



HTML + CSS

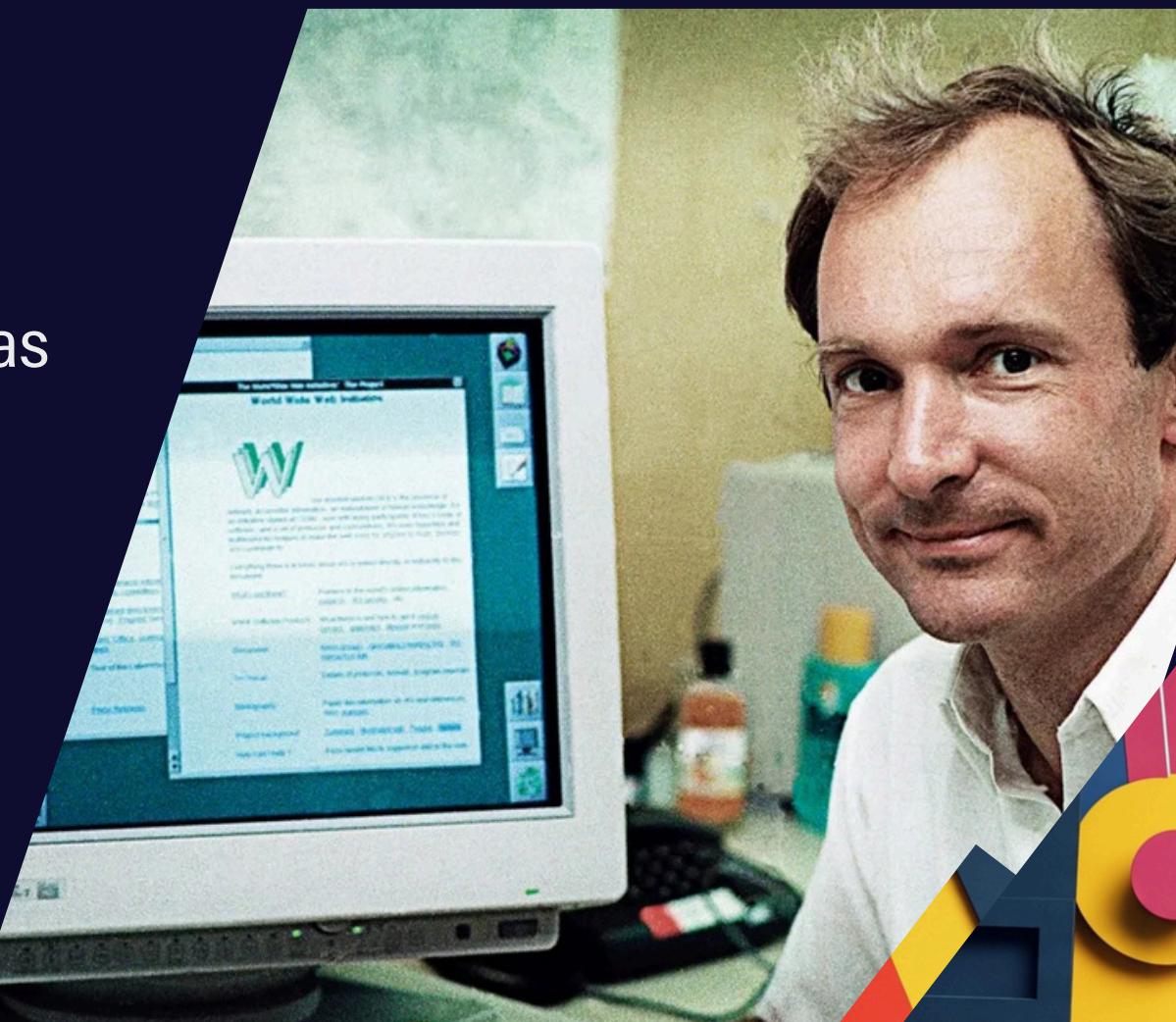
Introdução

HTML

Breve história

- **Criador:** Tim Berners-Lee em 1991.
- **Objetivo inicial:** permitir que cientistas compartilhassem documentos pela internet usando hipertexto.
- **Principais versões:**
 - HTML 1.0 (1993): básico, sem muitas funcionalidades.
 - HTML 2.0 (1995): padronizou a linguagem.
 - HTML 4.01 (1999): suporte a scripts e formulários.
 - HTML5 (2014 - atual): trouxe suporte a áudio, vídeo, semântica, APIs nativas (geolocalização, canvas, etc) e melhor integração com JavaScript.

Sem CSS



CSS

Breve história

- **Criador:** Håkon Wium Lie em 1994.
- **Motivação:** antes do CSS, a estilização era feita diretamente no HTML (com font, bgcolor, etc), o que tornava o código bagunçado.
- **Principais versões:**
 - CSS2 (1998): trouxe posicionamento, media types e mais.
 - CSS3 (2005+): modularizado em partes como transições, flexbox, animações, gradientes, etc.

Com CSS



HTML

O QUE É ISSO?

HTML (Linguagem de Marcação de Hipertexto), é uma **linguagem de marcação** usada para criar a estrutura de páginas da web.

- O termo "hipertexto" refere-se à capacidade de um documento conter links para outros documentos, permitindo a navegação entre eles.
- HTML fornece a base para a construção de páginas web, definindo como o conteúdo será organizado e exibido no navegador.

Como ele funciona?

Tags:

- O HTML utiliza tags para marcar elementos dentro do documento.

Estrutura:

- As tags definem a estrutura do documento, como a organização de títulos, parágrafos, listas, imagens, etc.

Navegadores:

- Os navegadores interpretam o código HTML e exibem o conteúdo da página web de acordo com a estrutura definida pelas tags.

</h1> </p>

</body>



CSS

O QUE É ISSO?

CSS (Folha de Estilo em Cascata) é o código utilizado para dar estilo às páginas Web.

- Com o CSS, você pode controlar a aparência dos elementos HTML, como cores, fontes, tamanho, espaçamento, bordas, layout, etc.
- O CSS é usado para criar layouts responsivos e flexíveis, que se adaptam a diferentes tamanhos de tela e dispositivos.

Como ele funciona?

Ele é estruturado em algumas partes, como:

- **Seletor:** O seletor especifica qual elemento HTML será estilizado com base em um id, classe ou alvo específico.
- **Bloco de declarações:** O bloco de declarações contém as propriedades que serão aplicadas em um seletor.
- **Propriedade:** A propriedade especifica qual aspecto do elemento você deseja estilizar (por exemplo, cor, fonte, tamanho).

“Grandes coisas não se fazem por impulso, mas pela junção de uma série de pequenas coisas.”

Van Gogh



Flexbox

Layout moderno e flexível

O Flexbox foi introduzido no CSS3 para facilitar o alinhamento e distribuição de espaço entre itens em um container, especialmente em layouts responsivos.



Situação problema

Imagine que você está montando uma interface de um aplicativo de tarefas. Você quer exibir uma lista de "coisas a fazer", uma embaixo da outra — tipo:

- Comprar pão
- Responder e-mails
- Estudar CSS



```
<div class="tarefas">  
  <div>🛒 Comprar pão</div>  
  <div>✉️ Responder e-mails</div>  
  <div>📚 Estudar CSS Flexbox</div>  
</div>
```

Aí você pensa: "Beleza, vou jogar tudo numa `<div>` e pronto." Mas quando renderiza... tá tudo grudado! Uma linha em cima da outra, sem respiro. **Então vem o toque de mágica:**

```
.tarefas {  
  display: flex;  
  flex-direction: column;  
  gap: 12px;  
}
```

De repente, sua lista respira. Cada tarefa tem seu espaço, bem organizadinho, uma embaixo da outra, como deve ser. O **flex-direction: column** fez os itens se empilharem verticalmente, e o **gap** deu aquele espaçamento elegante entre eles. Zero estresse com margem pra cima, pra baixo... só paz.



Propriedades

Principais instrumentos do display flex

- **flex-direction:** Define a direção que os itens vão se alinhar.
 - **row (padrão):** da esquerda pra direita.
 - **column:** de cima pra baixo.
 - **row-reverse:** da direita pra esquerda.
 - **column-reverse:** de baixo pra cima.
- **flex-wrap:** Permite que os itens quebrem linha caso não caibam.
 - **nowrap (padrão):** tudo numa linha só.
 - **wrap:** quebra pra próxima linha se necessário.
 - **wrap-reverse:** igual ao wrap, mas de baixo pra cima.
- **justify-content:** Controla o alinhamento horizontal dos itens (ou vertical, se a direção for column).
 - **flex-start:** encosta no início.
 - **flex-end:** encosta no final.
 - **center:** centraliza.
 - **space-between:** espaço entre os itens.
- **align-items:** Controla o alinhamento no eixo cruzado (vertical, se row; horizontal, se column).
 - **flex-start:** alinha no topo (ou esquerda).
 - **flex-end:** alinha embaixo (ou direita).
 - **center:** centraliza.

```
.container {  
  flex-direction: column;  
}
```

```
.container {  
  flex-wrap: wrap;  
}
```

```
.container {  
  justify-content: center;  
}
```

```
.container {  
  align-items: center;  
}
```



<https://developer.mozilla.org/en-US/docs/Glossary/Flexbox>

Vamos praticar?

Viagem espacial

<https://viagem-espacial.netlify.app/>

