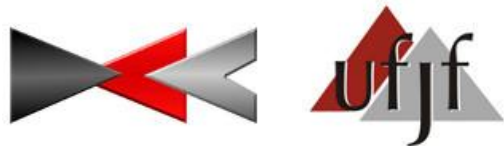


---

# Folhas de Estilo em Cascata: Modelo de Caixa, Flexbox e Grid

UFJF - DCC202 - Desenvolvimento Web

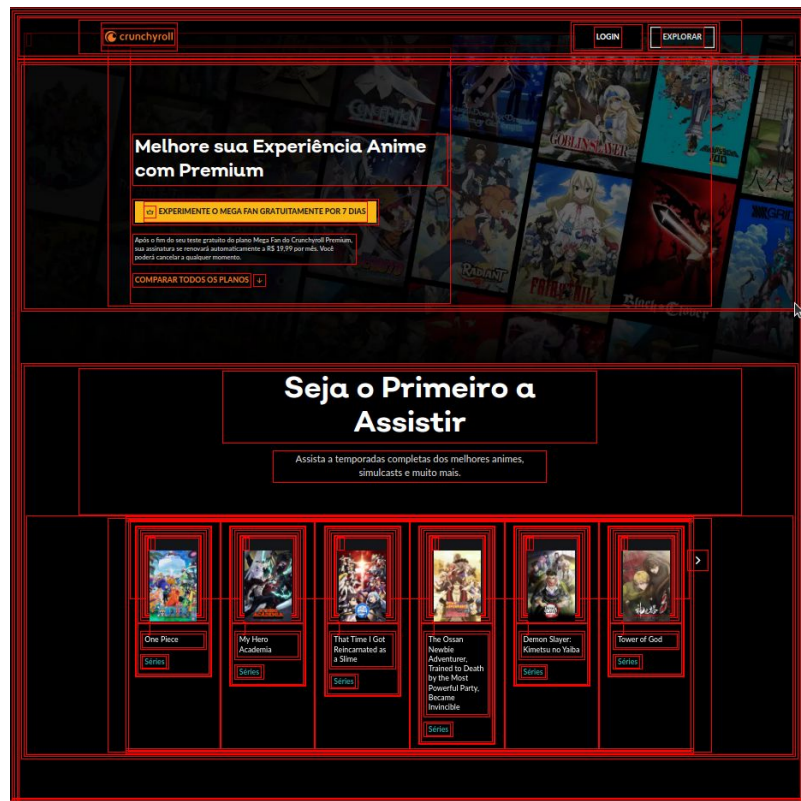
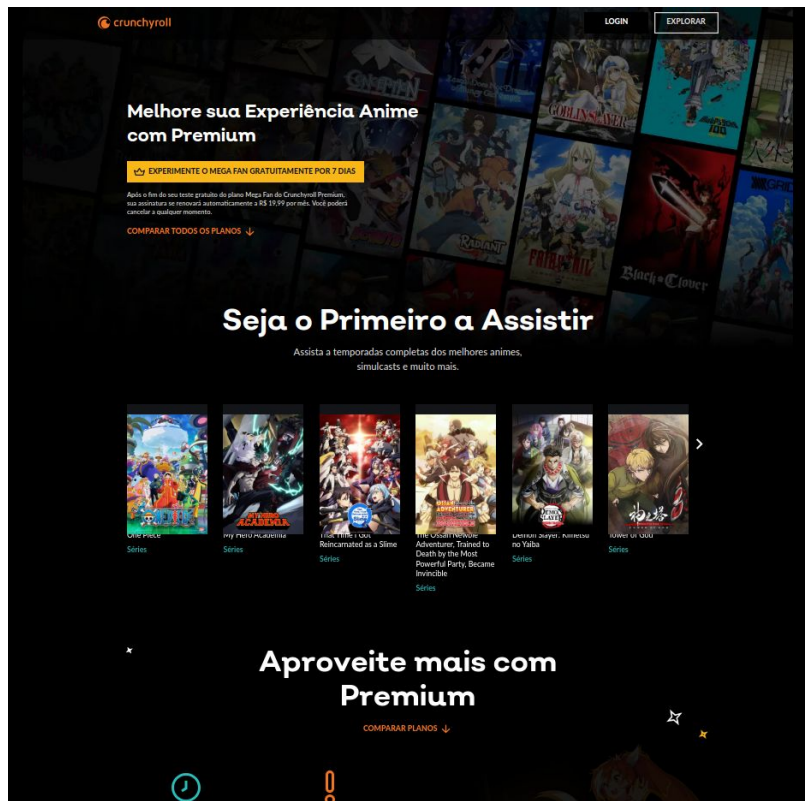
Prof. Igor Knop [igor.knop@ufjf.br](mailto:igor.knop@ufjf.br)



---

<https://bit.ly/4dtMQEE>

# Tudo se baseia em retângulos



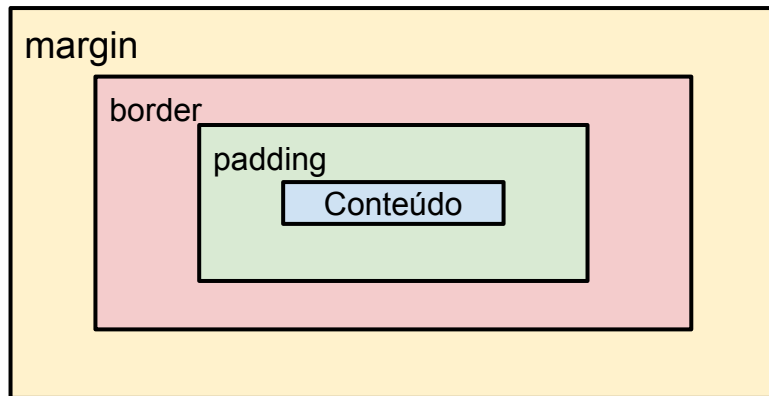
# Tudo se baseia em retângulos

O navegador é um programa que desenha retângulos.

Cada elemento HTML possui uma caixa que envolve o conteúdo e existem regras bem definidas para a exibição durante o desenho da página.

Os principais atributos são:

- **border**  
controla estilo, cor e tamanho;
- **padding**  
controla o espaçamento entre a borda e o conteúdo;
- **margin**  
controla a distância entre elementos.

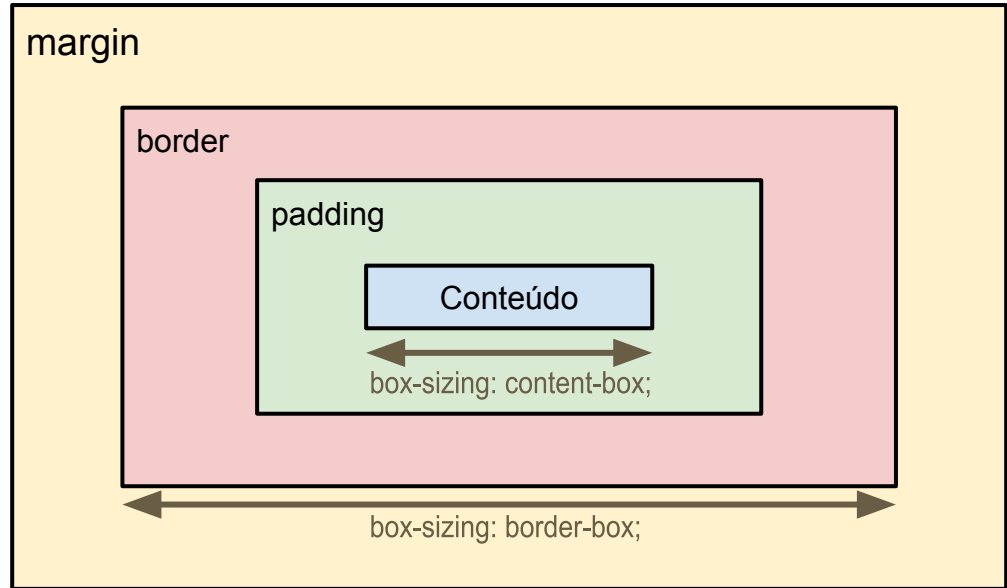


# Box Sizing

O navegador vai automaticamente associar um tamanho a cada elemento com base nas suas regras de exibição (propriedade **display**). Mas também é possível congelar o tamanho com as propriedades:

- **width**  
controla a largura do elemento;
- **height**  
controla altura do elemento;
- **aspect-ratio**  
controla a relação entre largura e altura.

Por padrão, isso é referente ao conteúdo. Para levar em conta a borda, a propriedade **box-sizing** deve ser alterada para **border-box**.



# Preenchimento

A cor de fundo de um elemento é definida pela propriedade **background-color**. A cor será aplicada ao conteúdo e espaçamento, mas não à margem.

As cores podem ser definidas pelos seus apelidos ou por sistemas de cores como o *red-green-blue*, *hue-saturation-luminance*, *hue-whiteness-blackness*. Bem comum é o uso do *rgb*, mas usando os valores 0-255 em hexadecimal 00-ff.

Exemplos:

```
background-color: red;  
background-color: rgb(255, 0, 0);  
background-color: hsl(0, 100%, 50%);  
background-color: hwb(0 0% 0%);  
background-color: #ff0000;  
background-color: #f00;
```

# Largura e altura: Elementos em bloco

A altura e largura podem ser definidos de forma independente do conteúdo, congelando a dimensão.

A largura padrão para um elemento em bloco é a área de conteúdo do elemento pai e a altura padrão é o mínimo necessário para envolver o conteúdo.

- `/* tamanhos absolutos*/`  
`height: 800px;`  
`width: 600px;`  
`height: 3cm;`  
`width: 60mm;`

- `/* tamanhos relativos à fonte*/`  
`height: 1ch;`  
`height: 1em;`
- `/* tamanhos relativos à tela*/`  
`height: 100vh;`  
`width: 100vw;`
- `/* tamanhos relativos ao contêiner */`  
`width: 50%;`

Lembrando que o atributo **box-sizing: content-box/border-box;** afetam esses valores.

# Espaçamento: padding

O espaçamento é a região que vai dos limites do conteúdo até o início da borda. É definido pela propriedade **padding**.

Possui dimensão de comprimento (px, pt, cm, mm, em, ch, lh) e não é incomum ser relativo ao tamanho de fonte.

Sua cor é transparente por padrão e irá assumir a cor de fundo aplicada ao elemento.

```
padding: 1cm;  
padding-top: 5px;  
padding-left: 1em;  
padding-bottom: 2ch;
```

# Borda: border

A borda é uma região opaca que envolve a área formada pelo espaçamento. É definida pela propriedade **border**.

Ela consegue destacar bem o elemento e é necessário definir três características diferentes para ser visível:

- Espessura da linha

```
border-width: 2px;
```

```
border-width: 1mm;
```

- Cor

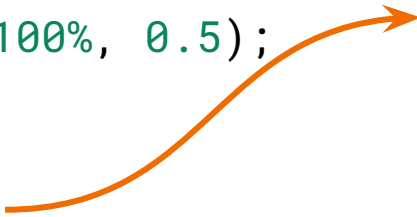
```
border-color: #c0c0c0;
```

```
border-color: rgba(120, 50%, 100%, 0.5);
```

- Estilo

```
border-style: solid;
```

```
border-style: dashed;
```



none:	nenhuma
hidden:	invisível
dotted:	pontilhada
dashed:	tracejada
solid:	sólida
double:	dupla
groove:	estilizada
ridge:	moldura
inset:	relevo interno
outset:	relevo externo



# Borda: atalhos

É bem comum utilizar o atalho para bordas ao invés de especificar individualmente cada propriedade.

```
border: 1px solid black;  
border: 2cm dotted blue;  
border: 5px double rgb(0, 255, 140);  
border: 2pt outset #a0a0a0;
```

# Margem: margin

A margem é uma região transparente que envolve a área formada pela borda. Uma margem define o espaçamento entre elementos diferentes e é controlada pela propriedade **margin** e possui dimensão de comprimento (px, pt, em, cm, mm).

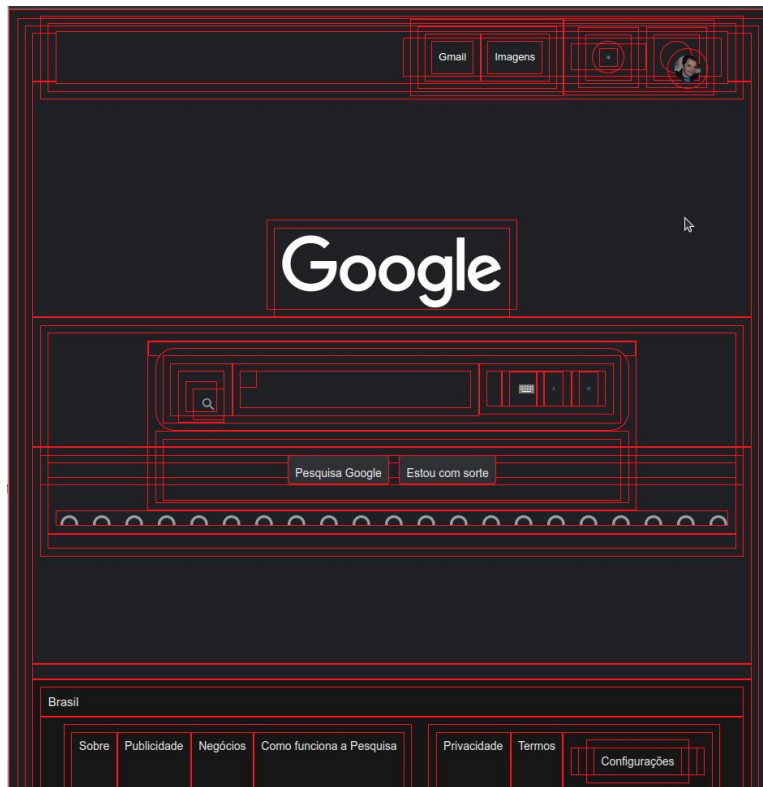
Sua cor é transparente, portanto a cor depende do preenchimento do elemento pai.

Margens horizontais se mesclam quando no fluxo normal.

# Ainda é uma árvore: relações de pais e filhos

Mesmo o navegador desenhando retângulos e os posicionando de formas diferentes, a relação pai-filho entre os elementos se mantém

A árvore do documento deve respeitar algumas regras de posicionamento para a exibição dos elementos.



# Fluxo

O fluxo é disposição dos elementos na tela de acordo com ordem de ocorrência no arquivo HTML.

Os elementos são apresentados usando uma busca em profundidade e é uma recursão:

- Começando pela raiz, exibir todos os nós filhos em ordem;
- Para cada filho, exibir os filhos destes em ordem.

As duas primeiras forma de exibição são:

- Elementos em bloco
- Elementos em linha

# Elementos: em linha vs em bloco

## Elementos em linha

- O tamanho de sua caixa é o necessário para envolver o conteúdo;
- As margens não são visíveis;
- O espaçamento não obedece à altura de linha.

## Elementos em bloco

- A largura vai ser igual ao espaço disponível do elemento pai;
- A altura e largura pode ser definidas independente do tamanho do conteúdo;
- As margens horizontais se sobrepõem.

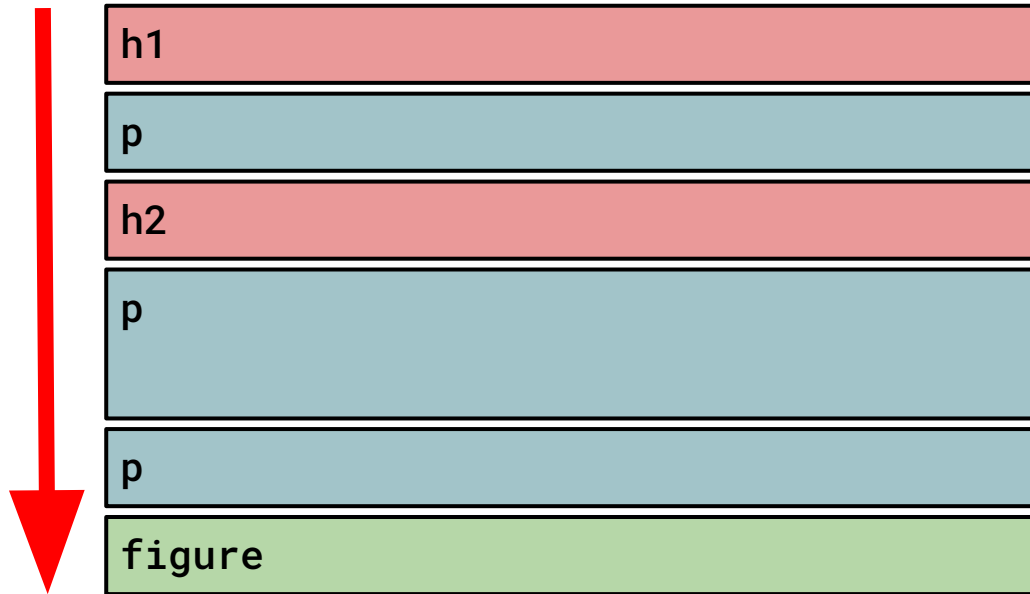
# Fluxo: elementos em bloco

Os elementos exibidos em bloco, por padrão, apresentam um fluxo vertical: são exibidos do topo para o fundo da página.

Um elemento em bloco sempre vai ocupar todo o espaço disponibilizado pelo seu pai, se ela não estiver congelada. E não aceita outro elemento ao seu lado.

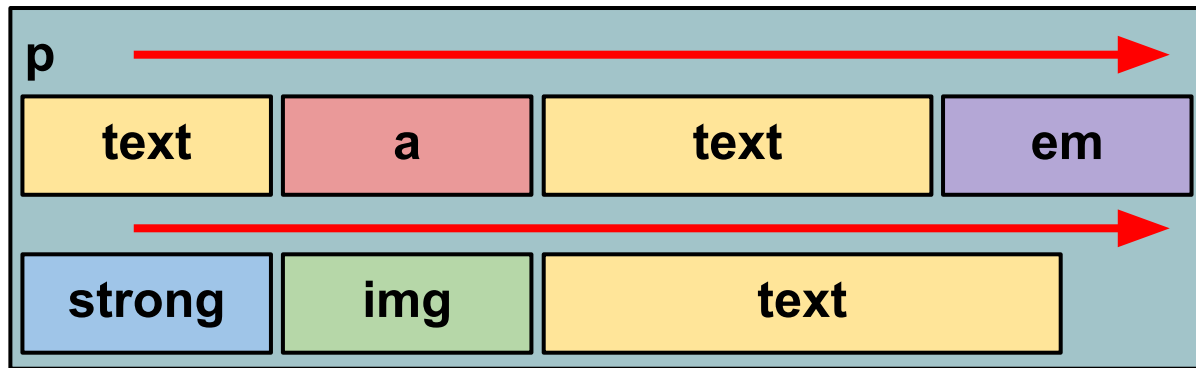
Um elemento em bloco tem a altura necessária para exibir seu conteúdo, se ela não estiver congelada.

A exibição em bloco é a padrão para o elemento livre de significado **div**.



# Fluxo: elementos em linha

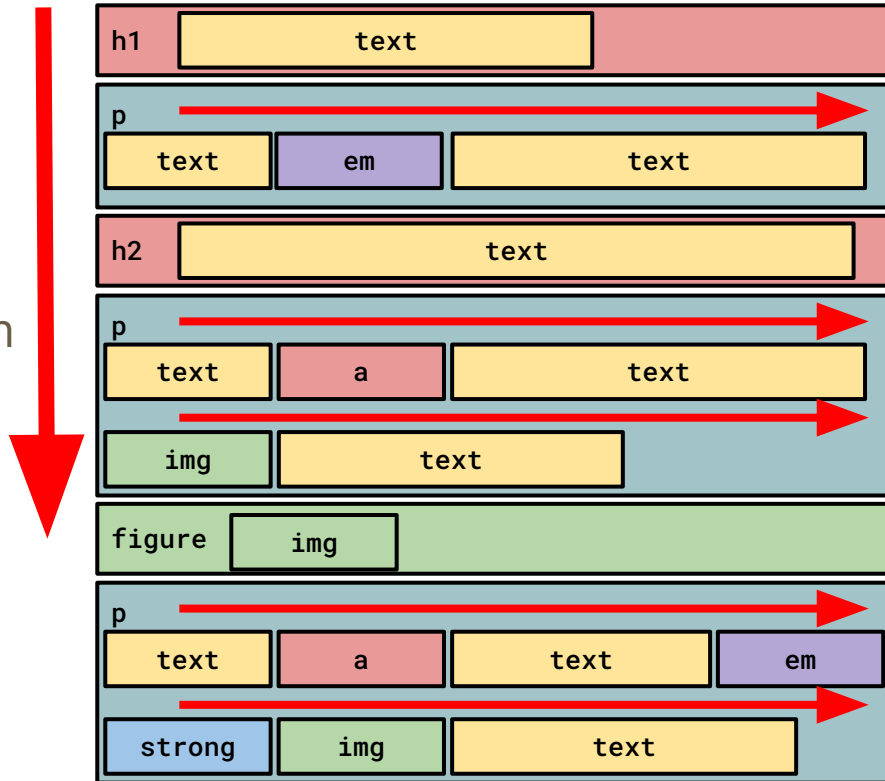
Elementos com exibição “em linha” se comportam com um fluxo horizontal, seguindo a exibição de texto fluído. Há uma quebra de linha quando não há espaço disponível na caixa do elemento contêiner. O elemento livre de significado **span** tem por padrão exibição em linha.



# Fluxo: blocos e elementos em linha

O fluxo é disposição dos elementos na tela de acordo com ordem de ocorrência no arquivo HTML.

A mais simples das páginas é organizada por elementos em bloco em sequência, que internamente possuem conteúdo em texto fluído intercalado com elementos em linha.





# O FlexBox

O FlexBox é uma forma de exibição moderna utilizada para dispor elementos em uma única dimensão. Um contêiner flex vai controlar o tamanho dos seus elementos filhos, tornando-os itens flex.

Três propriedades são as mais importantes para definir o flex:

**display:** `flex`;

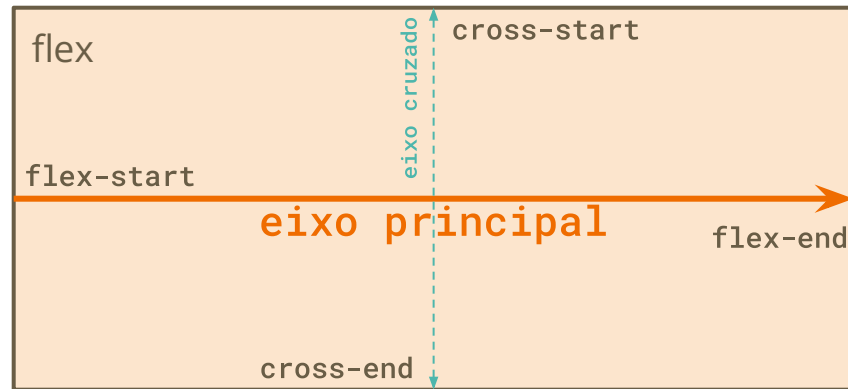
Coloca o elemento como elemento em bloco em seu pai e seus filhos como flex-items;

**justify-content:** `center`;

Define o alinhamento dos filhos em relação ao eixo principal.

**align-items:** `center`;

Define como alinhar os filhos em relação ao eixo cruzado.



# O Grid

O modo de exibição em grid é uma forma de realizar layout de elementos em duas dimensões.

Três propriedades são as mais importantes para definir o grid:

**display:** `grid`;

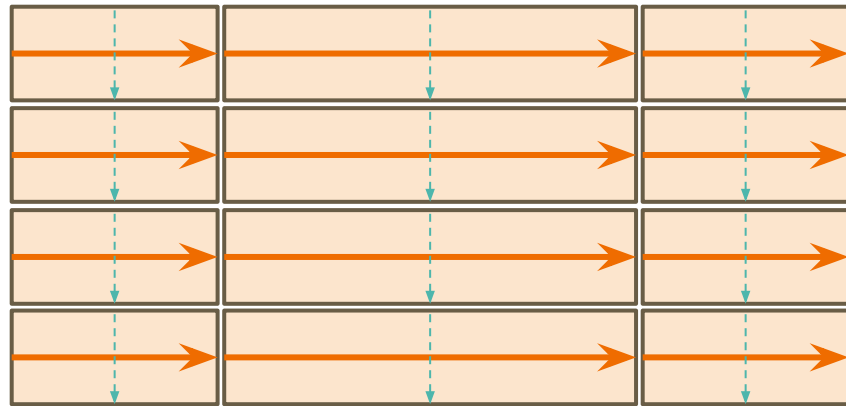
Coloca o elemento como elemento em bloco em seu pai e seus filhos como grid-items;

**grid-template-columns:** `1fr 2fr 1fr`;

Define em quantas e qual o tamanho para cada coluna.

**place-items:** `center center`;

Define como alinhar os filhos, em cada caixa, contra o eixo principal e cruzado



# O Grid autofill

Sempre que houver uma necessidade de organizar os elementos em duas dimensões, prefira o grid.

# 0 Grid: grid areas

Sempre que houver uma necessidade de organizar os elementos em duas dimensões, prefira o grid.

# Posicionamento

Podemos controlar como um elemento em bloco é posicionado na página através de um conjunto de propriedades. A primeira delas é a propriedade **position**.

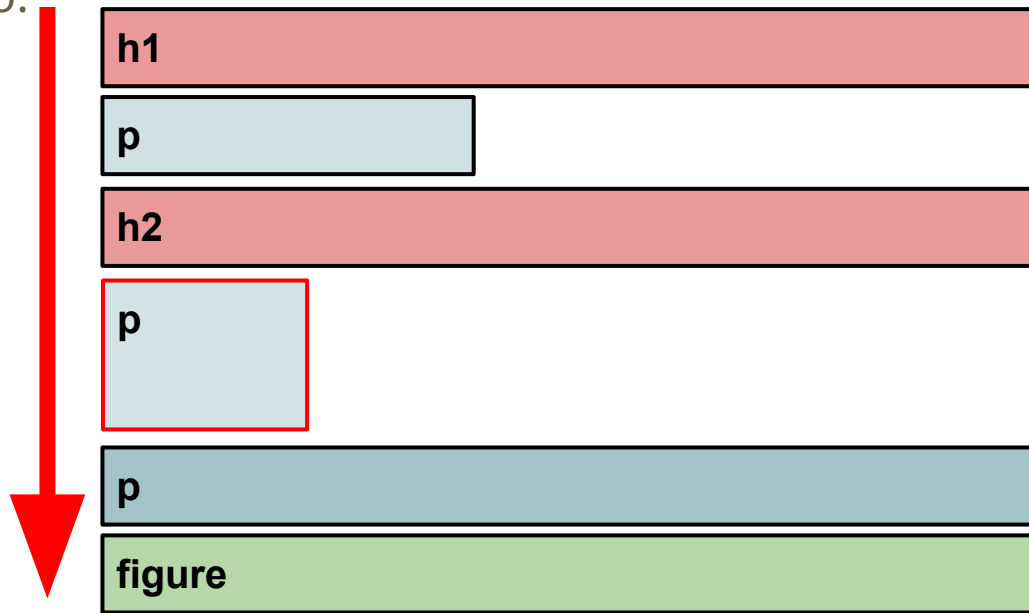
```
position: static;  
position: absolute;  
position: relative;  
position: fixed;  
position: sticky;
```

# Posicionamento: static

Elementos são posicionados estaticamente por padrão no fluxo.

Propriedades **top**, **bottom**, **left** e **right** não têm efeito neste caso

```
h2+p {  
  position: static;  
}
```



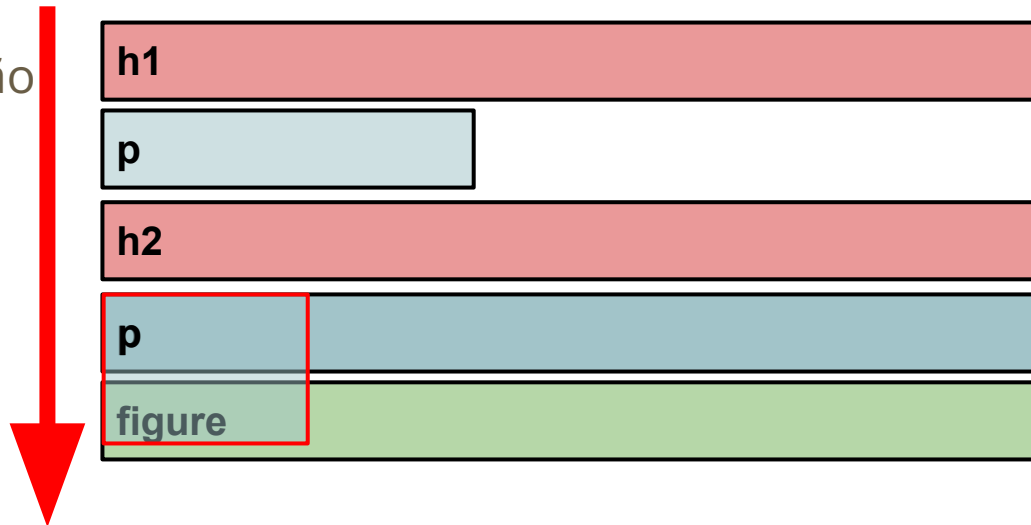
# Posicionamento: absoluto

Elementos posicionados absolutamente saem do fluxo.

Inicialmente fica na mesma posição que estaria no fluxo.

```
h2+p {  
  position: absolute;  
}
```

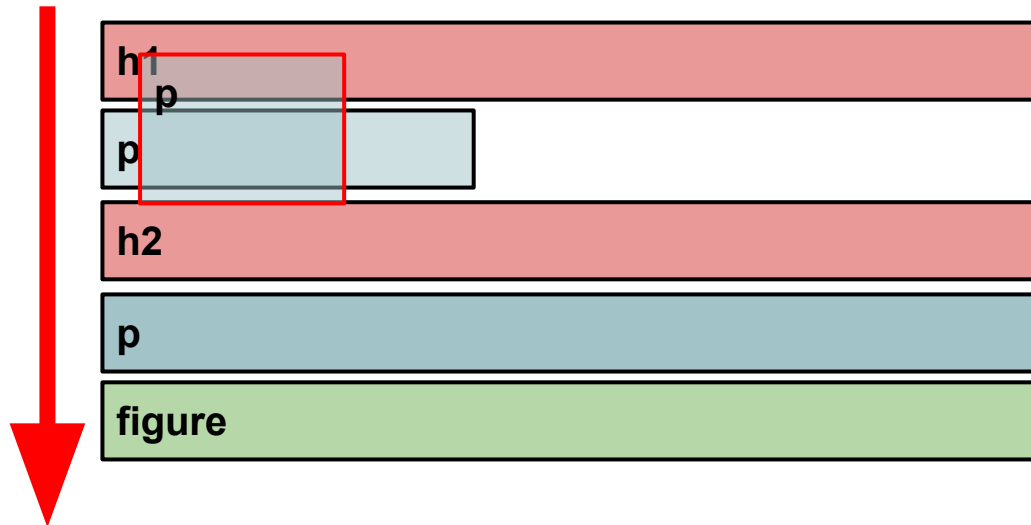
Não reservam espaço para ele no fluxo.



# Posicionamento: absoluto (2)

Elementos posicionados absolutamente podem ter sua posição definida em relação às bordas do primeiro ancestral posicionado.

```
h2+p {  
  position: absolute;  
  top: 10px;  
  left: 20px;  
}
```





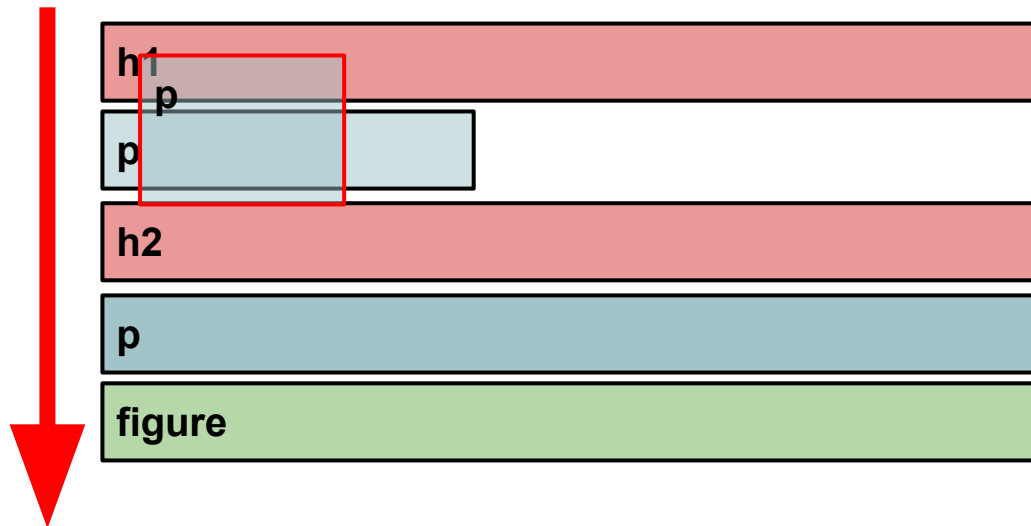
# Posicionamento: fixed

Elementos posicionados fixamente saem do fluxo.

Inicialmente ficam na mesma posição que estariam no fluxo.

```
h2+p {  
  position: fixed;  
}
```

Não reserva espaço no fluxo.

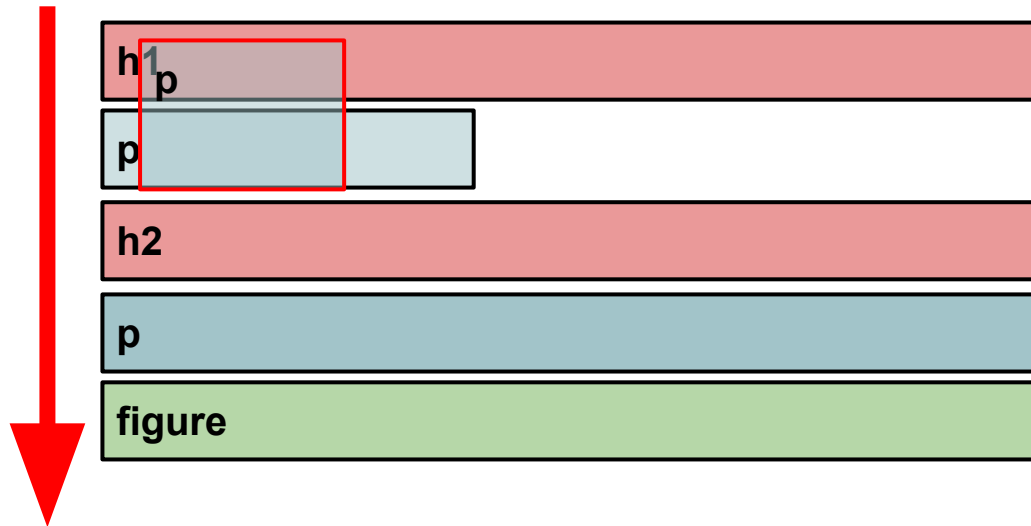


# Posicionamento: fixed (2)

Elementos posicionados absolutamente podem ter sua posição definida em relação à janela visível do navegador.

Eles "grudam na tela"!

```
h2+p {  
  position: fixed;  
  top: 10px;  
  left: 20px;  
}
```



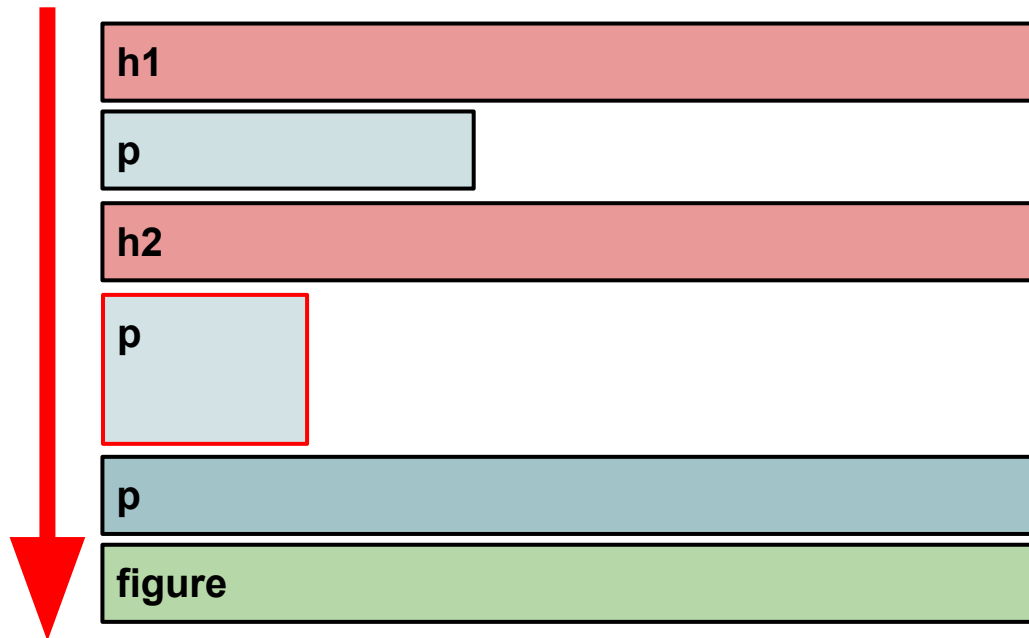
# Posicionamento: relative

Elementos posicionados relativamente não saem do fluxo.

Inicialmente ficam na mesma posição que estariam no fluxo.

```
h2+p {  
  position: relative;  
}
```

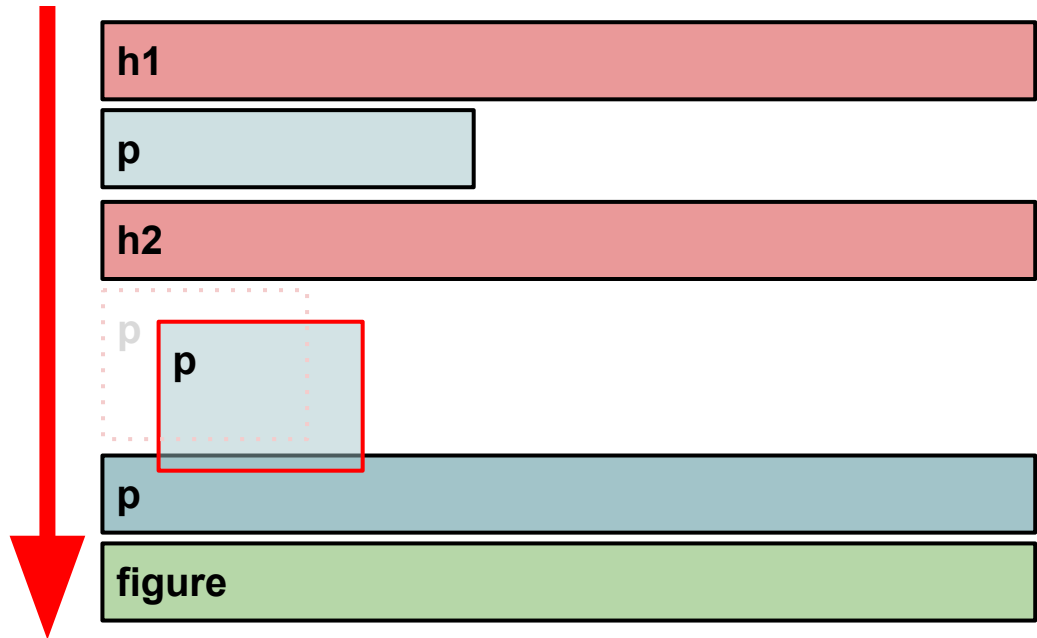
O espaço para ele fica reservado no fluxo.



# Posicionamento: relative (2)

Elementos posicionados absolutamente podem ter sua posição definida em relação à posição que ele iria ocupar no fluxo.

```
h2+p {  
  position: fixed;  
  top: 10px;  
  left: 20px;  
}
```



# Conteúdo em excesso

Quando um elemento possuir uma largura e altura fixadas, o conteúdo pode não caber no espaço disponível

Como isso será exibido é definido pela propriedade:

## **overflow**

Os valores possíveis são:

- **visible**: o excesso sai pelos limites da borda
- **hidden**: o excesso fica escondido dentro dos limites da borda
- **scroll**: barras de rolagem são adicionadas ao elemento
- **auto**: barras de rolagem aparecem apenas quando necessário

# Para saber mais...

- Yoksel. **FlexBox Cheat Sheet**. 2024. Available on Internet:  
<<https://yoksel.github.io/flex-cheatsheet/>>
- Coyier, Chris. **A Complete Guide to Flexbox**. 2024. Available on Internet:  
<<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>>
- MDN. **Basic Concepts of Grid Layout**. 2024. Available on internet:  
<[https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_grid\\_layout/Basic concepts of grid layout](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_grid_layout/Basic_concepts_of_grid_layout)>