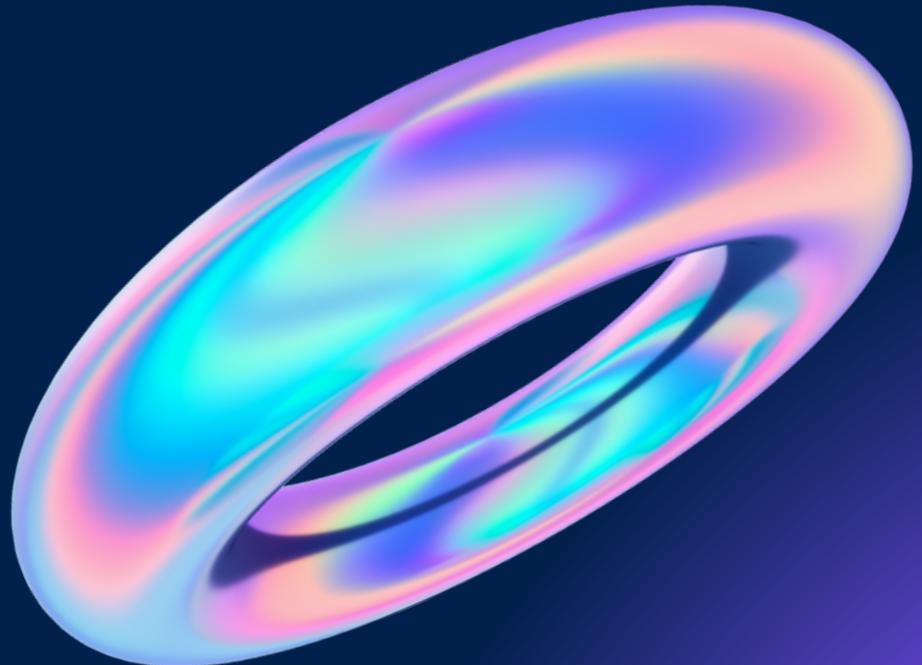




Apresentando

FLEXBOX CSS



Planejamento

O planejamento de layout é um passo crucial na criação de páginas da web eficazes e visualmente atraentes. Antes de mergulharmos no código CSS e na utilização do Flexbox, é fundamental estabelecer as bases corretas.



Inspiração e Pesquisa

- Importância da pesquisa
- Exploração de referências
- Melhores práticas do setor



Rascunho e Planejamento



Definir Paletas de Cores

- Esboçar ideias
- Visualizar estrutura
- Evitar retrabalho

- Escolha de cores
- Identidade da marca
- Experiência visual

Estrutura do código



Estrutura do Código HTML:

Organização das Divs e Subdivs

A organização é fundamental na estruturação do código HTML. Dividir seu layout em divs e sub-divs de forma lógica facilita a compreensão e manutenção do código. **Pense nas divs como blocos de construção e nas subdivisões como camadas que adicionam detalhes e estrutura.**

Nomeação de Classes e Subclasses

Escolher nomes descritivos para suas classes e subclasses é crucial. Use traços para separar palavras em nomes de classes, tornando-os mais legíveis. **Nomes descritivos refletem a função dos elementos e facilitam a colaboração com outros desenvolvedores.**

Reutilização de Elementos Pré-definidos

Otimize a eficiência da estilização reutilizando elementos pré-definidos. Frameworks CSS e bibliotecas oferecem estilos predefinidos que economizam tempo e mantêm a consistência visual em seu projeto.

Estilização do Código

Na etapa de estilização do código, é crucial considerar diversos fatores para garantir que seu layout seja atraente e eficaz.

CSS

Referenciais dos Marcadores

- Marcadores
- Alinhamento
- Organização

Nomenclatura Significativa

- Classes
- Tags
- IDs

Cores de Fundo

- Identificação
- Estilo
- Experiência Visual

Conhecendo o Flexbox

O Flexbox é uma ferramenta incrível que ajuda a organizar elementos em um site.

Ele é um conjunto de truques que tornam tudo isso mais fácil.

Inicialmente precisamos utilizar dentro da classe a propriedade **display** com o valor **flex** para podermos utilizar o Flexbox.

Exemplo de Código:

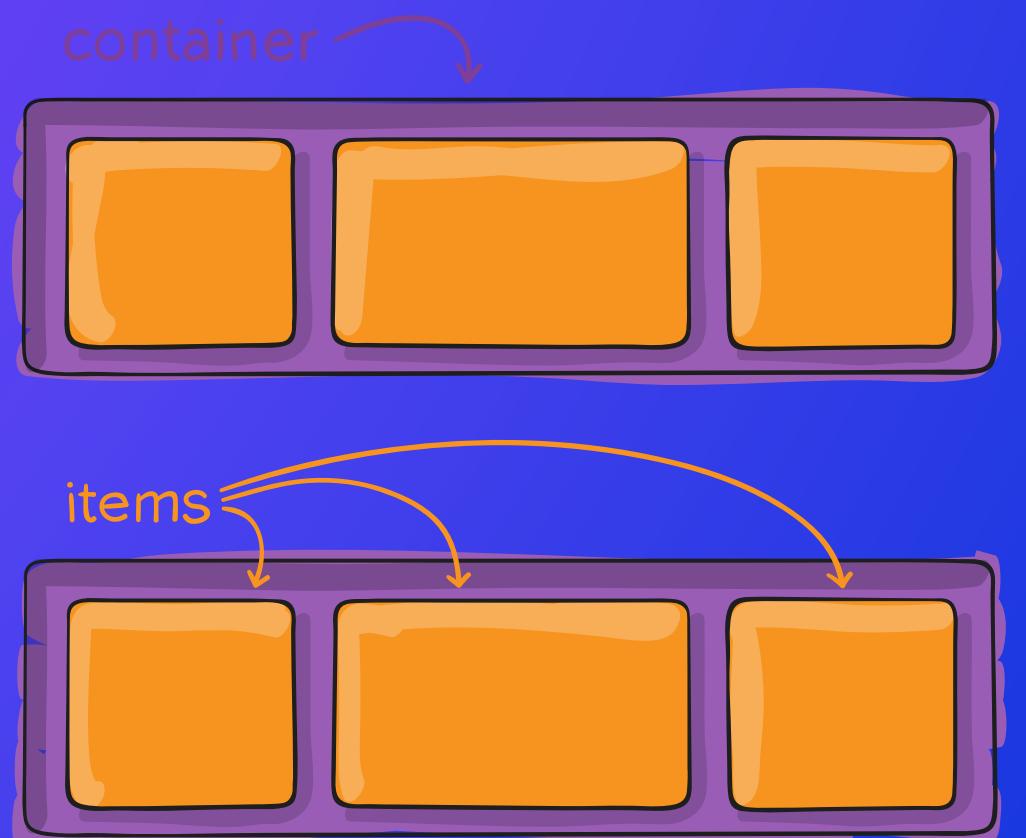
HTML:

```
<body>
  <div class="container">
    <div class="items"></div>
    <div class="items"></div>
    <div class="items"></div>
  </div>
</body>
```

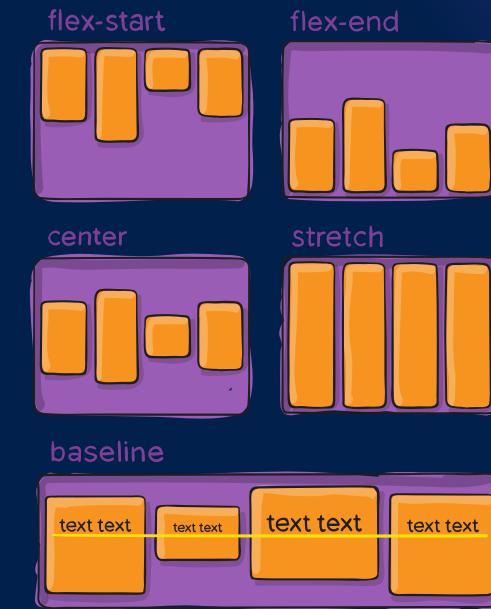
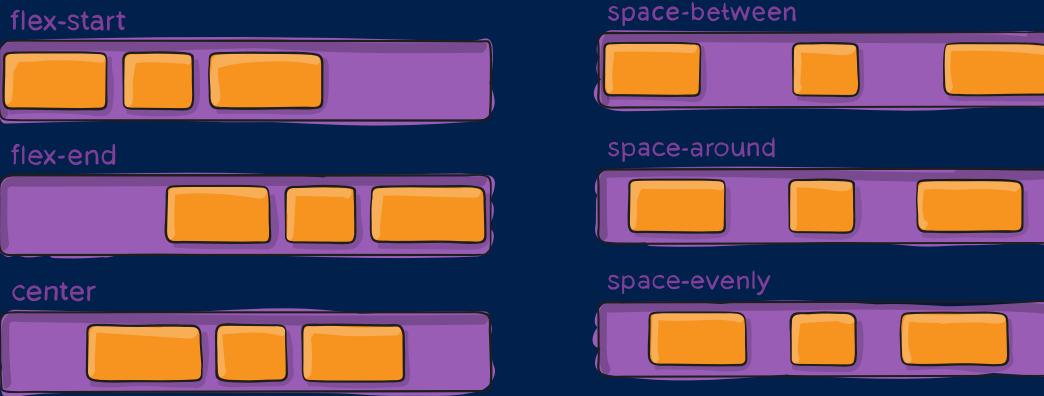
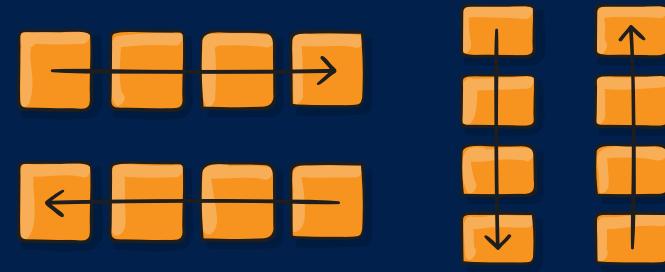
CSS:

```
.container {
  display: flex;
}

.items{
  background-color: orange;
}
```



Flexbox e suas Propriedades



flex-direction:

Direciona os elementos na horizontal ou vertical

Tendo como valores:

- row (default)
- row-reverse
- column
- column-reverse

```
.container{  
  display: flex;  
  flex-direction: row;  
}
```

justify-content:

Controla onde as coisas ficam no começo, no fim ou no meio da **caixa/flexbox**

Tendo como valores:

- flex-start (default)
- flex-end
- center
- space-between
- space-around
- space-evenly

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

align-items:

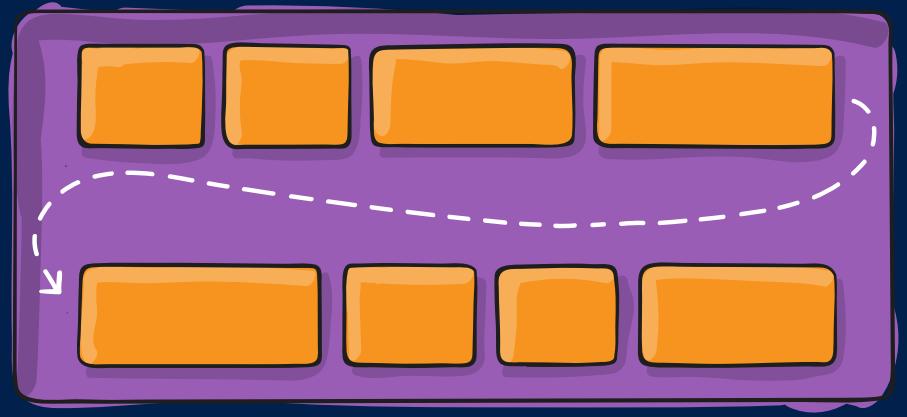
Decide como as coisas são alinhadas dentro da **caixa/flexbox**

Tendo como valores:

- stretch (default)
- flex-start
- flex-end
- center
- baseline

```
.container{  
  display: flex;  
  align-items: flex-end;  
}
```

Flexbox e suas Propriedades



flex-wrap:

Ele que decide se as coisas devem ir para uma nova linha ou se podemos dar um jeito de todas ficarem juntas.

Tendo como valores:

- nowrap (default)
- wrap
- wrap-reverse

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



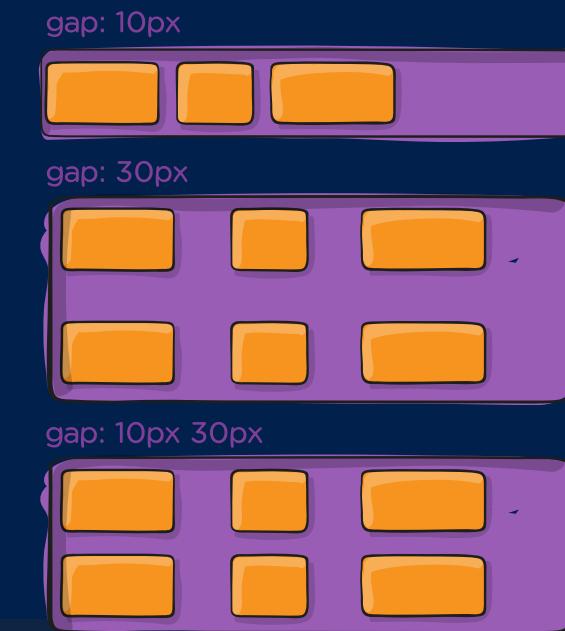
align-content:

Direciona os elementos de diversas linhas na horizontal ou vertical

Tendo como valores:

- flex-start (default)
- flex-end
- center
- space-between
- space-around
- space-evenly

```
.container{  
  display: flex;  
  flex-wrap: wrap;  
  align-content: flex-end;  
}
```



gap:

Ele é como um fôlego para seu layout, criando espaços entre seus elementos.

Tendo também:

- row-gap
- column-gap

```
.container {  
  display: flex;  
  gap: 10px; /* Isso cria um espaço entre todos os elementos */  
  row-gap: 15px; /* Espaçamento vertical entre elementos */  
  column-gap: 5px; /* Espaçamento horizontal entre elementos */  
}
```

Obrigado

Apresentado por:
Bruno; Enzo; Lucas.