

The background is a dark blue, semi-transparent image of an industrial factory floor. It features various pieces of machinery, conveyor belts, and structural elements. Overlaid on this background are several circular gauges or speedometers. One gauge on the right shows '79%', another above it shows '80%', and a larger one on the right shows '94%'. There are also some numerical labels like '01 - 1000' near the gauges.

# UnISENAI

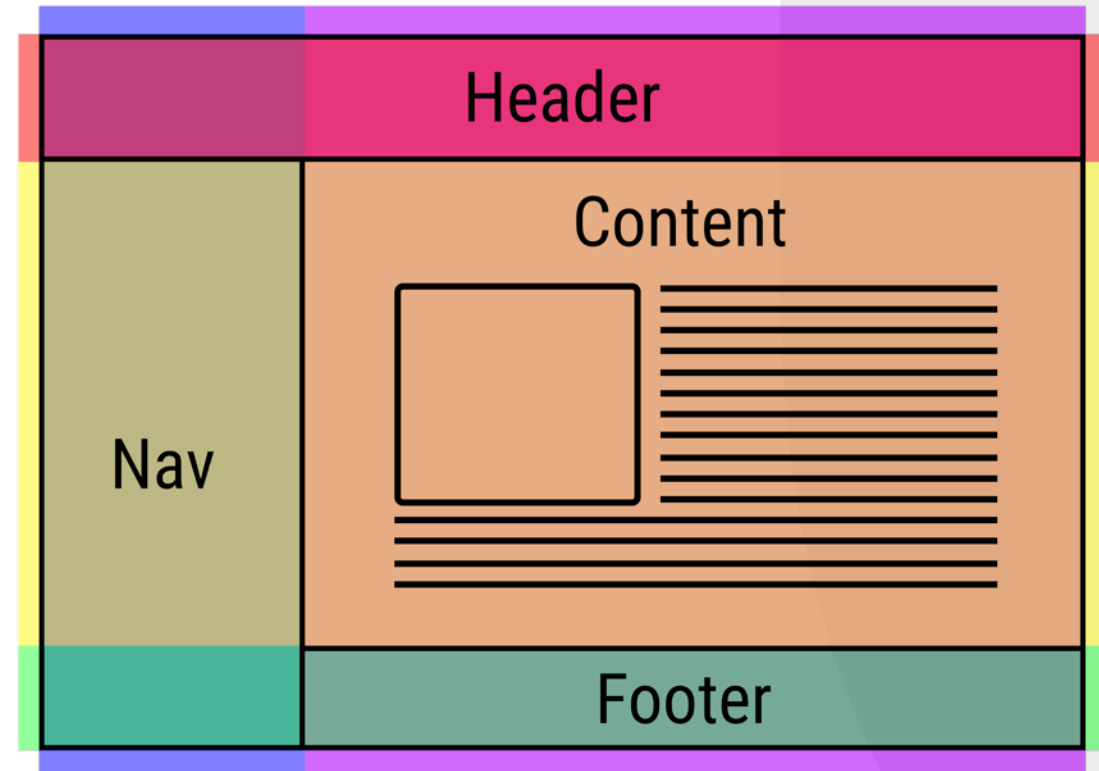
O FUTURO COMEÇA  
POR VOCÊ!

# Layouts com CSS

Desenvolvimento de Site

**UniSENAI**

# Blocos



# 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**

Oculto 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				

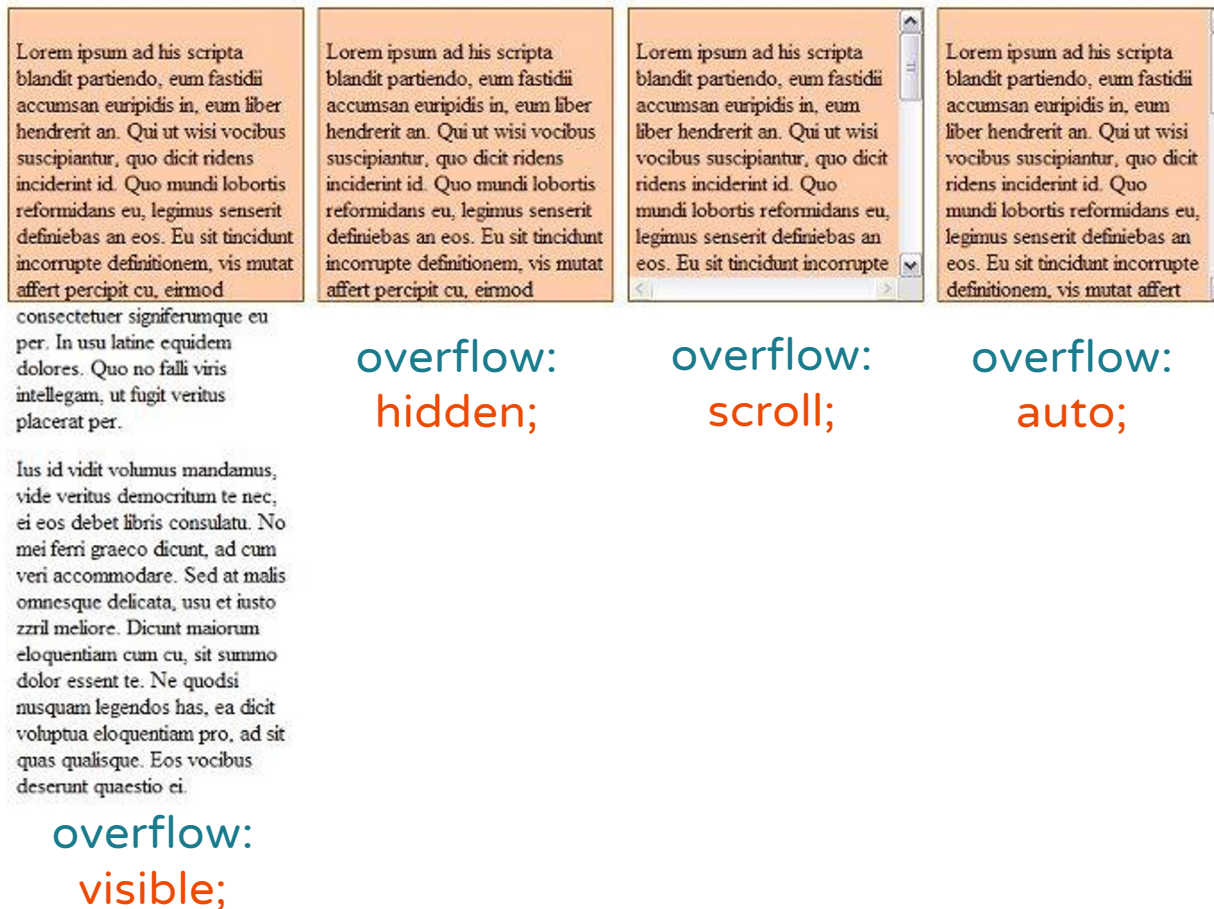
# Vamos testar!





# 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.

# Propriedades CSS

width/height

width

**elemento**

display:block;

ou

display:inline-block;

height

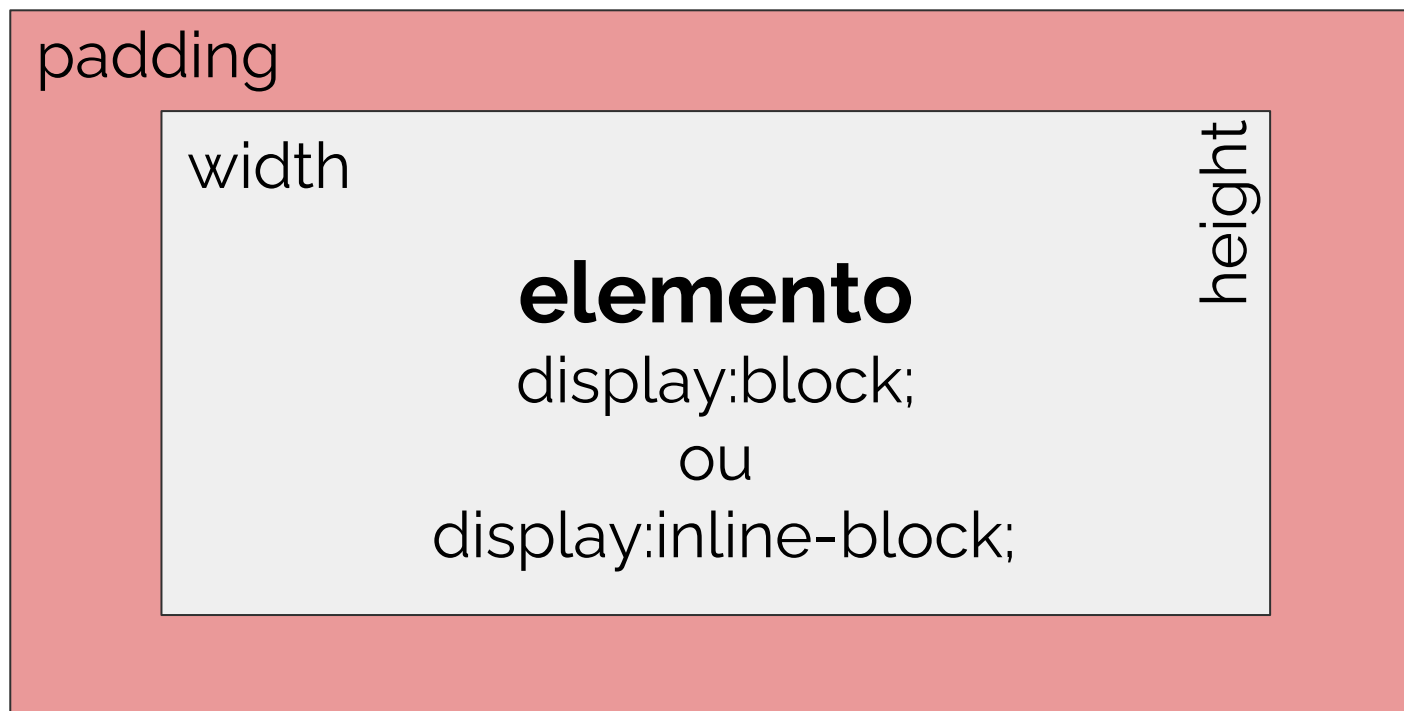
# Propriedades CSS

## 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

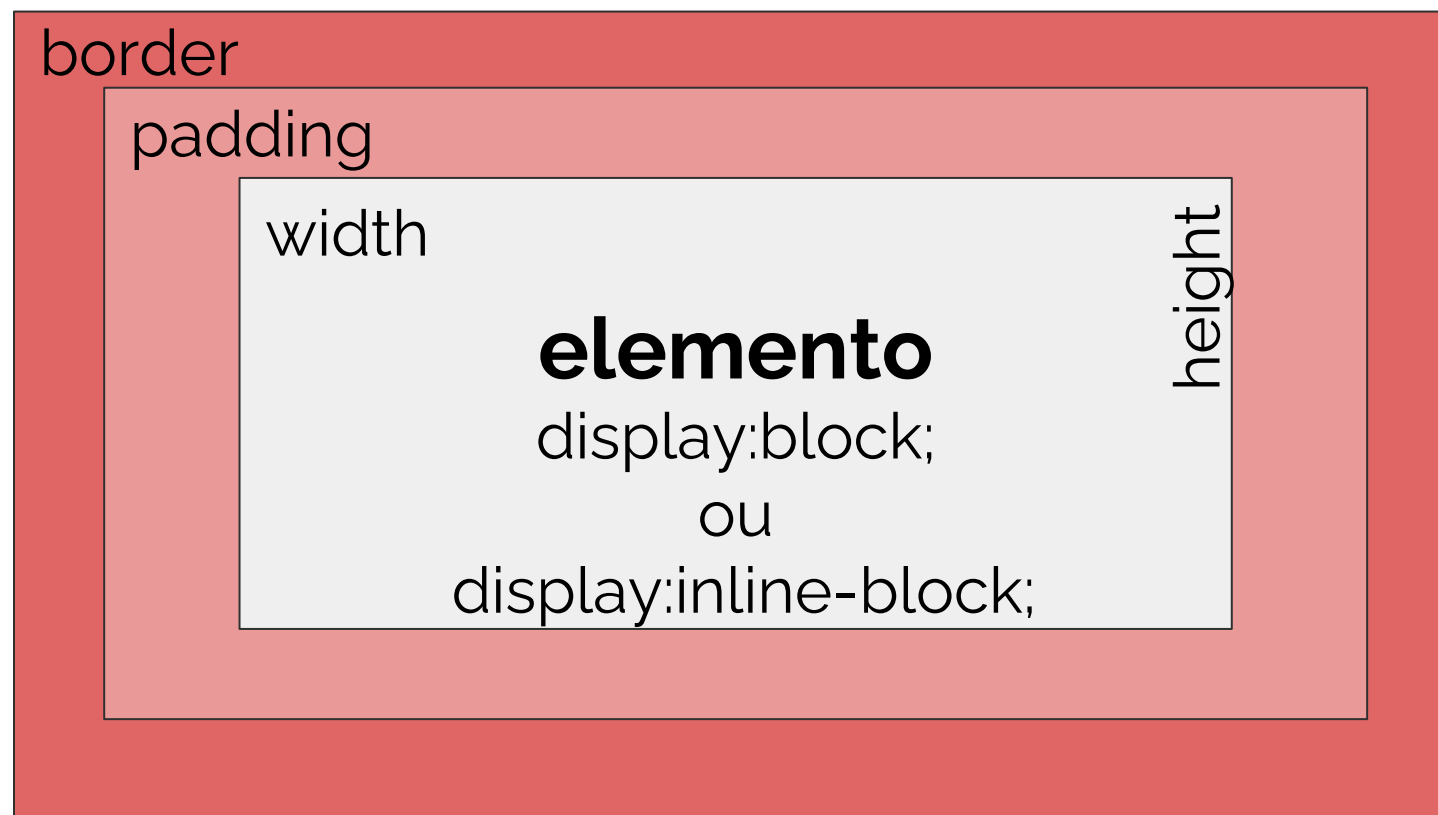
## border

```
elemento {  
    border-style: solid;  
    border-width: 2px;  
    border-color: #FF0000;  
  
    /* shorthand */  
    border: solid 2px #FF0000;  
}
```



# Propriedades CSS

## border



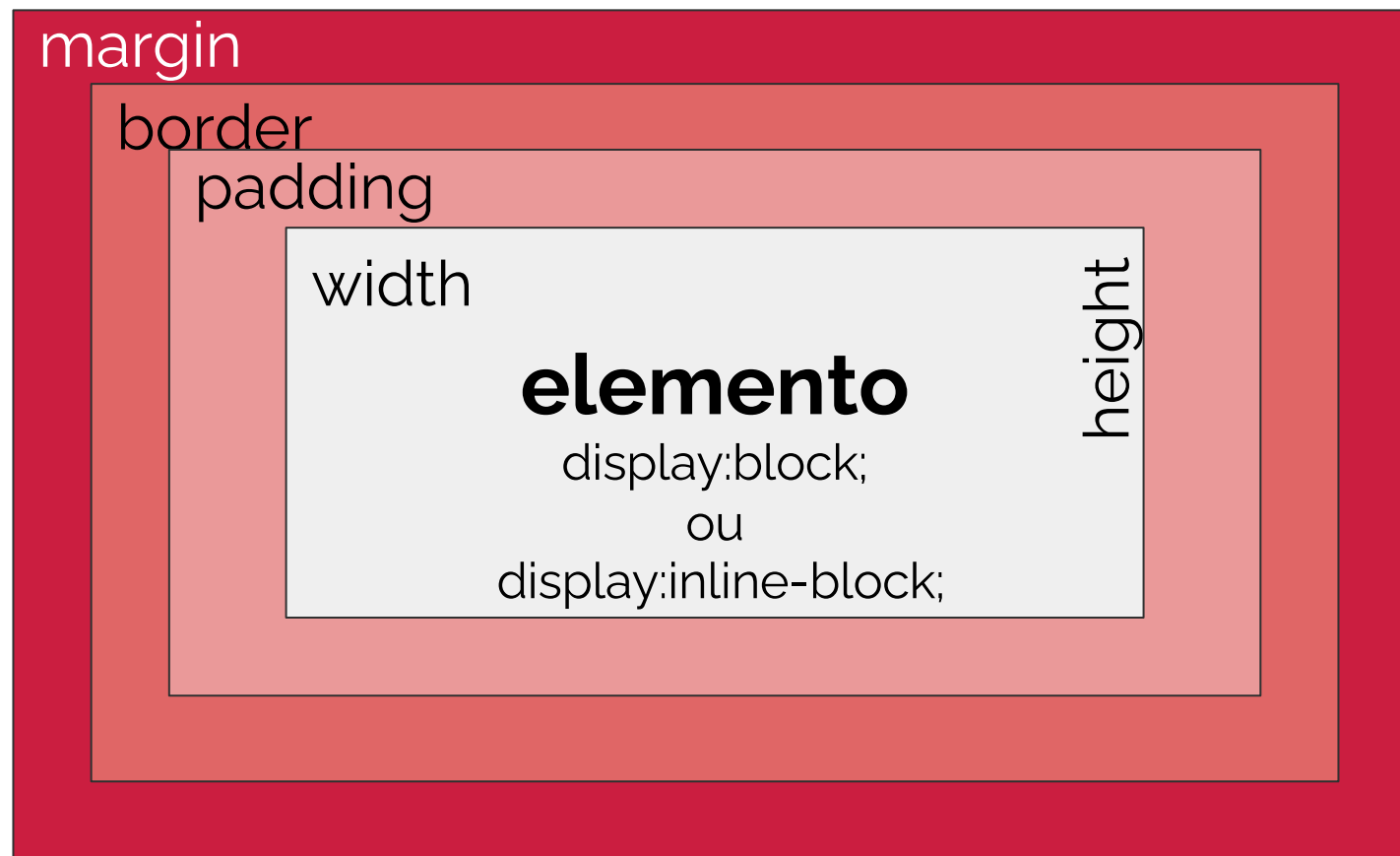
# Propriedades CSS

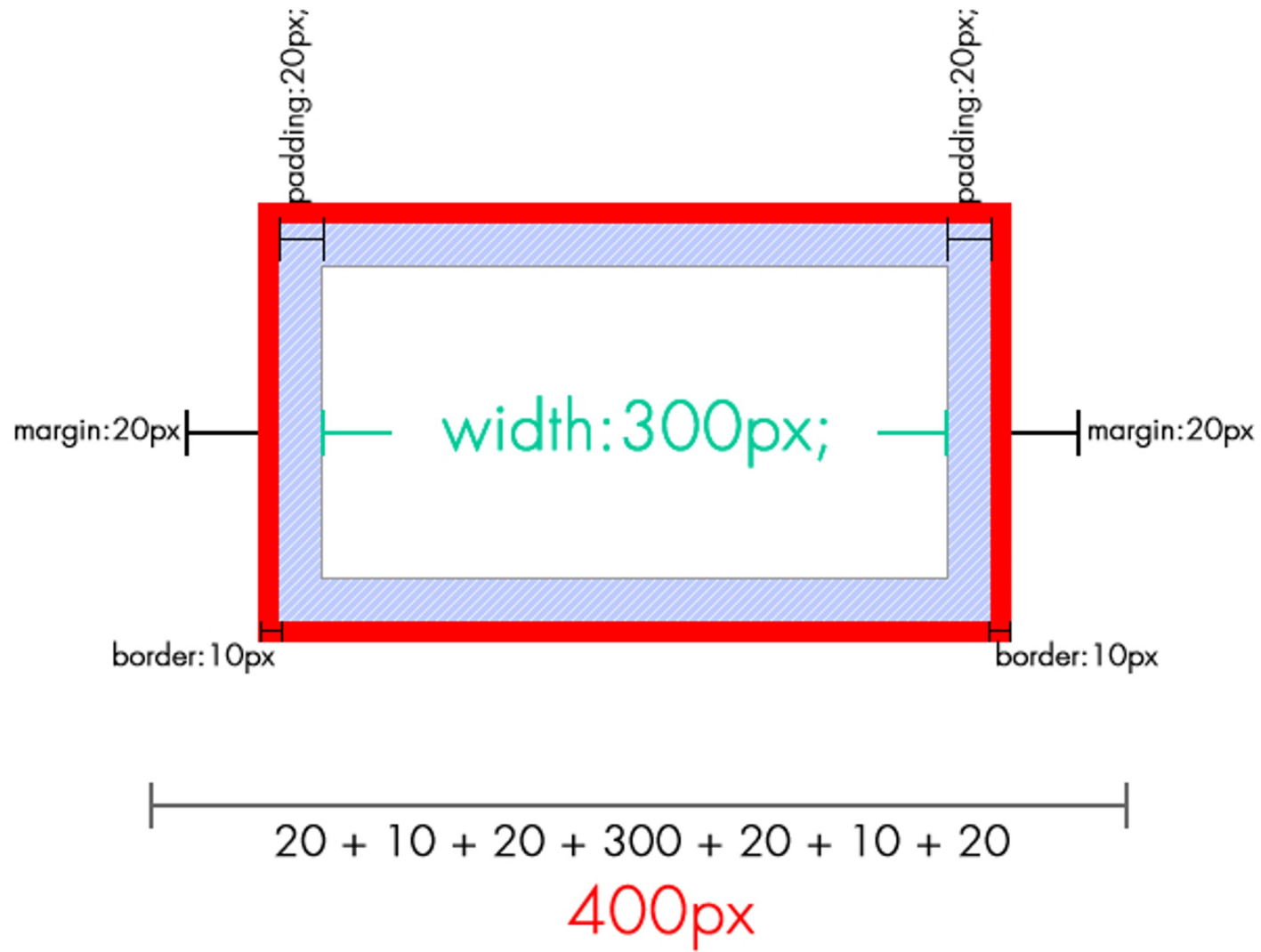
## margin

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

# Propriedades CSS

## margin





# 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.

# Propriedades CSS

## imagens de fundo

O CSS define 6 propriedades para estabelecer o fundo de qualquer elemento

**background-image:** `url(.. /images/fundo.jpg) ;`

**background-repeat:** `repeat ;`  
`[no-repeat | repeat-x | repeat-y]`

**background-position:** `X Y ;`  
`[X = left | center | right | length(px,%...)]`  
`[Y = top | center | bottom | length(px,%...)]`



# Propriedades CSS

## imagens de fundo

**background-attachment:** scroll;  
[fixed | local]

**background-color:** #F2F2F2;

**background-size:** cover;  
[contain | length]

# Propriedades CSS

## cores

Pode ser aplicado a qualquer propriedade que receba cor:

Hexadecimal

#F05331

RGB

rgb (240 , 83 , 49)

RGBA

rgba (240 , 83 , 49 , 0.5)

# Propriedades CSS

## opacity

Deixa todo o elemento transparente, não só o fundo

**opacity: 0.5;**

Valores em frações:

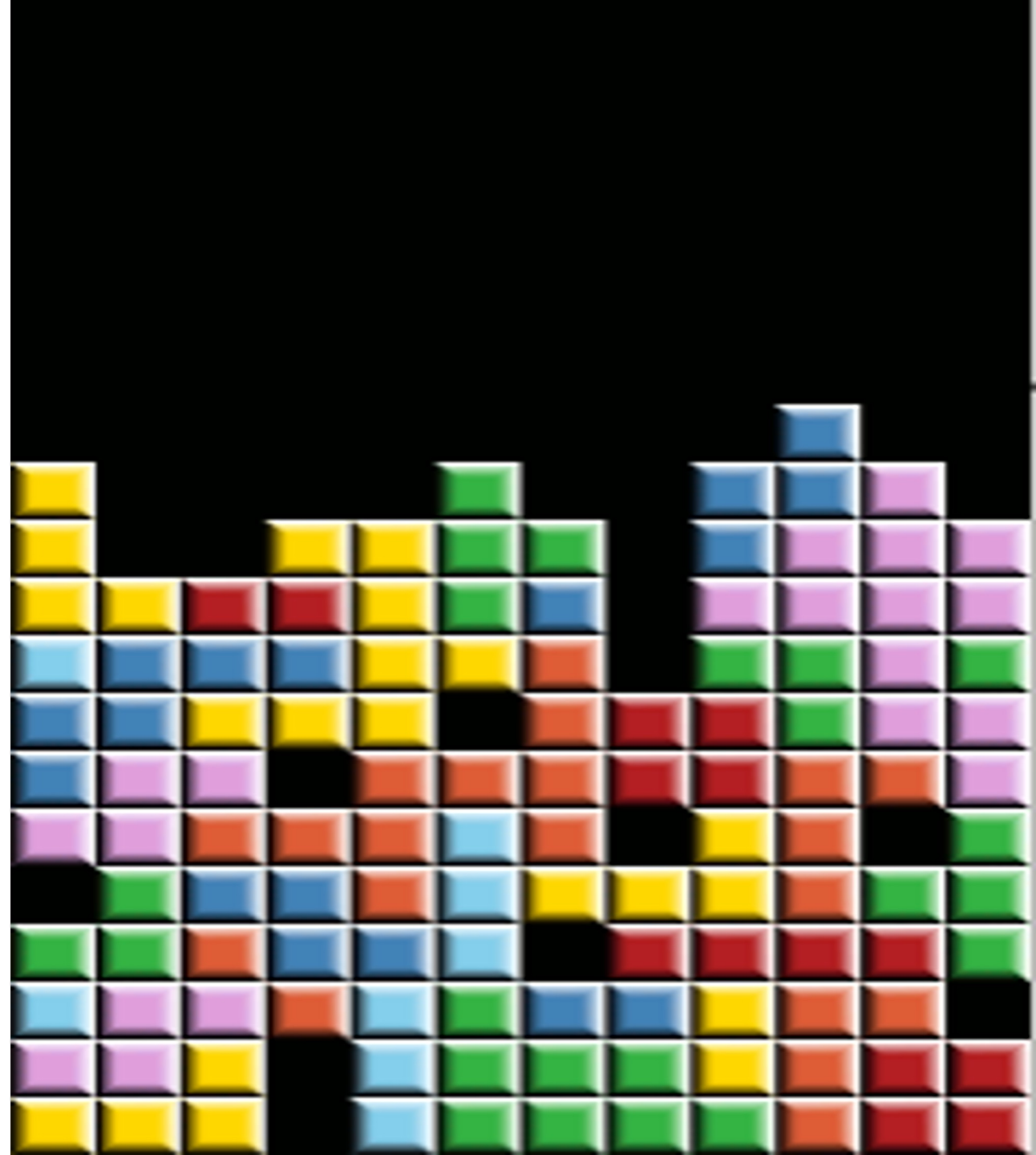
1.0 = 100%

0.5 = 50%

0.2 = 20%

0.05 = 5%

# Posicionamento

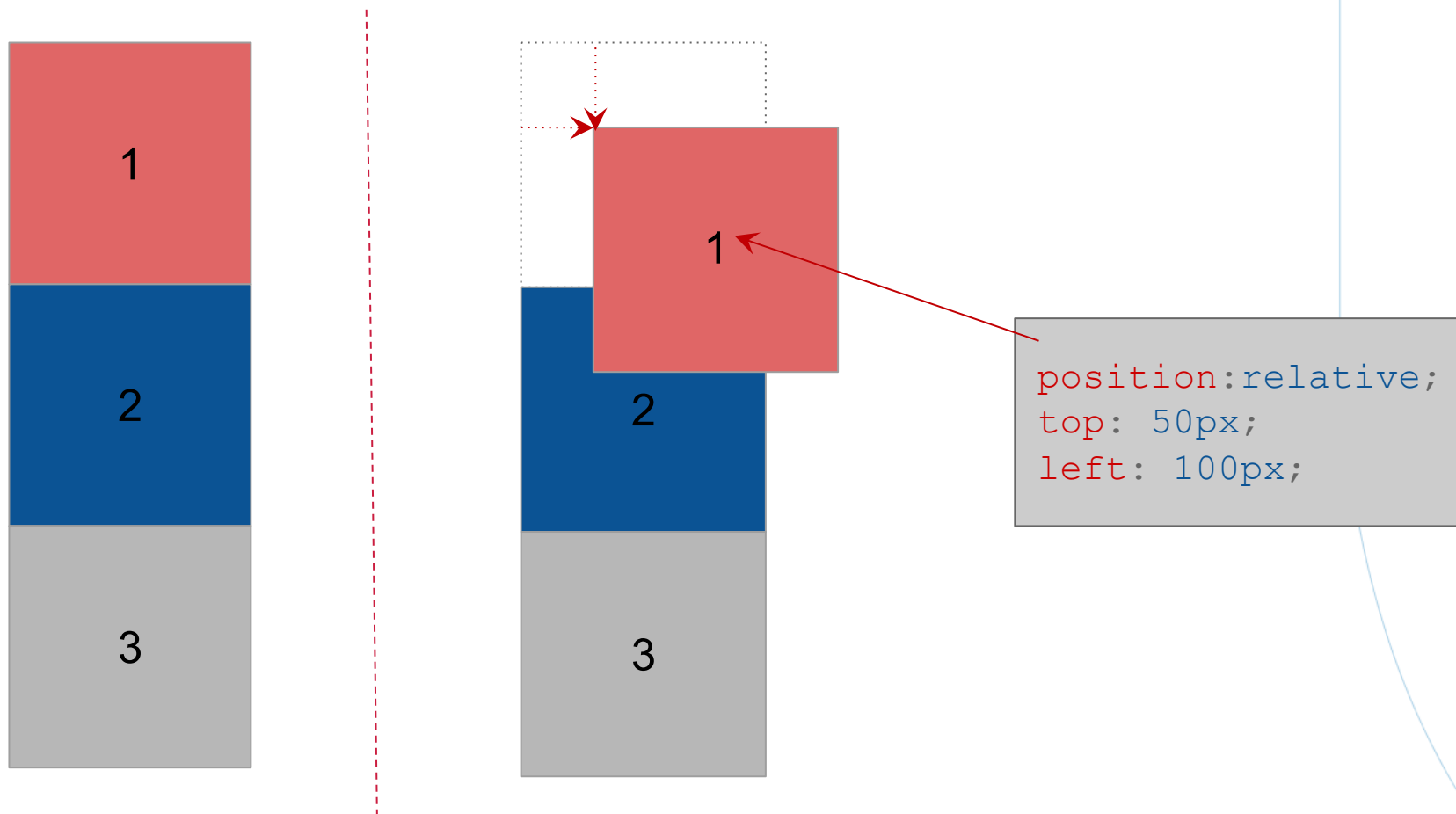


# Position Relative

```
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;**

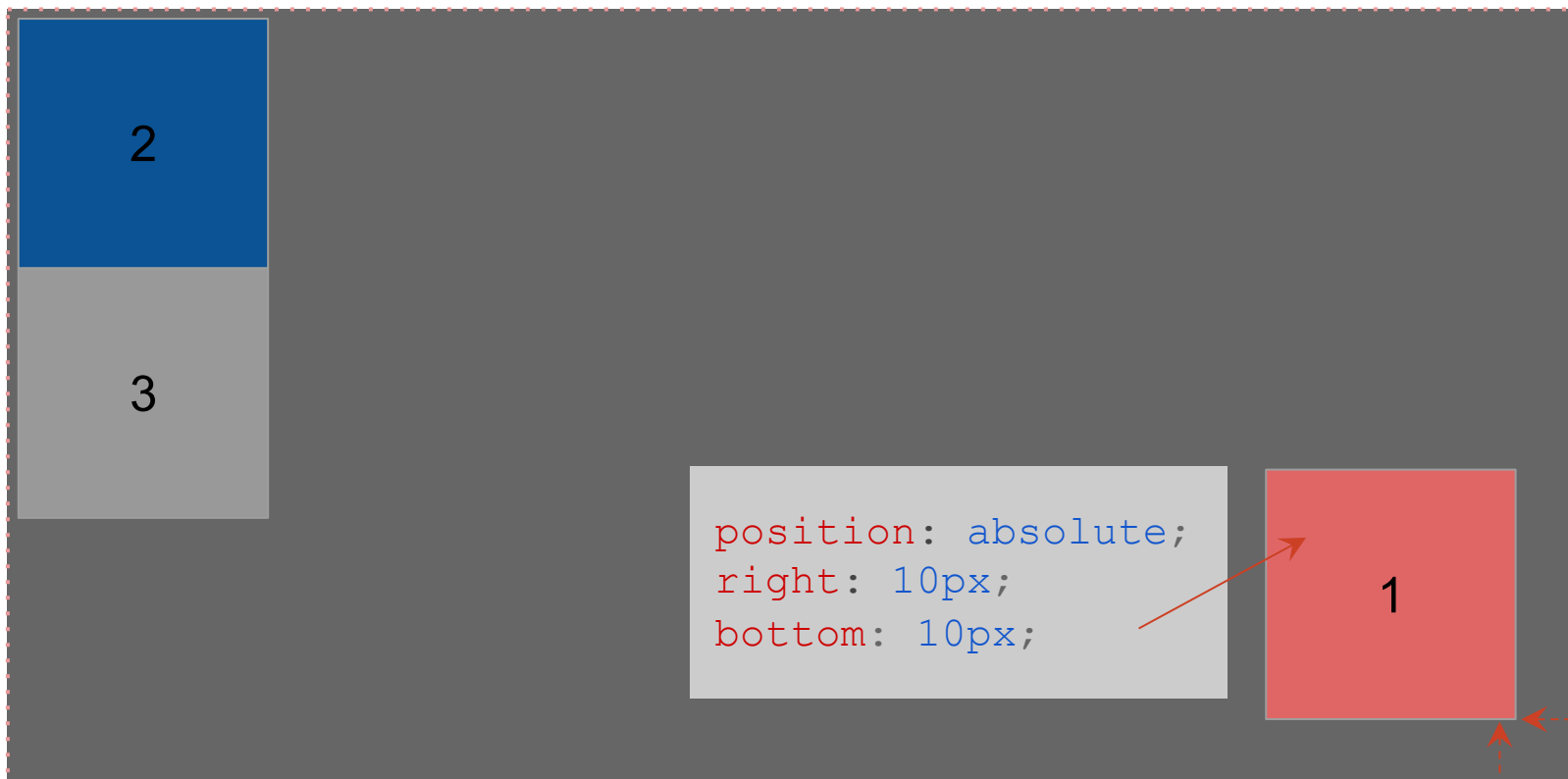
# Position Absolute

Comportamento fora do container



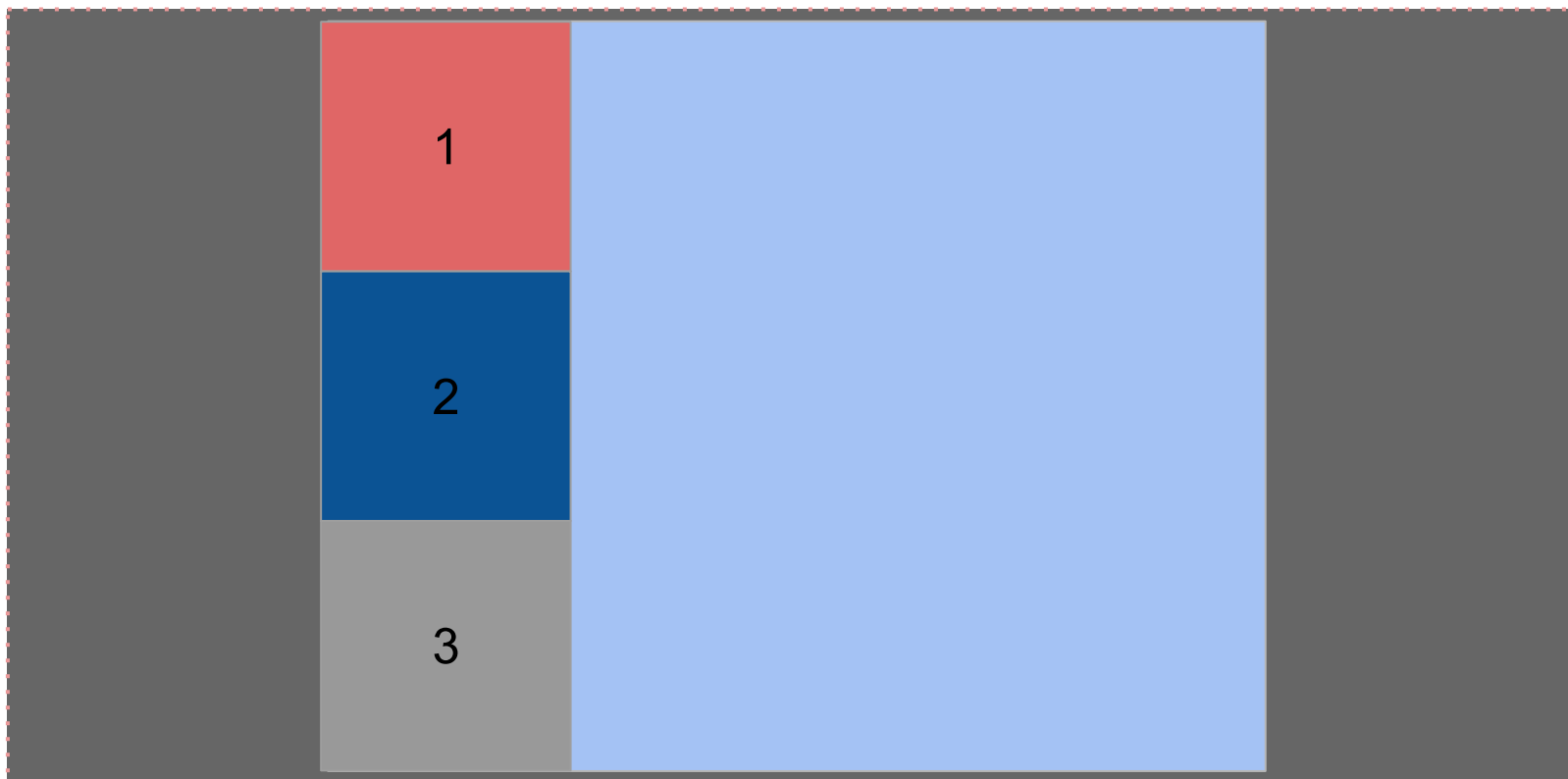
# Position Absolute

Comportamento fora do container



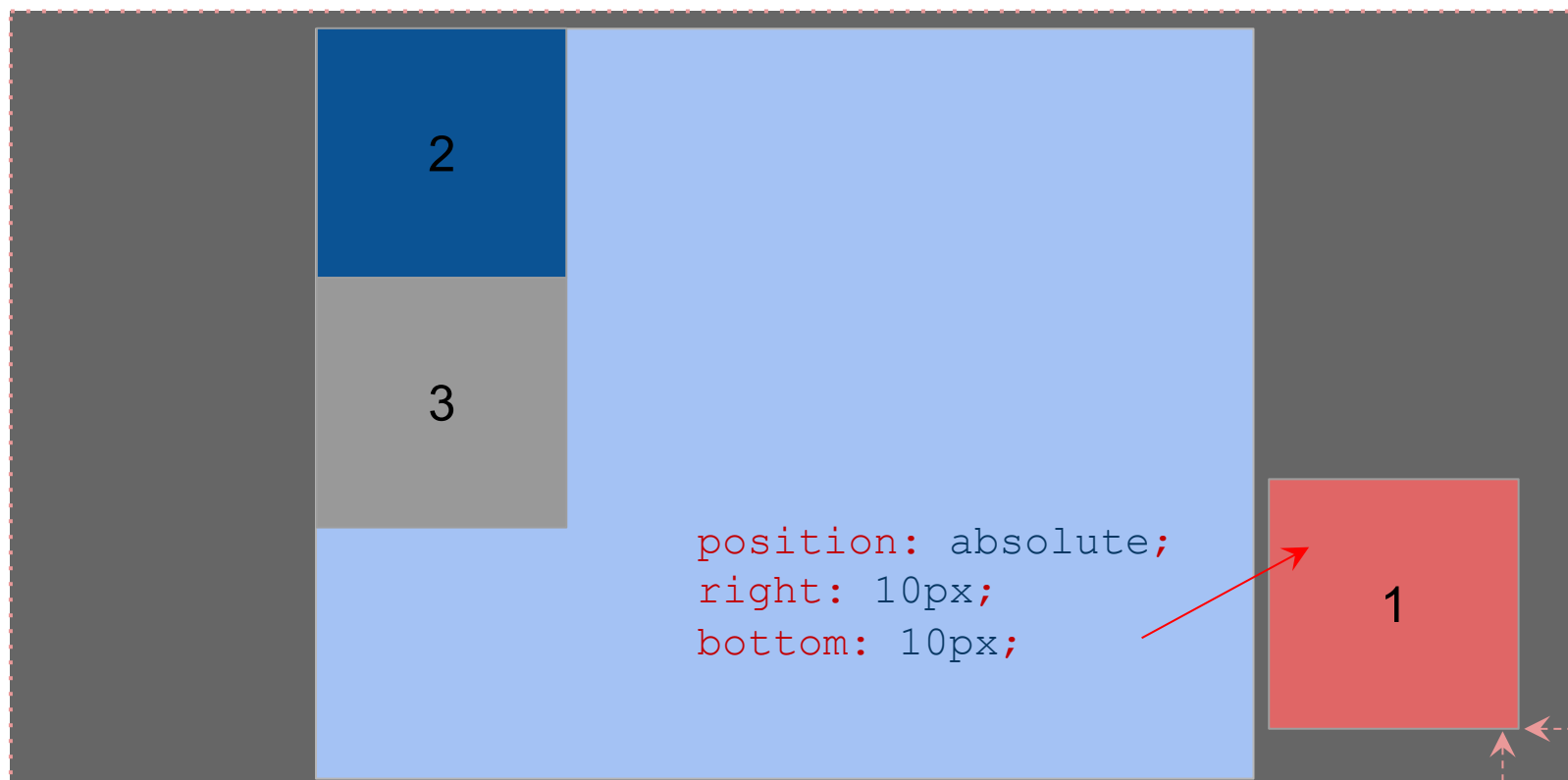
# Position Absolute

Comportamento dentro do container



# Position Absolute

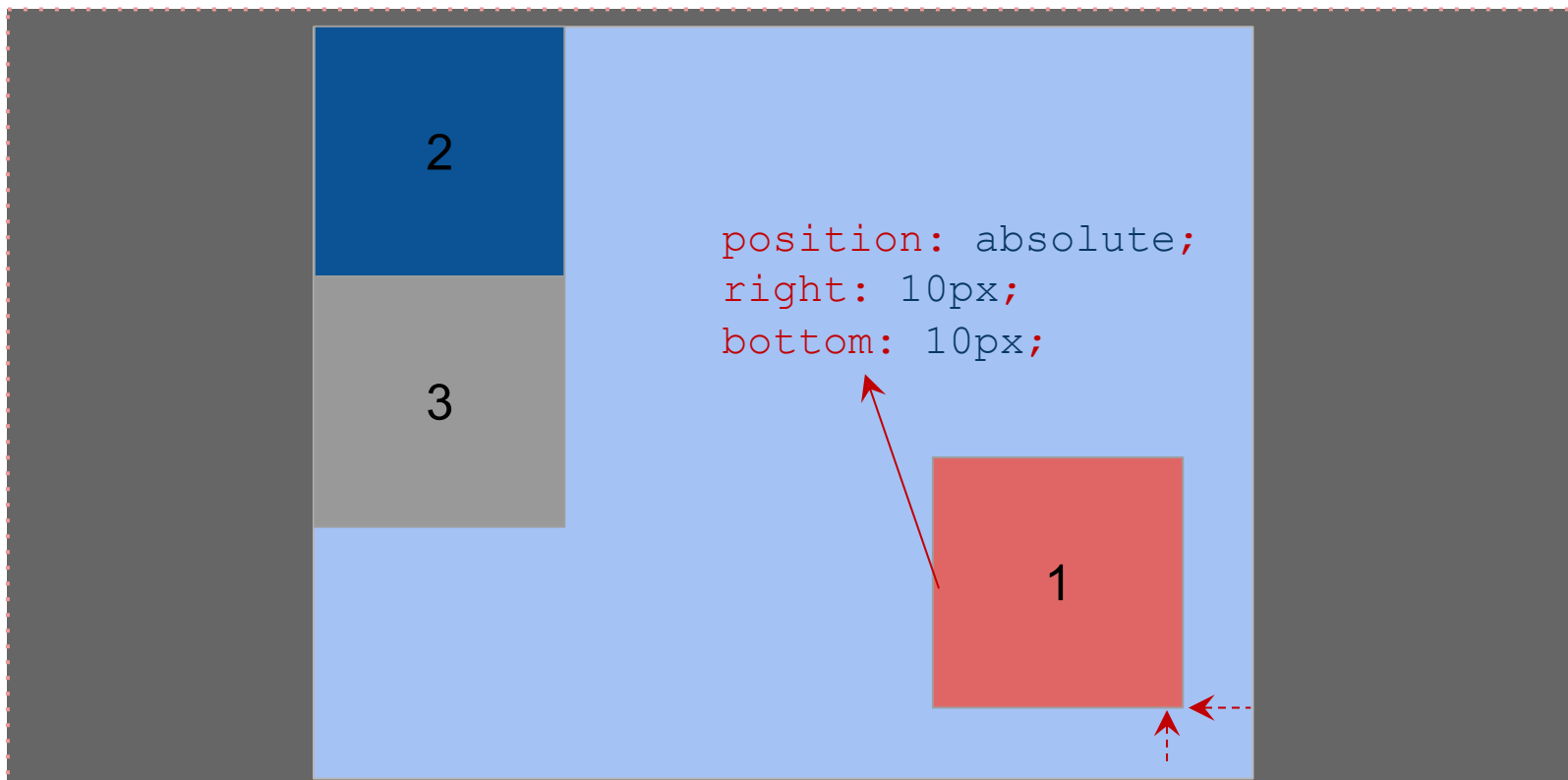
Comportamento dentro do container



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

# Position Absolute

Comportamento dentro do container



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

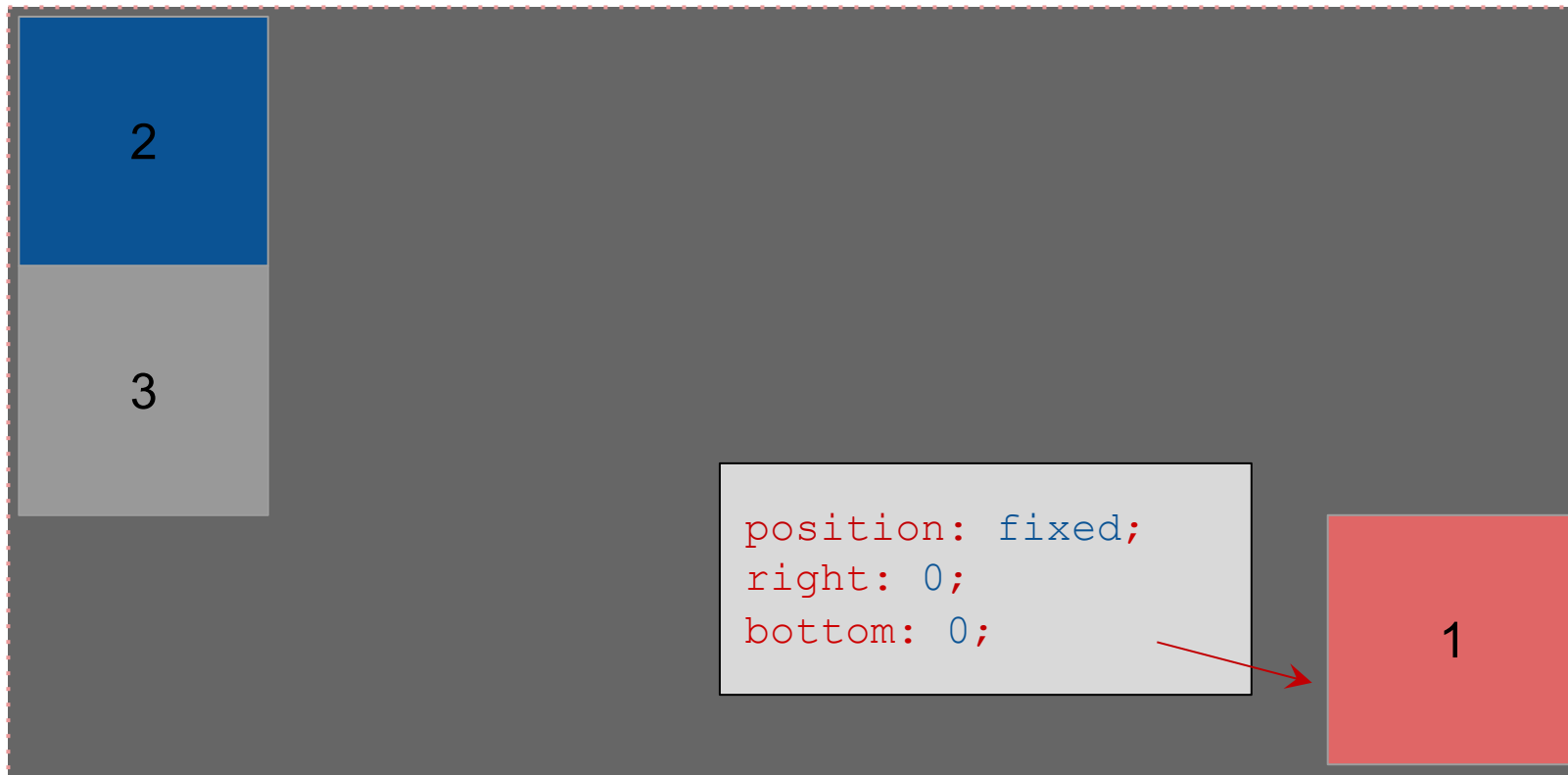


# Position Fixed

```
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

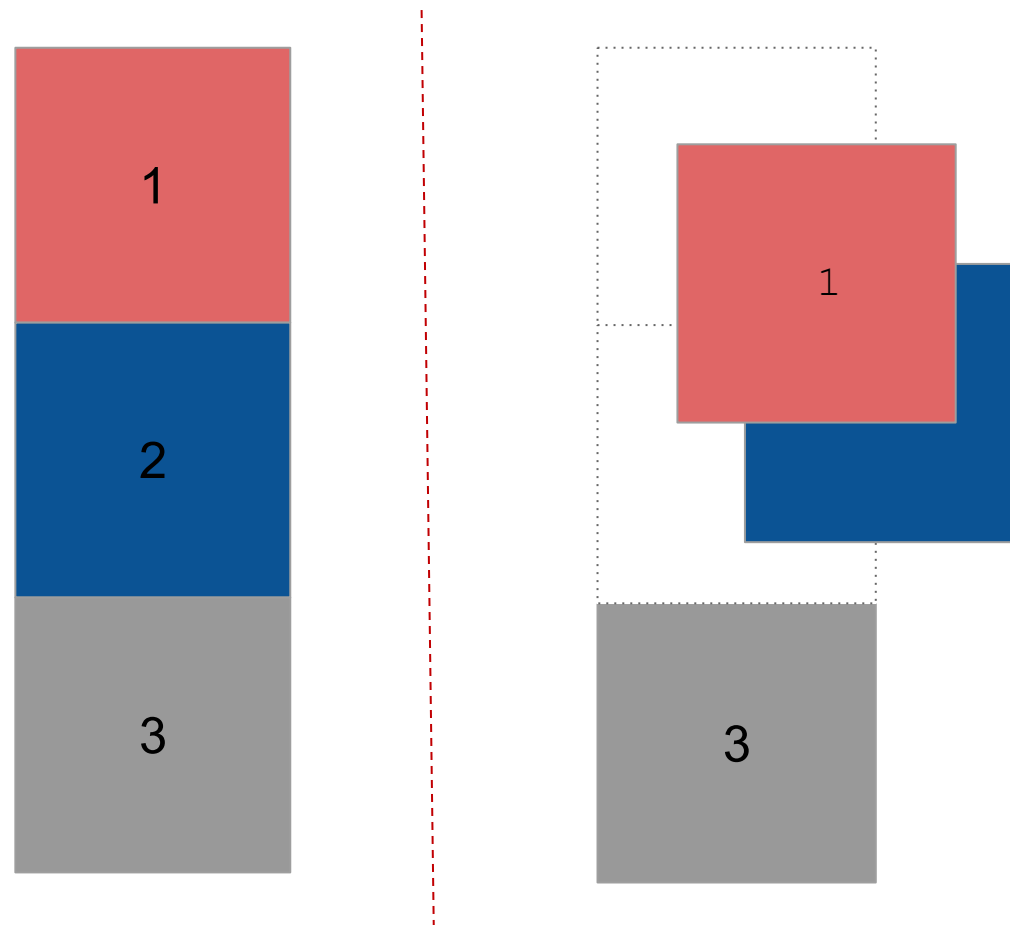


# Z-Index

```
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



# Flexbox

**flex-start**



**flex-end**



**center**



**space-between**



**space-around**



**space-evenly**



# 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

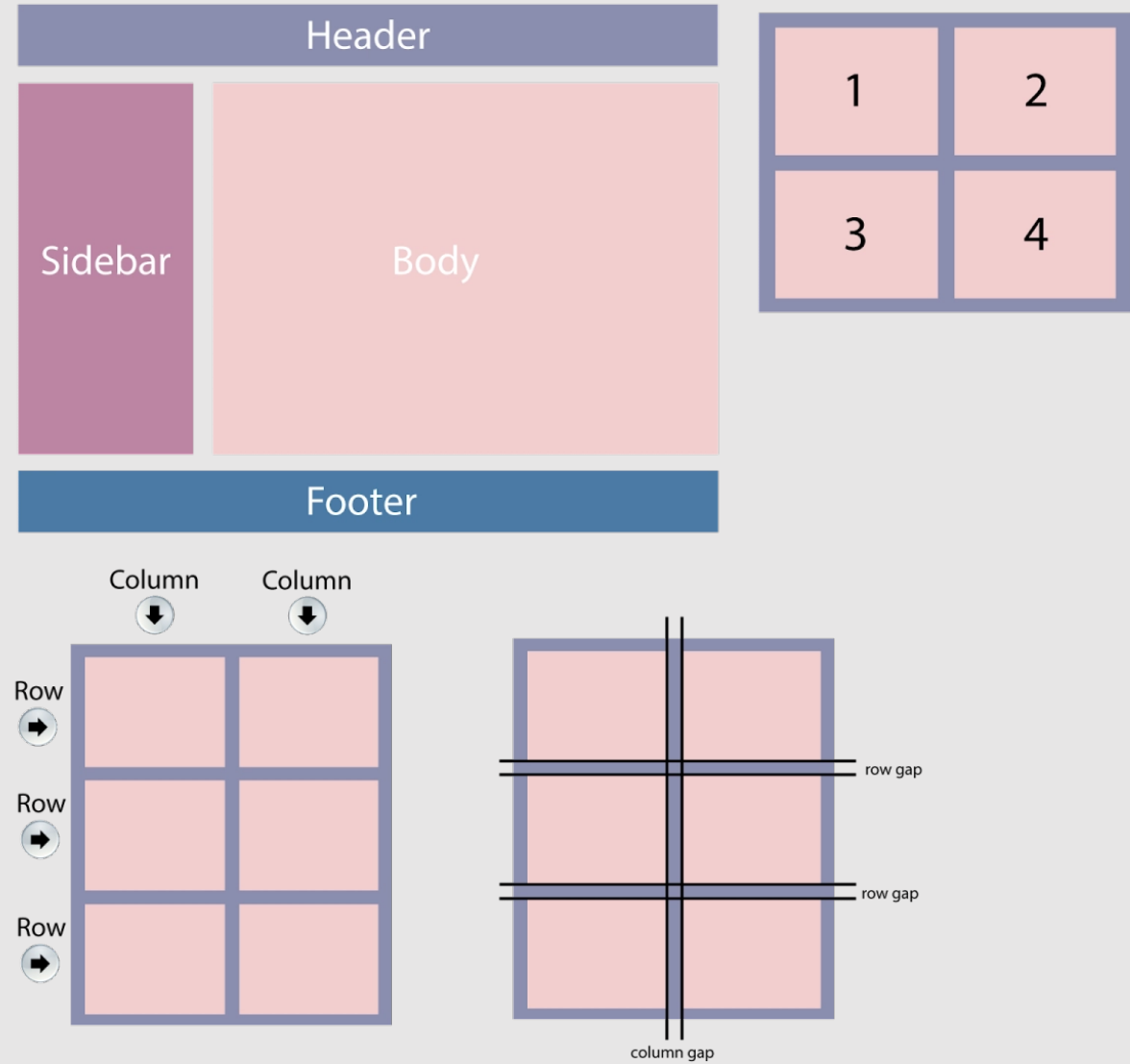
Para entender mais sobre Flexbox, veja os conteúdos abaixo:

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

<http://www.flexboxdefense.com/>

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

# Grid



# Grid

display: grid

```
elemento {  
    display: grid;  
}
```

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

# Referências Grid

Para entender mais sobre Grid, veja o conteúdo abaixo:

<https://www.origamid.com/projetos/css-grid-layout-guia-completo/>




# Let's Fun



# UnISENAI

SC-401, 3730, Saco Grande, Florianópolis - SC

 (48) 3239-5745

[unisenaisc.com.br](http://unisenaisc.com.br)



**SENAI**