Semântica II
& box model

O que aprendemos na aula anterior?

Formatar textos, formulários HTML, imagens, pseudo seletores em CSS

Seções não semânticas

`<div> </div>`
Divisão de conteúdo.

`<span> </span>`
Tag multifuncional (não é um bloco).

`<pre> </pre>`
Texto pré-formatado.

Seções semânticas

<section> </section>
Seção de conteúdo monotemático.

<article> </article>
Informações dentro de uma seção.

<header> </header>
Cabeçalho do conteúdo ou de um documento.

<footer> </footer>
Rodapé do conteúdo ou de um documento.

Tipos de elementos

display

inline

Define um elemento com comportamento de linha. Não recebe algumas propriedades do modelo de caixa.

block

Define um elemento com comportamento de bloco. Recebe “facilmente” propriedades do modelo de caixa.

Tipos de elementos

display

inline-block

Define um elemento com comportamento de semibloco, recebe “facilmente” propriedades do modelo de caixa. Também tem propriedades do elemento de linha.

none

Oculta visualmente um elemento sem eliminá-lo da estrutura de HTML. O efeito é apenas visual.

Tipos de elementos

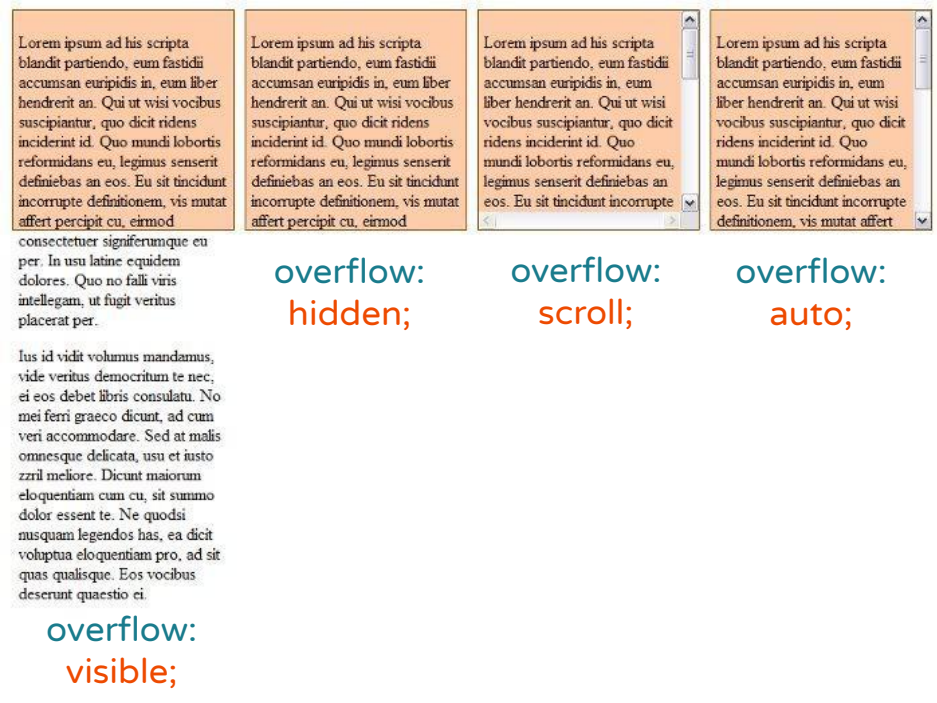
display

display	padding	margin	width/height	quebra de linha
inline				
block	✔	✔	✔	✔
inline-block	✔	✔	✔	
none				

Vamos testar!

Propriedades CSS

overflow



Propriedades CSS

width

```
elemento {
    width: 560px;
}
```

Atenção: se um elemento não tiver a width declarada, ela será igual a 100% do container pai, desde que seja um bloco.

Propriedades CSS

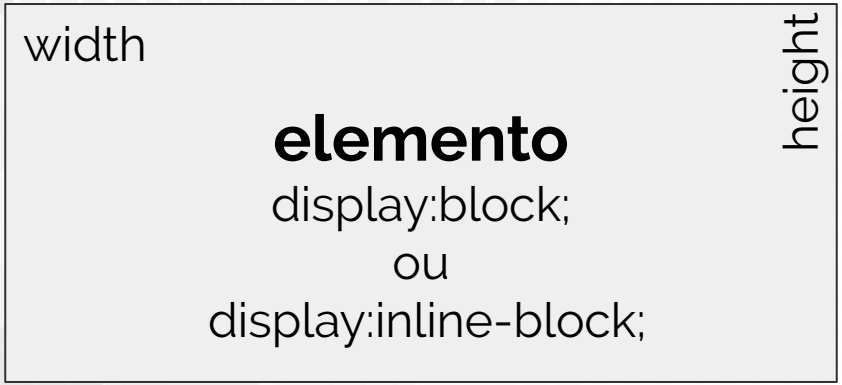
height

```
elemento {
    height: 560px;
}
```

Atenção: se um elemento não tiver a height declarada, ela será igual à altura do conteúdo interno, seja ele um bloco ou uma linha.

Título Principal

width/height



Propriedades CSS

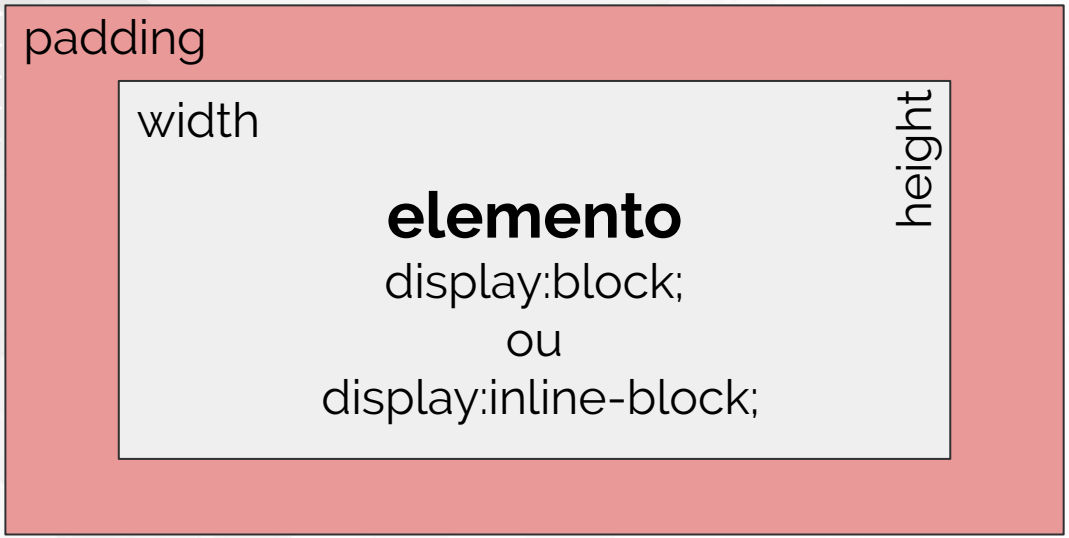
padding

```
elemento {
    padding-top: 10px;
    padding-right: 20px;
    padding-bottom: 30px;
    padding-left: 40px;

    /* shorthand */
    padding: 10px 20px 30px 40px;
}
```

Propriedades CSS

padding



Propriedades CSS

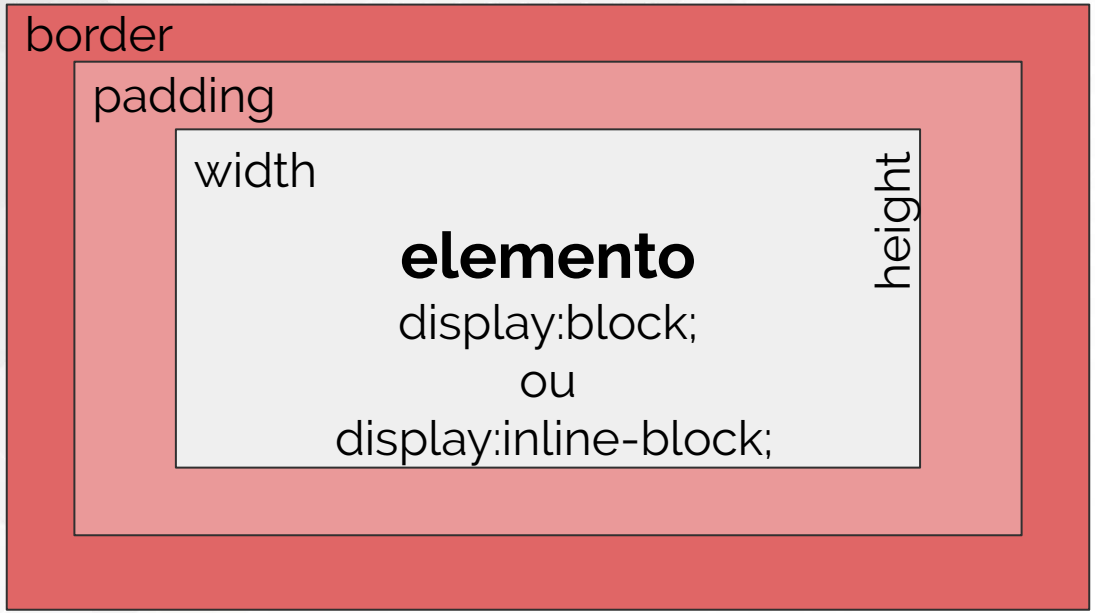
border

```
elemento {
    border-style: solid;
    border-width: 2px;
    border-color: #FF0000;

    /* shorthand */
    border: solid 2px #FF0000;
}
```

Propriedades CSS

border



Propriedades CSS

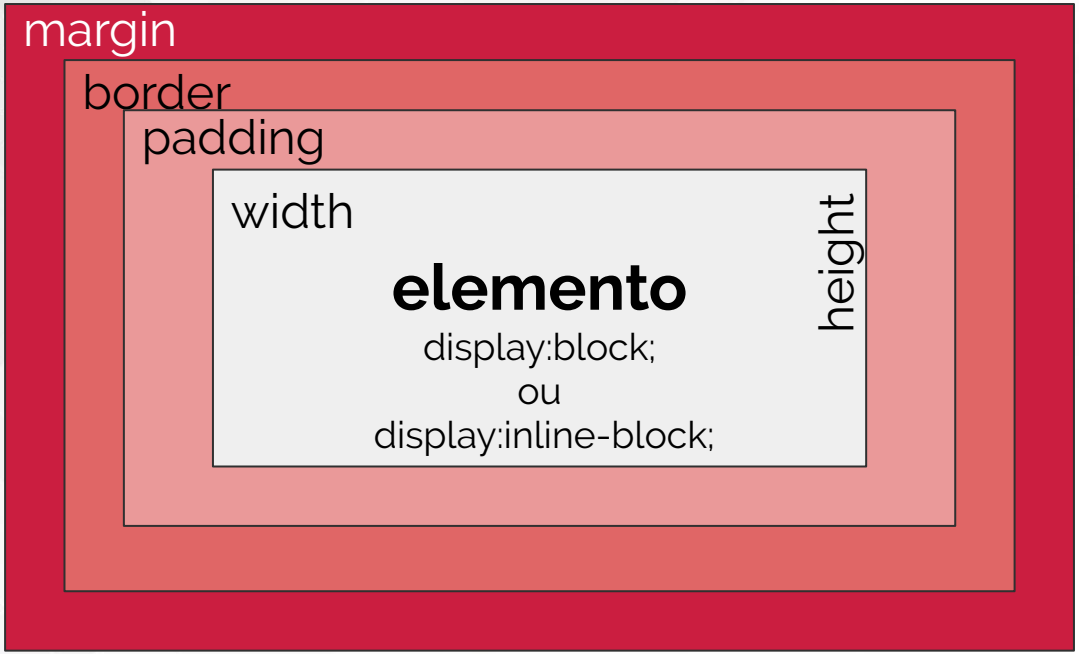
margin

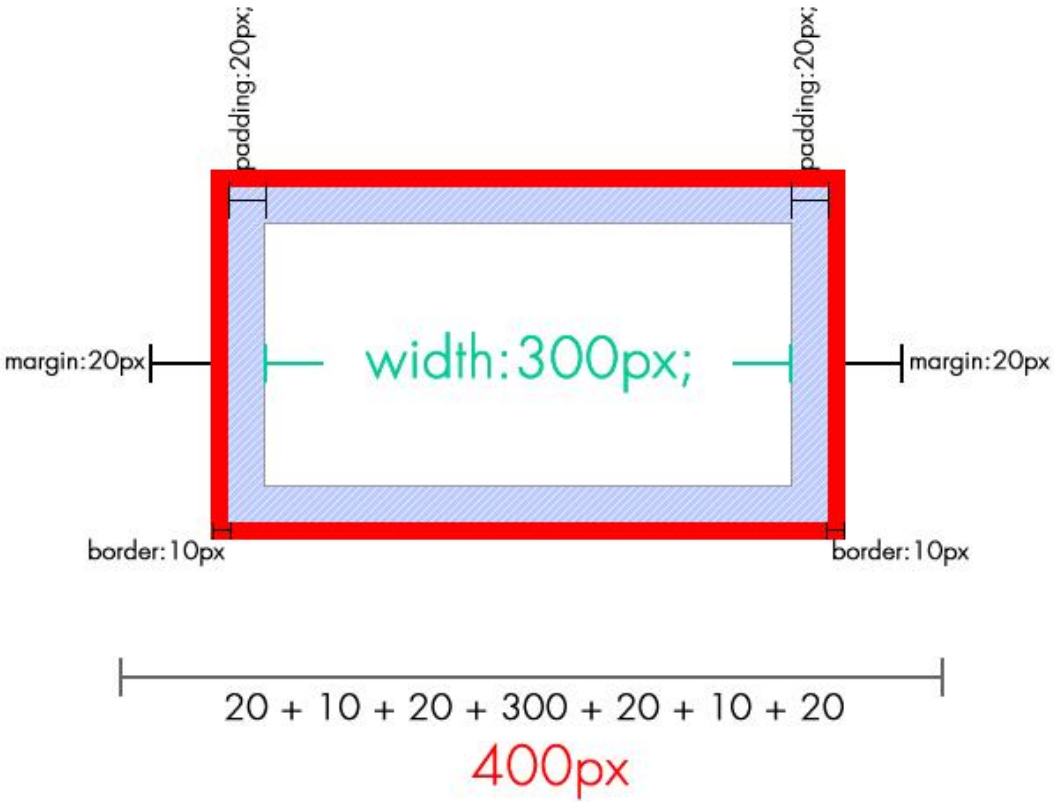
```
elemento {
    margin-top: 10px;
    margin-right: 20px;
    margin-bottom: 30px;
    margin-left: 40px;

    /* shorthand */
    margin: 10px 20px 30px 40px;
}
```

Propriedades CSS

margin




padding:20px;
padding:20px;
margin:20px
width:300px;
margin:20px
border:10px
border:10px
20 + 10 + 20 + 300 + 20 + 10 + 20
400px

Propriedades CSS

box-sizing

```
elemento {
    box-sizing: border-box;

    /* content-box (padrão) */
}
```

Definição: esta propriedade facilita o uso do modelo de caixa, pois desconta automaticamente da largura e da altura o que será somado ao preenchimento e à borda. Atenção: a margem ainda precisa ser somada.

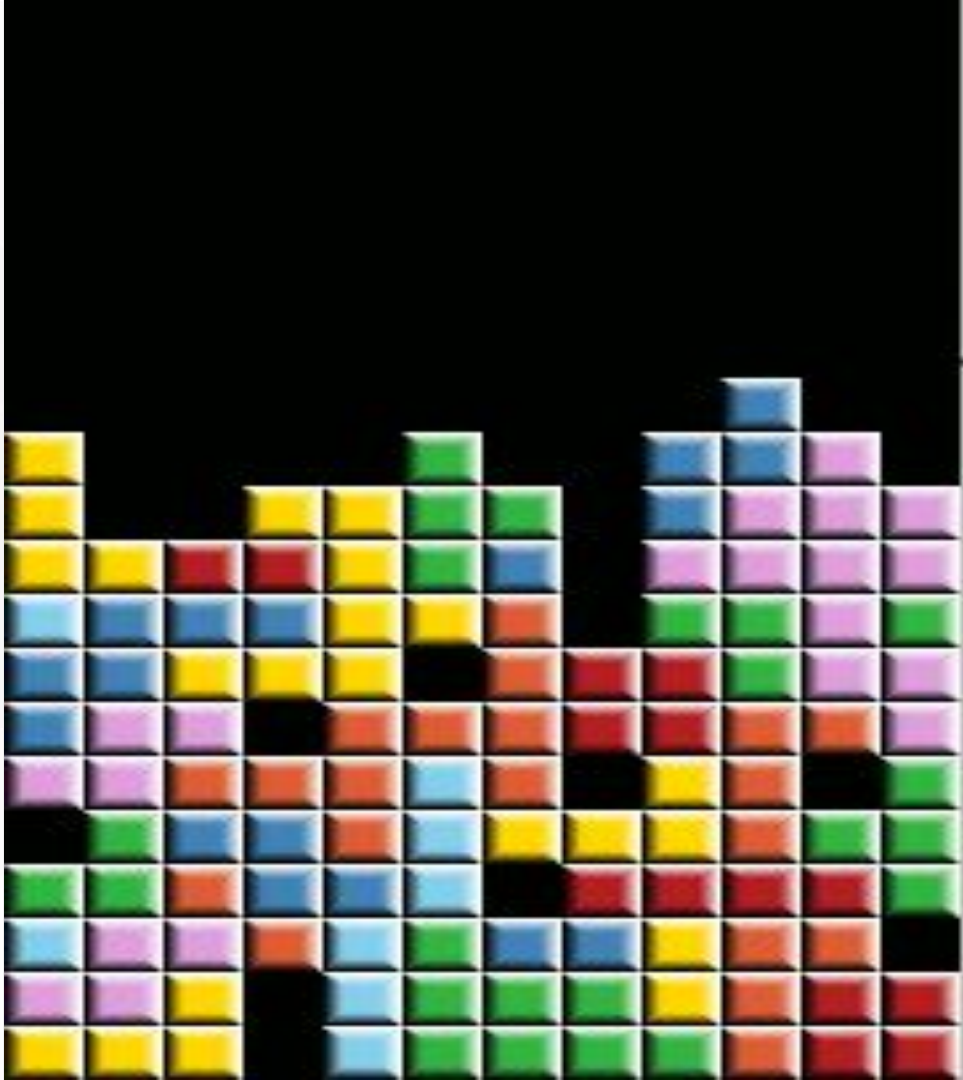
Referências

flexbox
https://origamid.com/projetos/flexbox-guia-completo/

http://www.flexboxdefense.com/

https://flexboxfroggy.com/#pt-br

Posicionamento

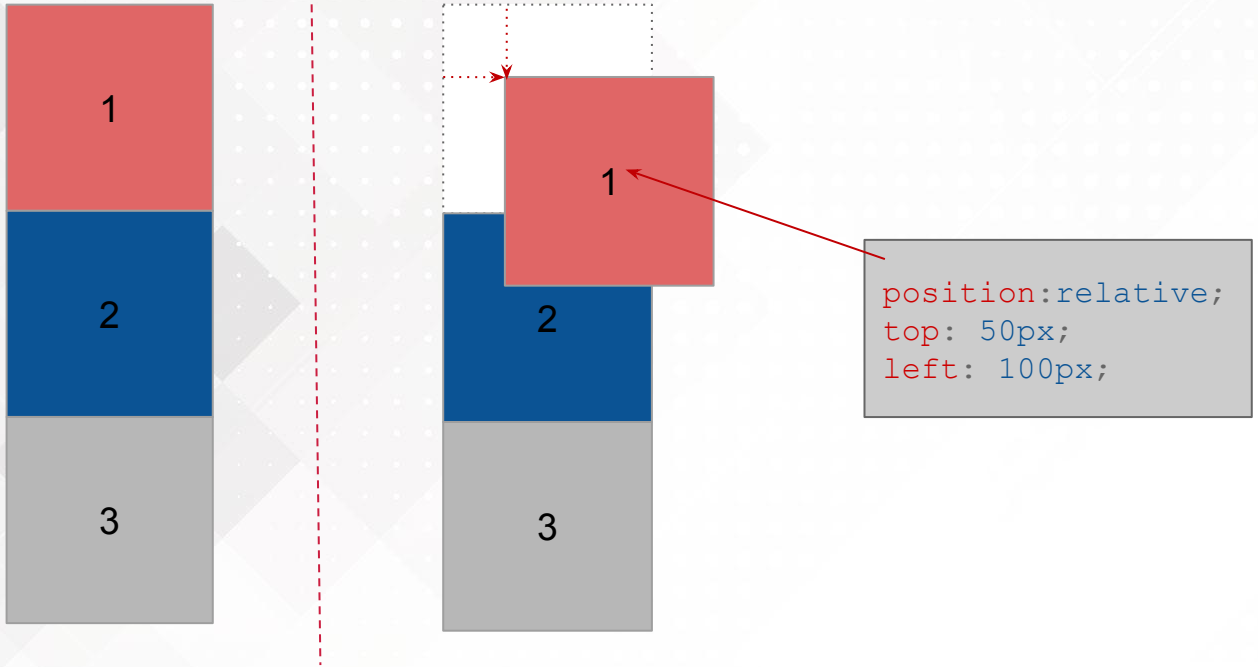


Position Relative

```
elemento {
    position: relative;
    top: 50px;
    left: 100px;
}
```

Definição: Permite mover um elemento da posição original para uma nova posição, usando como referência sua posição relativa ao elemento pai e elementos irmãos.

Position Relative



Position Absolute

```
elemento {
    position: absolute;
    right: 10px;
    bottom: 10px;
}
```

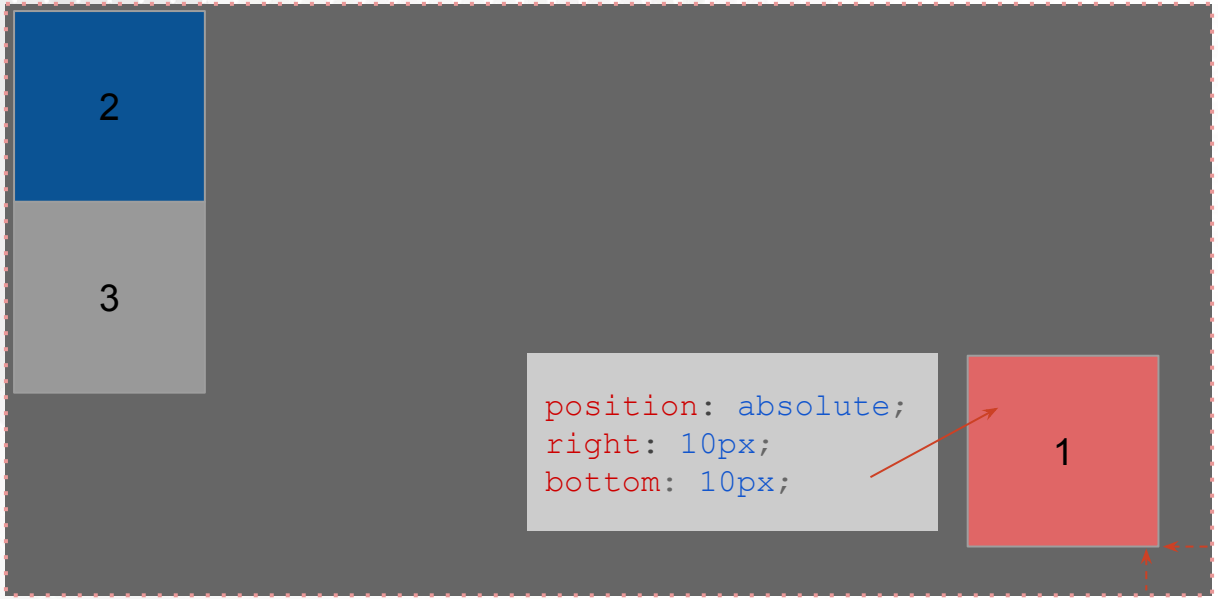
Definição: Permite mover um elemento da posição original para uma nova posição, usando como referência o **body** ou um elemento com **position: absolute;**

Position Absolute

Comportamento fora do container

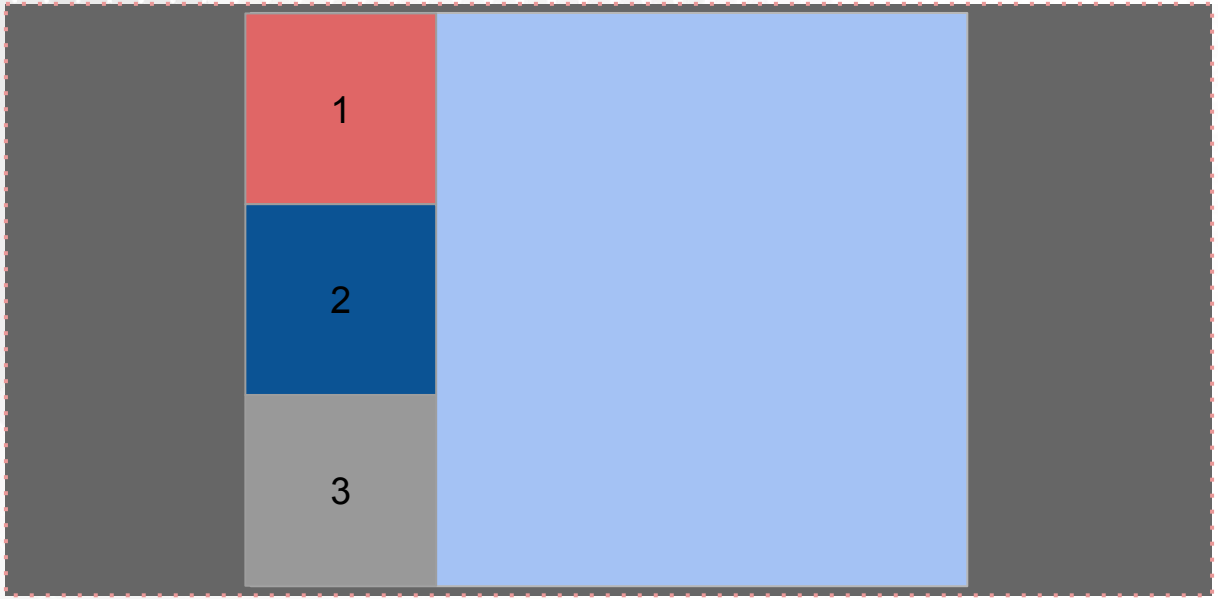
Position Absolute

Comportamento fora do container



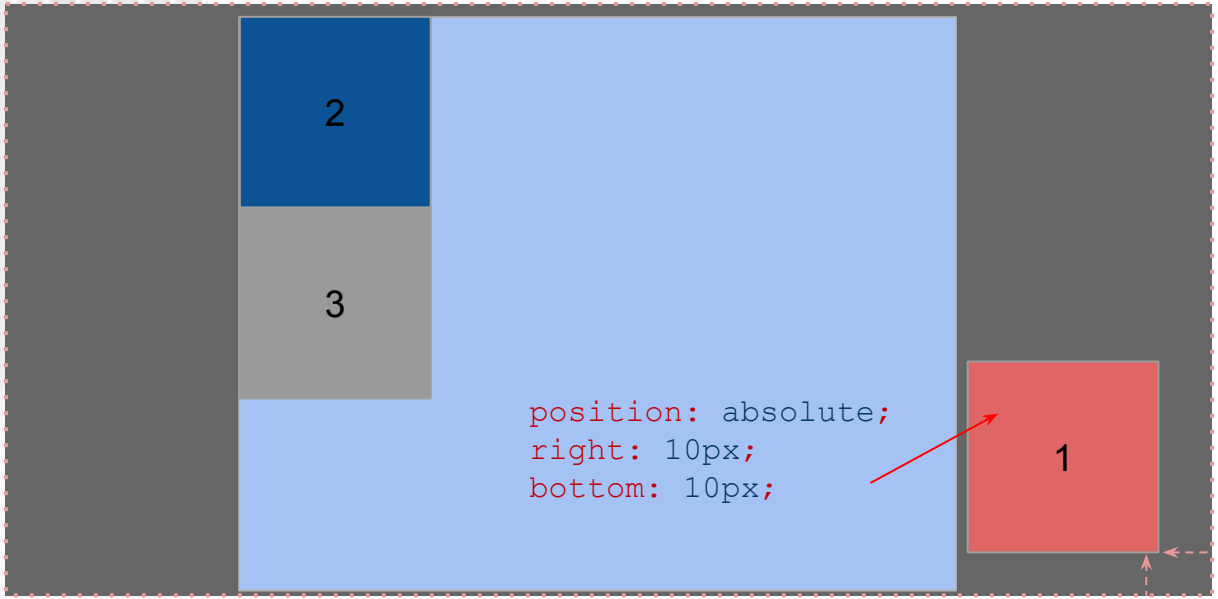
Position Absolute

Comportamento dentro do container



Position Absolute

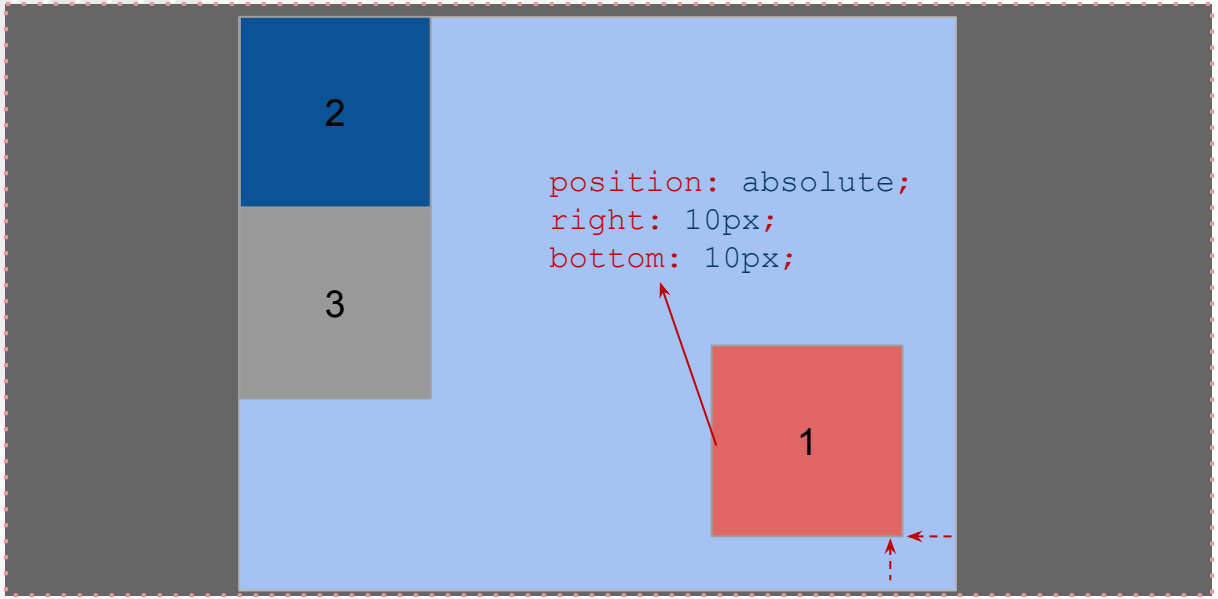
Comportamento dentro do container



Container sem **position** definido ou **position: absolute;**

Position Absolute

Comportamento dentro do container



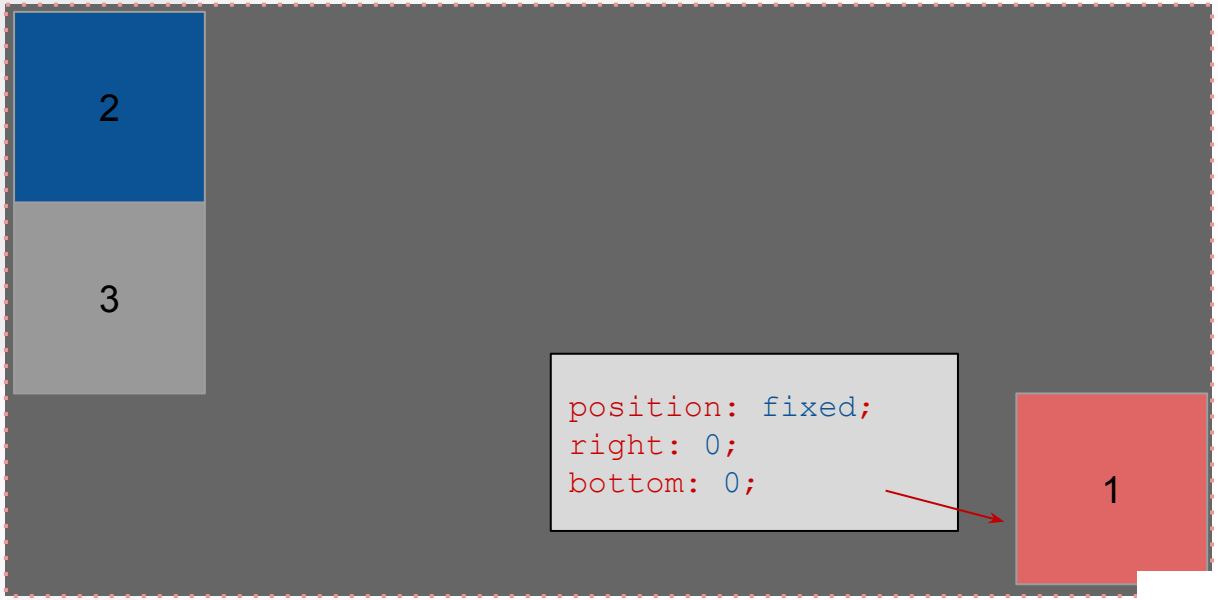
O comportamento se muda quando o container está com **position: relative;**

Position Fixed

```
elemento {
    position: fixed;
    right: 0;
    bottom: 0;
}
```

Definição: Permite mover um elemento da posição original para uma nova posição, usando **SEMPRE** como referência as **laterais da janela**. **O elemento não sai do lugar quando a página é rolada.**

Position Fixed



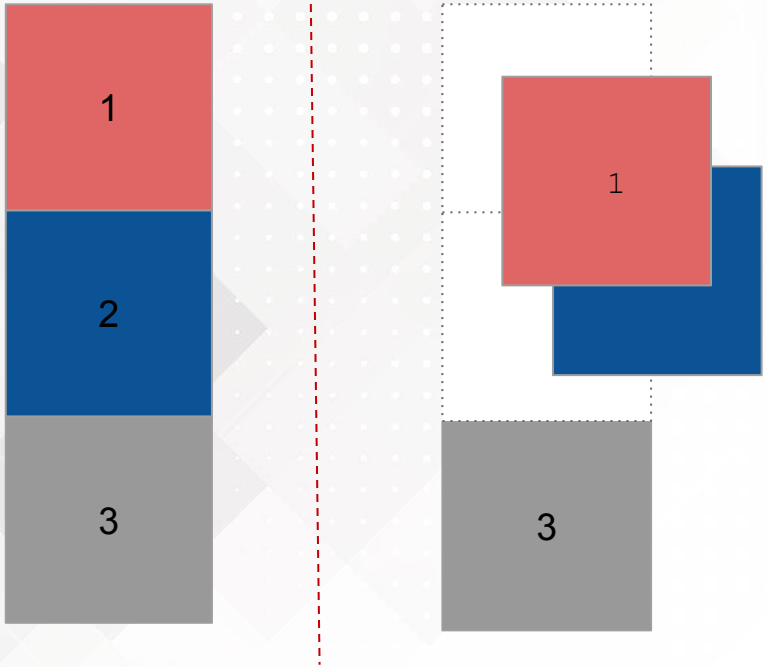
Z-index

```
elemento {
    z-index: 5;
    position: relative;
    bottom: 0;
    right: 0;
}
```

Definição: Permite mudar a ordem das “camadas” dentro de um documento HTML.

Só funciona se o elemento tiver posicionamento relativo, absoluto ou fixo.

Z-index



Referências

Position

http://www.escolaw3.com/tutoriais/css/css-position

https://www.devmedia.com.br/como-usar-a-propriedade-position-css/24451

Z-index

https://www.devmedia.com.br/css-z-index-entendendo-sobre-o-eixo-z-na-web/28698

https://css-tricks.com/almanac/properties/z/z-index/

Flexbox

display: flex

```
elemento {
    display: flex;
}
```

Definição: Torna o elemento um flex container automaticamente transformando todos os seus filhos diretos em flex itens.

Flexbox

flex-direction

```
elemento {
    display: flex;
    flex-direction: column;
}

[row, row-reverse, column-reverse]
```

Definição: Define a direção dos flex itens. Por padrão ele é row (linha), por isso quando o display: flex; é adicionado, os elementos ficam em linha, um do lado do outro.

Flexbox

justify-content

```
elemento {
    display: flex;
    justify-content: flex-start;
}

[flex-end, center, space-between, space-around]
```

Definição: Alinha os itens flex no container de acordo com a direção. A propriedade só funciona se os itens atuais não ocuparem todo o container. Isso significa que ao definir flex: 1; ou algo similar nos itens, a propriedade não terá mais função.

Flexbox

flex-wrap

```
elemento {
    display: flex;
    flex-wrap: wrap;
}

[nowrap, wrap-reverse]
```

Definição: Define se os itens devem quebrar ou não a linha. Por padrão eles não quebram linha, isso faz com que os flex itens sejam compactados além do limite do conteúdo.

Flexbox

align-items

```
elemento {
    display: flex;
    align-items: center;
}

[flex-end, flex-start, stretch]
```

Definição: O align-items alinha os flex itens de acordo com o eixo do container. O alinhamento é diferente para quando os itens estão em colunas ou linhas

Flexbox

align-content

```
elemento {
    display: flex;
    align-content: center;
}

[flex-end, flex-start, stretch]
```

Definição: Alinha as linhas do container em relação ao eixo vertical. A propriedade só funciona se existir mais de uma linha de flex-itens. Para isso o flex-wrap precisa ser wrap.

Flexbox

align-self

```
elementoPai {
  display: flex;
  align-items: flex-start;
}
elementoFilho {
    align-self: flex-end;
}

[flex-end, flex-start, stretch]
```

Definição: O align-self aceita os mesmos valores que a propriedade align-items mas alinha somente o item selecionado.

Referências

flexbox
https://origamid.com/projetos/flexbox-guia-completo/

http://www.flexboxdefense.com/

https://flexboxfroggy.com/#pt-br

Pseudo-elementos

Permitem inserir conteúdo com CSS.
São usados sobre um seletor pré-definido.
Podemos inserir conteúdo de texto, imagens ou elementos de bloco.

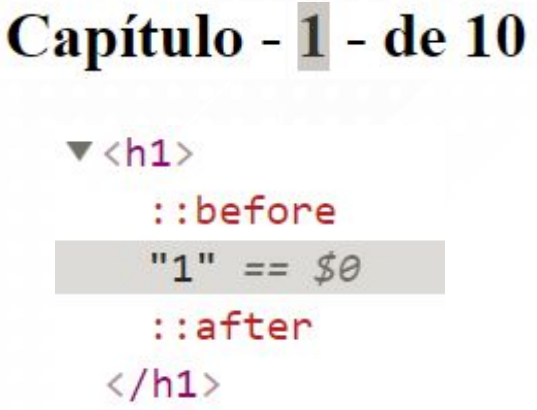
Saiba mais:
https://www.w3schools.com/css/css_pseudo_elements.asp
https://www.w3schools.com/css/css_pseudo_classes.asp
https://www.w3schools.com/cssref/sel_nth-child.asp

Pseudo-elementos

::before & ::after

```
h1::before {
    content: "Capítulo - ";
}

h1::after {
    content: " - de 10";
}
```



Pseudo-elementos

::before & ::after

```
.caixa::before {
    content: url(../images/icone-1.gif);
}

.caixa::after {
    content: url(../images/icone-2.gif);
}
```

Obrigada!