

INSTITUTO FEDERAL

Rio Grande do Norte

Campus Caicó

Design Web e Arquitetura da Informação

Prof. Romerito Campos

Plano de Aula

- **Objetivo:** Compreender o da propriedade `position`
- **Conteúdos:**
 - Position: definição
- Position: static, absolute, relative, sticky e fixed

position

position

- A propriedade `position` indica como podemos posicionar um elemento para sua renderização.
- O valor padrão desta propriedade é `position: static`.
 - Não é necessário indicar este valor
 - Todos os elementos por definição já vem com esta propriedade definida assim.
- Há 5 positions: `static`, `relative`, `absolute`, `fixed` e `sticky`.

position

- Um conceito importante sobre position é a ideia de **Containing Block**.
- Basicamente, ele indica qual é o box que contém um elemento.
- Por exemplo, a tag `<html>` é o containing block da tag `<body>`.
- Ou seja, o containing block é o elemento pai em relação aos os elementos filhos.

Position

- Exemplo de Containing Block

```
<!-- trecho de uma página -->
<body>
  <div class="container">
    </div>
</body>
```

- A tag `<body>` é o containing block do elemento `<div>`.

Position

- É importante considerar o Containing block para saber onde o elemento começa a ser desenhando.
- Vimos que os elementos são desenhados na tela considerando o **fluxo normal de renderização**.
 - Em idiomas cuja escrita é da esquerda para direita: português, inglês entre outros
 - Portanto, o fluxo normal é o mesmo da escrita
- Além disso, devemos considerar que os elementos tem níveis de `block` ou `inline`.

Position

- Por que preciso saber sobre *containing block* e nível de bloco|inline?
- Para alterar o posicionamento de um elemento é necessário ter noção de onde ele inicia o seu desenho.
- Veja a imagem no próximo slide:

-
- Vejamos este exemplo:



- Ele contém uma `div` com um `h1` dentro conforme o código a seguir.

-
- Código do exemplo anterior

```
<body>
  <div>
    <h1>Position</h1>
  </div>
</body>
```

- O containing block da `div` é o `body`.

- Ele está no fluxo normal de renderização: esquerda para direita.
- O ponto inicial da div é `top: 0` e `left: 0`.
 - canto superior esquerdo do `body`

-
- Essa noção é importante porque a propriedade `position` se relaciona com as propriedades `left`, `top`, `bottom` e `right`
 - `top: 0` e `left: 0` indica o canto superior esquerdo
 - `top: 0` e `right: 0` indica o canto superior direito.
 - Em essência utilizaremos `position` e essas propriedades citadas para:

Retirar os elementos do fluxo normal de renderização

position: relative

position: relative

- O posicionamento relativo permite que se retire o elemento do fluxo normal em relação ao seu posicionamento original.
- Ao sair do local que seria sua posição original, o espaço que o elemento deveria ocupar é mantido.
- Vejamos um exemplo para entender essas definições [Código-Fonte](#):

-
- Na imagem temos:
 - Fluxo Normal;
 - `position: relative` com deslocamento do elemento a partir do topo (`top: 25px`)
 - `position: relative` com deslocamento no topo e na esquerda: `top: 25px; left: 25px`



- No primeiro conjunto de quadrados não temos alteração no fluxo de desenho.
- No segundo caso, perceba que o elemento foi deslocado para abaixo.
 - Ele saiu de sua posição original;
 - Ultrapassa o limite da `div` que o contém;
 - Mas seu espaço original é mantido.
- No terceiro caso, acontece o mesmo que no caso dois. Além disso, temos um deslocamento da esquerda para direita
 - Observe que o elemento sobrepôs a terceira caixinha ao ser deslocamento.

-
- Este é o comportamento relativo.
 - Segue o código que modifica os elementos apresentados.

```
.top {  
  position: relative;  
  top: 25px;  
  border: 1px solid;  
}
```

```
.left {  
  position: relative;  
  left: 25px;  
  border: 1px solid;  
}
```

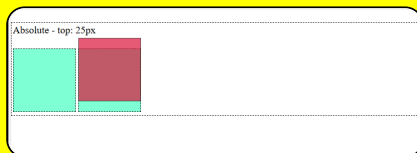
- Como desafio, crie o exemplo completo.

position: absolute

position: absolute

- O posicionamento absoluto indica o local em específico onde o elemento é desenhado.
- Além disso, ele não preserva o espaço original do elemento como acontece com `position: relative`.
- Com o `position: absolute` é muito fácil compreender a ideia de **containing block** apresentado no início.
- Observe a imagem a seguir:

-
- Exemplo de `position: absolute`:



- Vejamos os detalhes.

-
- É notável que uma caixa esta sobrepondo a outra.
 - No caso, a caixa com tom avermelhado está deslocada de sua posição.
 - Além disso, o elemento seguinte que é uma caixa ocupou seu lugar.
 - Neste caso, o deslocamento aplicado (`top: 25px`) não funciona da mesma maneira que o `position: relative`.
 - Observe que não definimos `left`.
 - O `position absolute` vai usar o **containing block** `<body>` ao invés de usar a `div`.

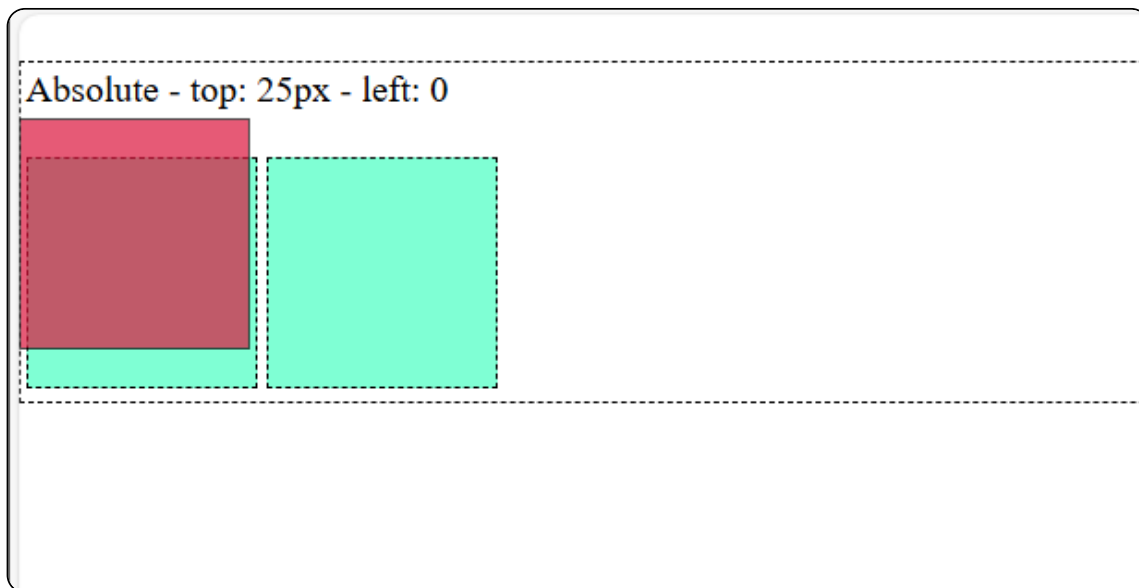
-
- Portanto, o elemento considera o `top: 0` como sendo o limite do `body`. Ele não considera a `div`.
 - A posição em relação a `left` permanece a posição que deveria ser a original no fluxo normal. Veja a regra CSS:

```
.top {  
  position: absolute;  
  top: 25px;  
  border: 1px solid;  
  background-color: crimson;  
  opacity: 70%;  
}
```

- Agora considere que vamos aplicar a seguinte regra CSS:

```
.top {  
  position: absolute;  
  top: 25px;  
  left: 0px;  
  border: 1px solid;  
  background-color: crimson;  
  opacity: 70%;  
}
```

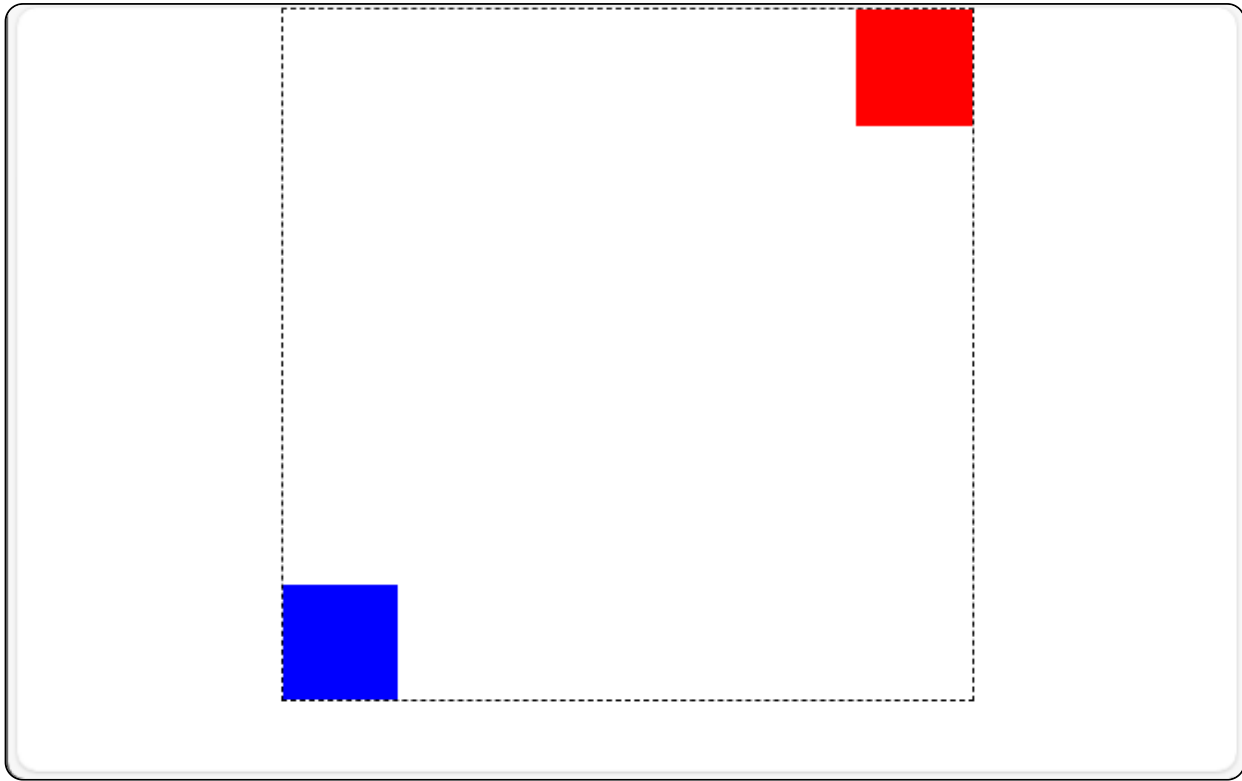
- Mantenha em mente que o **containing block** é o **body**
- O que será que vai acontecer? Vejamos no slide seguinte.



- Na imagem do slide anterior, temos o resultado da aplicação de `position: absolute` junto com `top` e `left`.
- O elemento será posicionado a partir da esquerda (`left`) da página na posição correspondente a 0.
 - Isso corresponde a ficar na borda da página.
- Com relação ao topo (`top`), mantém-se o distanciamento da borda superior da página.
- Isso acontece porque o **containing block** padrão para `position: absolute` é o viewport (dimensão da página).

position: relative e position: absolute

- Caso seja necessário alterar o **containing block** para um elemento com `position: absolute`, podemos relacioná-lo com o `position: relative`.
 - O próximo exemplo explora a relação entre uma div (`position: relative`) com dois elementos filhos (`position: absolute`)
-



- O código HTML é o seguinte:

```
<body>
  <div class="container">
    <div class="caixa top-right">
  </div>
    <div class="caixa bottom-left">
  </div>
  </div>
</body>
```

- Neste exemplo temos:
 - `.container: position: relative`
 - `.caixa: position: absolute`

- Além disso, temos também as classes:

```
.top-right {
  top: 0;
  right: 0;
  background-color: red;
}
```

```
.bottom-left{
  bottom: 0;
```



```
left: 0;  
background-color: blue;  
}
```

- Observe que aplicamos as propriedades para determinar o posicionamento absoluto para as caixas coloridas.
- A posição top:0 para a caixa vermelha considera a div. Neste caso, não considera a página (body) como sendo seu **containing block**.
- Ou seja a referência para o absolute é o elemento pai da caixa.

position: fixed

position: fixed

- O posicionamento fixado é semelhante ao posicionamento absoluto em alguns aspectos:
 - O elemento é retirado do fluxo normal;
 - Seu espaço original é ocupado;
- A diferença crucial é a seguinte: dada uma página que apresenta rolagem na horizontal, o elemento fixado ficará sempre no mesmo local independentemente da movimentação da página.

position: fixed

