Nzila-X

< Sessão 06 - Dev Web />

TIPOS DE LAYOUTS (CSS3)

29 de Julho de 2025

Ready, set, code! 🚀

DIA 07

MAY THE CODE BE WITH US

1	**Boas-vindas e revisão da aula anterior (5-10 min)**
2	CSS Flexbox (30 min)
3	Introdução ao SASS (30 min)
4	Introdução ao Git (20-25 min)
5	GitHub (20-25 min)
6	Caderno de exercícios (25-30 min)
#	**Tarefa de casa (5min)**

Nzila-X

#01

BOAS-VINDAS E REVISÃO DA AULA ANTERIOR

REVISÃO DA AULA ANTERIOR

1) HTML Semântico

2) Atributos Globais

3) Formulários

#01 - INTRODUÇÃO AO CSS

TIPOS DE LAYOUTS

Um layout define como os elementos HTML são organizados na página, controlando sua posição, tamanho e alinhamento. O CSS oferece vários modelos de layout, cada um com propósitos específicos, evoluindo do básico para soluções mais modernas e flexíveis.

POR QUE LAYOUTS SÃO IMPORTANTES?

Layouts são a base para organizar o conteúdo de uma página web de forma visualmente agradável, funcional e responsiva.

Nzila-X

TIPOS DE LAYOUTS

1) Layout Block (Bloco de Layout):

O layout de bloco é o comportamento padrão dos elementos HTML como , <p>, <h1>, etc., quando não há outras propriedades de layout aplicadas. Elementos de bloco ocupam toda a largura disponível e são empilhados verticalmente.

Para que serve?

- Organizar conteúdo em uma estrutura linear, como parágrafos ou seções.
- Ideal para layouts simples, como artigos de texto ou páginas com conteúdo estático.

Nzila-X

TIPOS DE LAYOUTS

Propriedades principais:

- display: block: Faz o elemento ocupar toda a largura do contêiner pai.
- margin e padding: Controlam espaçamento.
- width e height: Definem dimensões fixas ou relativas.

```
<div class="block-container">
  <div class="block-item">Item 1</div>
  <div class="block-item">Item 2</div>
</div>
```

Nzila-X

TIPOS DE LAYOUTS

Layout de Bloco (CSS)

```css
.block-container {
  background-color: #f0f0f0;
  padding: 10px;
}
.block-item {
  display: block;
  background-color: lightblue;
  margin: 10px 0;
  padding: 10px;
}
```

Nzila-X

TIPOS DE LAYOUTS

2) Layout Inline e Inline-Block

Elementos com display: inline (como ou <a>) fluem na mesma linha, enquanto display: inline-block combina características de inline (fluir na linha) e block (permitir largura/altura definidas).

Para que serve?

- inline: Organizar elementos pequenos, como links ou ícones, na mesma linha.
- inline-block: Criar elementos que ficam na mesma linha, mas com controle de dimensões e margens.

Nzila-X

TIPOS DE LAYOUTS

Propriedades principais:

```html
<div class="inline-container">
  <span class="inline-item">Item 1</span>
  <span class="inline-item">Item 2</span>
  <div class="inline-block-item">Item 3</div>
  <div class="inline-block-item">Item 4</div>
</div>
```

```css
.inline-container {
  background-color: #f0f0f0;
  padding: 10px;
}
```

Nzila-X

TIPOS DE LAYOUTS

Propriedades principais:

```
.inline-item {
  display: inline;
  background-color: lightgreen;
  padding: 5px;
}
.inline-block-item {
  display: inline-block;
  width: 100px;
  background-color: lightcoral;
  padding: 5px;
}
```

Nzila-X

TIPOS DE LAYOUTS

3) Layout com Floats

A propriedade float (ex.: float: left ou float: right) permite que elementos flutuem para um lado do contêiner, com outros elementos fluindo ao redor.

Para que serve?

- Criar layouts como colunas de texto ao lado de imagens.
- Usado em layouts de grade simples antes do Flexbox e Grid.

Nzila-X

TIPOS DE LAYOUTS

Layout Float

```html
<div class="float-container">
  <div class="float-item">Item 1</div>
  <div class="float-item">Item 2</div>
</div>
```

```css
.float-container {
  overflow: auto; /* Corrige colapso do contêiner */
  background-color: #f0f0f0;
  padding: 10px;
}
```

Nzila-X

TIPOS DE LAYOUTS

Layout Float

```html
<div class="float-container">
  <div class="float-item">Item 1</div>
  <div class="float-item">Item 2</div>
</div>
```

```css
.float-container {
  overflow: auto; /* Corrige colapso do contêiner */
  background-color: #f0f0f0;
  padding: 10px;
}
```

Nzila-X

TIPOS DE LAYOUTS

Layout Float

```
.float-item {
  float: left;
  width: 45%;
  margin: 2.5%;
  background-color: lightblue;
  padding: 10px;
}
```

Nzila-X

TIPOS DE LAYOUTS

4) Layout com Posicionamento Absoluto/Fixo

A propriedade float (ex.: float: left ou float: right) permite que elementos flutuem para um lado do contêiner, com outros elementos fluindo ao redor.

Para que serve?

- position: absolute: Posiciona elementos fora do fluxo normal, com base em coordenadas (top, left, etc.).
- position: fixed: Fixa elementos na janela do navegador (ex.: barras de navegação fixas).
- position: sticky: Combina comportamento fixo e relativo, útil para cabeçalhos que "grudam" ao rolar.

TIPOS DE LAYOUTS

Absoluto/Fixo

```html
<div class="position-container">
  <div class="absolute-item">Item Absoluto</div>
  <div class="normal-item">Item Normal</div>
</div>
```

```css
.position-container {
  position: relative;
  height: 200px;
  background-color: #f0f0f0;
  padding: 10px;
}
```

Nzila-X

TIPOS DE LAYOUTS

Absoluto/Fixo

```html
<div class="position-container">
  <div class="absolute-item">Item Absoluto</div>
  <div class="normal-item">Item Normal</div>
</div>
```

```css
.position-container {
  position: relative;
  height: 200px;
  background-color: #f0f0f0;
  padding: 10px;
}
```

Nzila-X

TIPOS DE LAYOUTS

Absoluto/Fixo

```
.absolute-item {
  position: absolute;
  top: 20px;
  left: 20px;
  background-color: lightblue;
  padding: 10px;
}
.normal-item {
  background-color: lightgreen;
  padding: 10px;
}
```

Nzila-X

TIPOS DE LAYOUTS

5) Layout com Flexbox

O Flexbox (Flexible Box Layout) é um modelo de layout unidimensional que organiza elementos em linhas ou colunas, com controle avançado sobre alinhamento, espaçamento e dimensionamento.

Para que serve?

- Criar layouts responsivos com alinhamento dinâmico.
- Distribuir espaço entre elementos de forma flexível.
- Ideal para navegações, barras laterais, cartões e layouts de uma dimensão.

TIPOS DE LAYOUTS

2) Flexbox:

Layout unidimensional para alinhar elementos numa linha ou coluna.

Propriedades principais:

- `display: flex;` : Ativa o Flexbox no contêiner.
- `flex-direction` : Define a direção (`row` , `column`).
- `justify-content` : Alinha itens no eixo principal (`flex-start` , `center` , `space-between`).
- `align-items` : Alinha itens no eixo transversal (`flex-start` , `center` , `stretch`).

Nzila-X

TIPOS DE LAYOUTS

6) Vantagens:

- Fácil de alinhar e distribuir elementos.
- Suporta layouts responsivos sem hacks.
- Controla crescimento e encolhimento de elementos.

TIPOS DE LAYOUTS

2) Flexbox:

```
<div class="container">
  <div class="item">Item 1</div>
  <div class="item">Item 2</div>
  <div class="item">Item 3</div>
</div>
```

TIPOS DE LAYOUTS

2) Flexbox:

```css
.container {
  display: flex;
  flex-direction: row;
  justify-content: space-between;
  align-items: center;
}
.item {
  background-color: #007bff;
  color: white;
  padding: 10px;
}
```

Nzila-X

TIPOS DE LAYOUTS

2) Grid:

Layout bidimensional para criar grelhas com linhas e colunas.

Propriedades principais:

- `display: grid;` : Ativa o Grid no contêiner.
- `grid-template-columns` : Define colunas (ex.: `1fr 1fr` para duas colunas iguais).
- `grid-template-rows` : Define linhas.
- `gap` : Espaço entre células.

TIPOS DE LAYOUTS

2) Grid:

```
<div class="grid-container">
  <div class="grid-item">Item 1</div>
  <div class="grid-item">Item 2</div>
  <div class="grid-item">Item 3</div>
</div>
```

Nzila-X

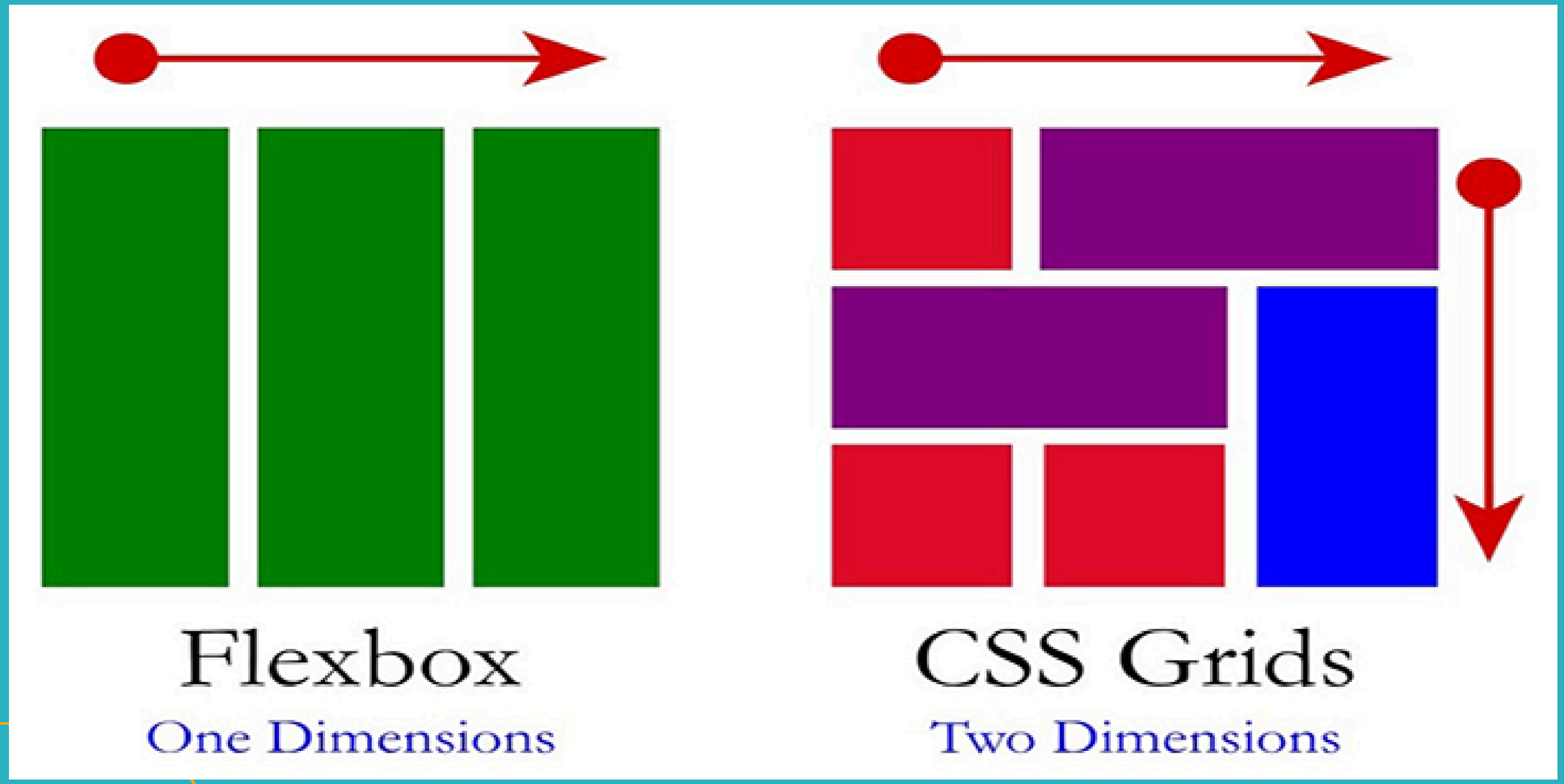
TIPOS DE LAYOUTS

2) Grid:

```
.grid-container {
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
  gap: 10px;
}
.grid-item {
  background-color: #28a745;
  color: white;
  padding: 10px;
}
```

Nzila-X

TIPOS DE LAYOUTS



Nzila-X

EXERCÍCIO #04

1. Cria um ficheiro `estilos.css` e liga-o ao `sobre.html` do exercício de HTML.
2. Estiliza a página com:
   - O `<header>` com fundo azul e texto branco.
   - O formulário com borda e padding.
   - A lista de navegação com itens alinhados horizontalmente usando Flexbox (`display: flex`, `justify-content: space-around`).
   - Uma secção `<main>` com três artigos dispostos numa grelha de 3 colunas usando Grid (`grid-template-columns: 1fr 1fr 1fr`).

Nzila-X

#03 - INTRODUÇÃO AO SASS


Nzila-X
Q&A
-time

TASK 02:

Criação de uma página HTML com CSS e SASS

TASK 02:

> **Criação de um Mini Site Currículo Pessoal**

1. Página HTML com estrutura semântica.
2. Uso de formulários para contacto.
3. CSS para estilizar toda a página.

POR HOJE, É TUDO!

ada linha de código que escreves é um passo para criares experiências digitais incríveis – continua a aprender e a construir!

17/07/2025