Posicionamento

Aprenda mais sobre como utilizar classes relacionadas à *position, overflow, z-index, box-sizing* entre outros que regem a posição espacial do seu elemento visual dentro do ambiente de trabalho.

- Posicionamento
- Transbordamento

Posicionamento

Essas classes oferecem diferentes opções de posicionamento para elementos HTML. Elas são úteis ao precisar controlar a posição de um elemento em relação ao seu contêiner ou à janela do navegador.

Classes Disponíveis

- `.static`: Define o posicionamento do elemento como estático, ou seja, ele segue o fluxo normal do documento e não é afetado por outros elementos.
- `.sticky`: Define o posicionamento do elemento como "sticky", o que significa que ele se comporta como "relative" até que atinja um determinado ponto de rolagem. Após atingir esse ponto, ele se torna "fixed", mantendo-se fixo na tela.
- `.relative`: Define o posicionamento do elemento como relativo ao seu posicionamento normal. Isso permite que você mova o elemento usando as propriedades **`top`**, **`right`**, **`bottom`** e **`left`**.
- `.absolute`: Define o posicionamento do elemento como absoluto em relação ao seu ancestral mais próximo que possui uma posição diferente de `static`. Se nenhum ancestral for encontrado, o elemento será posicionado em relação ao elemento `<html>`.
- `.fixed`: Define o posicionamento do elemento como fixo em relação à janela do navegador. Ele permanecerá na mesma posição, independentemente da rolagem da página.

Uso

Para aplicar esses estilos de posicionamento a um elemento HTML, basta adicionar a classe correspondente conforme necessário. Por exemplo:

```
<div class="static">
  Este é um elemento com posicionamento estático.
</div>

<div class="sticky">
  Este é um elemento com posicionamento "sticky".
</div>

<div class="relative">
  Este é um elemento com posicionamento relativo.
</div>
```

```
<div class="absolute">
    Este é um elemento com posicionamento absoluto.
</div>

<div class="fixed">
    Este é um elemento com posicionamento fixo.
</div>
```

Cada classe de posicionamento alterará o comportamento do elemento conforme descrito acima, permitindo um controle preciso sobre o layout e a aparência da página.

Configurando posicionamento por funções

Há casos onde será necessário instruir o elemento a se comportar de uma maneira ainda mais específica e, portanto, o WPF oferece algumas funções que auxiliam o usuário a configurar o posicionamento do seu elemento de forma específica diretamente no seu arquivo SCSS/SASS:

Confira elas aqui:

setPosition()

centerPosition()

Esses mixins oferecem maneiras simples e flexíveis de configurar o posicionamento de elementos em seus projetos, permitindo uma maior personalização e controle sobre o layout.

Além dos mixins, você também pode optar por usar uma versão dessas funções *em classe.* Juntamente com as classes auxiliares que definem um posicionamento. Se elas forem `absolute`, `relative` ou `fixed` , é possível utilizar as funções `top()`, `right()`, `bottom()` e `left()`, com valores em **%, px, dvh** ou **dvw** para setar valores **diretamente no atributo class** do seu componente HTML. É possível até mesmo utilizar o mixin centerPosition() via classe com a função `center()`.

Transbordamento

As classes de overflow são utilizadas para definir o comportamento de overflow (transbordamento) de conteúdo para um elemento. Elas oferecem opções para controlar o comportamento de overflow em várias direções.

Uso Geral:

```
<div class="overflow-y">
   <!-- Conteúdo aqui -->
</div>
```

Opções Disponíveis:

- **`.overflow`**:
  - Aplica **`overflow: auto !important;`** para habilitar barras de rolagem quando necessário.
- **`.overflow-hidden`**:
  - Aplica **`overflow: hidden !important;`** para ocultar o conteúdo que ultrapassa os limites do elemento.
- **`.overflow-scroll`**:
  - Aplica **`overflow: scroll !important;`** para exibir barras de rolagem, independentemente se o conteúdo ultrapassa os limites ou não.
- **`.overflow-visible`**:
  - Aplica **`overflow: visible !important;`** para exibir todo o conteúdo, mesmo que ele ultrapasse os limites do elemento.

Opções de Direção:

As classes também oferecem opções para controlar o comportamento de overflow em direções específicas: vertical e horizontal.

- **`.overflow-y`**:
  - Aplica **`overflow-y: auto !important;`** para habilitar barras de rolagem vertical quando necessário.
- **`.overflow-y-hidden`**:
  - Aplica **`overflow-y: hidden !important;`** para ocultar o conteúdo que ultrapassa os limites verticalmente.

- **`.overflow-y-scroll`**:
  - Aplica **`overflow-y: scroll !important;`** para exibir barras de rolagem vertical, independentemente se o conteúdo ultrapassa os limites ou não.
- **`.overflow-y-visible`**:
  - Aplica **`overflow-y: visible !important;`** para exibir todo o conteúdo verticalmente, mesmo que ele ultrapasse os limites do elemento.
- **`.overflow-x`**:
  - Aplica **`overflow-x: auto !important;`** para habilitar barras de rolagem horizontal quando necessário.
- **`.overflow-x-hidden`**:
  - Aplica **`overflow-x: hidden !important;`** para ocultar o conteúdo que ultrapassa os limites horizontalmente.
- **`.overflow-x-scroll`**:
  - Aplica **`overflow-x: scroll !important;`** para exibir barras de rolagem horizontal, independentemente se o conteúdo ultrapassa os limites ou não.
- **`.overflow-x-visible`**:
  - Aplica **`overflow-x: visible !important;`** para exibir todo o conteúdo horizontalmente, mesmo que ele ultrapasse os limites do elemento.