

História do HTML

O inglês Tim Berners-Lee é o criador da linguagem HTML e da World Wide Web em 1989.

Em 1991 foi criado o protocolo Gopher para navegação textual em documentos online.

Em 1991 foi publicado o primeiro website no endereço <http://info.cern.ch/hypertext/WWW/TheProject.html>.

O primeiro navegador com interface gráfica foi o Erwise em 1992.

O navegador ViolaWWW em 1992 foi o primeiro com suporte a linguagens de script e folhas de estilo.

O navegador Mosaic lançado em 1993 foi muito popular até 1997 e suportava vários protocolos como HTTP e FTP.

O Netscape lançado em 1994 implementou carregamento de páginas on-the-fly e se popularizou.

O Internet Explorer da Microsoft foi lançado em 1995 e logo dominou o mercado.

Em 1999 a Netscape foi vendida e seus programadores criaram o Mozilla Firefox.

O Google Chrome foi lançado em 2008 e teve grande adoção.

As versões do HTML ao longo do tempo foram: 1 em 1991, 2 em 1995, 3 em 1997, 4 em 1997, XHTML em 2000 e HTML5 em 2007.

O HTML5 prioriza semântica, separação de estrutura, estilo e interatividade.

Elementos básicos do HTML

Elementos HTML: a estrutura é formada por tags de abertura (<elemento>), conteúdo e tags de fechamento (</elemento>). Alguns elementos são de auto-fechamento como .

Listas: o elemento cria uma lista não ordenada com itens . Já o cria uma lista ordenada numericamente. Podem ser aninhados para criar sublistas.

Imagens: o elemento exibe imagens. O atributo src define o caminho da imagem e alt é o texto alternativo. Podemos definir largura e altura com width e height.

Áudio: o elemento <audio> incorpora áudio na página. Dentro dele vamos o elemento <source> que aponta para o arquivo de áudio. Podemos definir controles com o atributo controls.

Vídeo: o elemento <video> incorpora vídeo. Assim como no áudio, dentro dele vamos um <source> apontando para o arquivo de vídeo. Os atributos width, height e controls também se aplicam.

Iframes: o elemento <iframe> incorpora conteúdo externo na página, como vídeos do YouTube.

SVGs: arquivo de imagem vetorial no formato SVG que é descrito textualmente usando XML. Pode ser incluído com ou código SVG.

Caracteres especiais: exibidos com códigos como © para © e > para >.

Div e span: elementos genéricos sem semântica para agrupar conteúdo. Div ocupa largura total e span somente o necessário.

Elementos semânticos: nav, main, header, section, article, aside e footer provêm significado e ajudam na acessibilidade.

Estrutura básica: html engloba tudo, head possui metadados e body o conteúdo visível.

DOCTYPE: informa a versão do HTML ao navegador, como <!DOCTYPE html> para HTML5.

Meta tags: charset define o encoding e lang o idioma, importante para acessibilidade.

Links: o elemento <a> cria links entre páginas com o atributo href.

Formulários, tabelas e acessibilidade em HTML

Formulários:

O elemento form serve para agrupar e enviar os dados digitados pelo usuário. Não tem efeito visual por si só.

Input text cria caixa de texto simples. Email valida endereço de e-mail. Number aceita apenas números. E outros tipos criam controles específicos.

Label é texto clicável que serve como rótulo e está associado a um input pelo atributo for ou por aninhamento.

Select cria menu suspenso. Option define cada opção textual do menu.

Radio permite seleção única entre vários com mesmo name. São círculos selecionáveis.

Checkbox permite múltiplas seleções. São quadrados selecionáveis.

Textarea cria caixa de texto multilinhas com atributos rows e cols.

Tabelas:

Table contém as linhas da tabela com elemento tr.

tr contém os dados de cada coluna com td.

thead agrupa linha de cabeçalho da tabela.

tbody agrupa as linhas de dados.

tfoot representa rodapé da tabela.

Acessibilidade:

WAI-ARIA adiciona semântica com roles e propriedades para leitores de tela.

Progress e meter exibem barras de progresso com semântica.

Microdados no HTML seguem vocabulário do schema.org.

Tableless recomenda não usar table para layout e sim CSS.

Semântica correta com elementos apropriados melhora acessibilidade.

CSS

CSS é usado para estilizar elementos HTML

O CSS (Cascading Style Sheets) é uma linguagem de estilo utilizada para estilizar elementos HTML. Com o CSS, é possível controlar a aparência e o layout de uma página da web, tornando-a visualmente atraente e organizada. É uma parte essencial do desenvolvimento web moderno, pois permite separar o conteúdo (HTML) da apresentação (CSS), facilitando a manutenção e a flexibilidade do design de um site.

Principais versões: 1, 2 e 3. CSS3 é dividido em módulos

O CSS tem evoluído ao longo do tempo, com várias versões lançadas. As principais versões incluem CSS1, CSS2 e CSS3. A versão CSS3 é a mais recente e é dividida em módulos, o que significa que as especificações são divididas em pequenas partes para que os navegadores possam implementá-las gradualmente. Isso permite que os desenvolvedores utilizem recursos específicos do CSS3 conforme necessário, sem depender da implementação completa de toda a especificação.

Seletores: tipo, id, classe, descendentes, filhos diretos

Os seletores são parte fundamental das regras CSS, pois determinam quais elementos HTML serão estilizados por essas regras. Existem vários tipos de seletores, incluindo seletores de tipo (por exemplo, `p` para parágrafos), seletores de ID (por exemplo, `#minha-id` para um elemento com um ID específico), seletores de classe (por exemplo, `.minha-classe` para elementos com uma classe específica), seletores de descendentes (por exemplo, `div p` para estilizar parágrafos dentro de divs) e seletores de filhos diretos (por exemplo, `div > p` para estilizar apenas parágrafos que são filhos diretos de divs).

Regras CSS: propriedade: valor

As regras CSS consistem em pares de propriedade e valor, que definem como um elemento HTML deve ser estilizado. Por exemplo, a regra `color: blue;` define que a cor do texto deve ser azul. As propriedades determinam diversos aspectos do estilo, como cor, tamanho, margens, bordas e muito mais, enquanto os valores especificam as configurações específicas para essas propriedades.

Cascata: regras definidas por último sobrescrevem regras anteriores conflitantes

A cascata é um princípio fundamental do CSS que determina como as regras são aplicadas quando há conflitos. Quando várias regras afetam o mesmo elemento e têm a mesma especificidade, a regra definida por último em seu código CSS prevalece. Isso significa que as regras mais recentes sobrescrevem regras anteriores, o que permite a personalização e a adaptação do estilo de acordo com as necessidades específicas.

Especificidade: regras com seletores mais específicos sobrescrevem regras mais genéricas

A especificidade é um conceito importante no CSS que determina qual regra será aplicada quando há conflitos. As regras com seletores mais específicos têm uma prioridade maior do que as regras com seletores mais genéricos. Por exemplo, uma regra com um seletor de ID (`#minha-id`) é mais específica do que uma regra com um seletor de classe (`.minha-classe`). Portanto, a regra com o seletor de ID terá precedência sobre a regra com o seletor de classe, mesmo que a regra de classe seja definida posteriormente.

Cores: nomes, RGB, hexadecimal

O CSS oferece várias maneiras de definir cores para elementos. É possível utilizar nomes de cores predefinidas, como "red" (vermelho), "blue" (azul), entre outros. Além disso, as cores podem ser definidas em formato RGB (Red, Green, Blue), onde os valores de intensidade de cada cor são especificados, ou em formato hexadecimal, que utiliza uma combinação de números e letras para representar cores de forma mais precisa. Por exemplo, #FF0000 representa a cor vermelha.

Texto: font-family, font-size, font-style, font-weight

O CSS oferece um controle detalhado sobre a estilização de texto em páginas da web. A propriedade font-family permite especificar a família de fontes a ser usada, como "Arial" ou "Times New Roman". A propriedade font-size define o tamanho do texto, podendo ser especificada em unidades como pixels, ems ou porcentagens. A propriedade font-style permite definir o estilo da fonte, como "italic" (itálico) ou "normal" (normal). Por fim, a propriedade font-weight controla a espessura da fonte, podendo ser "bold" (negrito) ou valores numéricos como 400 ou 700.

Display: block, inline, inline-block, none

A propriedade display determina como um elemento HTML é exibido na página. Existem várias opções, incluindo:

block: O elemento é exibido como um bloco que ocupa toda a largura disponível e começa em uma nova linha.

inline: O elemento é exibido em linha com o texto e não começa em uma nova linha.

inline-block: O elemento é exibido em linha, mas permite a definição de largura, altura e margens.

none: O elemento não é exibido na página, tornando-o invisível.

Essas propriedades de display permitem um controle preciso sobre o layout e a estrutura de uma página da web.

Box Model: border, padding, margin, box-sizing

O modelo de caixa (Box Model) é um conceito central no CSS que define como os elementos HTML são renderizados em relação ao espaço ao seu redor. Cada elemento é composto por quatro partes principais: conteúdo, padding, borda e margem.

O conteúdo é a área onde o texto ou outros elementos são exibidos.

O padding é o espaço entre o conteúdo e a borda.

A borda é a linha que envolve o conteúdo e o padding.

A margem é o espaço entre a borda do elemento e outros elementos adjacentes.

A propriedade box-sizing permite controlar como as dimensões do elemento são calculadas, com opções como content-box (o tamanho total inclui apenas o conteúdo) e border-box (o tamanho total inclui conteúdo, padding e borda).

Responsividade: layouts fixos x fluidos, viewport, media queries

A responsividade é a capacidade de um site se adaptar a diferentes tamanhos de tela e dispositivos. Existem duas abordagens principais para criar layouts responsivos: layouts fixos e layouts fluidos.

Layouts fixos têm larguras definidas em pixels e não se adaptam automaticamente a tamanhos de tela menores.

Layouts fluidos usam unidades percentuais em vez de pixels, permitindo que o layout se ajuste à largura da tela.

Para tornar um site responsivo, os desenvolvedores utilizam a metatag viewport para controlar a escala inicial em dispositivos móveis e as "media queries" para aplicar estilos específicos com base nas características da tela, como largura, altura ou orientação.

Unidades: pixel, percentual. Pixel fixa medidas, percentual adapta

As unidades de medida no CSS desempenham um papel crucial na criação de layouts flexíveis e responsivos. Duas das unidades mais comuns são:

Pixel (px): É uma unidade de medida fixa, que não se adapta ao tamanho da tela. Usar pixels é útil quando se deseja um controle preciso sobre o tamanho e posicionamento dos elementos.

Percentual (%): É uma unidade de medida relativa que se adapta ao tamanho do elemento pai. Usar porcentagens é fundamental para criar layouts responsivos, pois permite que os elementos se ajustem ao tamanho da tela ou do elemento pai.

A escolha entre pixels e porcentagens depende das necessidades específicas do layout e do design.

Ferramentas: Chrome DevTools

O Chrome DevTools é uma ferramenta poderosa e amplamente usada para desenvolvedores web. Ele é integrado ao navegador Google Chrome e oferece uma variedade de recursos que auxiliam no desenvolvimento, depuração e otimização de sites. Com o DevTools, os desenvolvedores podem inspecionar elementos na página, modificar o CSS em tempo real, depurar JavaScript, analisar o desempenho da página e muito mais. É uma ferramenta essencial para o desenvolvimento web moderno.

Boas práticas: externalizar CSS, semântica

Seguir boas práticas no desenvolvimento web é fundamental para criar sites eficientes e de fácil manutenção. Duas boas práticas essenciais são:

Externalizar o CSS: Em vez de incluir estilos diretamente no HTML, é recomendável criar arquivos CSS externos. Isso torna o código mais organizado, facilita a manutenção e permite o reuso de estilos em várias páginas.

Usar semântica: A semântica HTML envolve a escolha adequada de elementos HTML para representar o conteúdo de forma significativa. Usar elementos HTML semânticos, como <header>, <nav>, <section>, <article>, entre outros, melhora a acessibilidade e a indexação por mecanismos de busca, além de tornar o código mais compreensível e legível.