo, para validar um formulário era necessário fazer uso de 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 de validação possíveis:

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

Que verifica se um campo obrigatório foi preenchido.

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 pré-definida: 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, neste 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]2[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”.

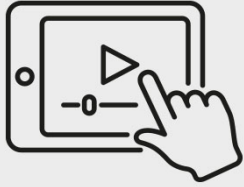
Atenção!

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>.



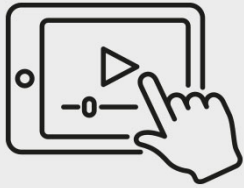
Como funciona a validação

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



Vamos praticar?

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



Falta pouco para atingir seus objetivos.

Vamos praticar alguns conceitos?

Questão 1

Sobre a composição e o comportamento do formulário HTML é incorreto afirmar que:

- A É formado por um ou mais elementos que permitem ao usuário interagir com a página, selecionando opções, inserindo textos etc.
- B 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.
- C Apenas os formulários validados podem ser enviados/submetidos.

- D Possui atributos que definem o endereço/URL para onde os dados serão enviados e o método como serão enviados.
- E As seções dentro de um formulário HTML servem para ajudar na separação dos campos no código e na visualização da página no navegador.

Parabéns! A alternativa C está correta.

Os formulários são compostos por tags que permitem a entrada e seleção de dados. Tal funcionalidade concede a este elemento a característica de interatividade, possibilitando que, a partir de ações do usuário, dados sejam enviados/informações sejam cadastradas ou comportamentos e conteúdo da própria página sejam modificados – como acontece, por exemplo, quando filtramos o resultado de uma pesquisa de produtos em um site de e-commerce. Embora existam diferentes tipos de campos de formulários, permitindo a inserção padronizada de diferentes tipos de dados, é prática recomendada validar os seus dados antes de serem enviados/submetidos.

Questão 2

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 validação de formulários na HTML5 é habilitada por padrão. Entretanto, é possível desabilitar este recurso com a utilização de um atributo na tag <form>.
- C A partir da HTML5, para facilitar o trabalho do desenvolvedor, novas tags foram criadas para que a validação aconteça apenas no lado servidor.
- 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 ao HTML5, devendo ser feita através de alguma linguagem de programação, como a linguagem Javascript.

Parabéns! A alternativa B 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.

Considerações finais

Vimos os conceitos de linguagens de marcação, mais especificamente da HTML. Ao longo dos tópicos, além dos conceitos e de sua história, conhecemos a sua estrutura e os elementos que compõem a linguagem de marcação, como tags e atributos, e para melhor fixação de cada conteúdo, utilizamos exercícios práticos.



Podcast

Para encerrar, ouça mais sobre linguagem de marcação de hipertexto.

Para ouvir o *áudio*, acesse a versão online deste conteúdo.



Referências

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

W3SCHOOLS. HTML Tutorial.

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

Explore +

Confira as indicações que separamos especialmente para você!

Pesquise o texto: **Estrutura de documentos e sites – MDN**.

Para conhecer as referências das tags HTML, leia: **HTML Element Reference – W3Schools**.