



UNIVERSIDADE  
FEDERAL DO CEARÁ

CAMPUS QUIXADÁ

# Fundamentos de CSS

QXD0020 - Desenvolvimento de Software para Web

Prof. Bruno Góis Mateus ([brunomateus@ufc.br](mailto:brunomateus@ufc.br))

# Agenda

- Introdução
- Seletores
- Pseudo-classes
- Pseudo-elementos
- O efeito em cascata
- CSS Box Model
- Posicionamento
- Layouts
- Boa práticas

# Introdução



# Introdução

## CSS - Cascade Style Sheet

- Descreve a **aparência e a disposição das informações** em uma página web
  - Diferentemente do HTML que descreve o conteúdo
  - Pode ser utilizado de duas formas
    - Incorporado ao HTML
    - Em arquivos separados **.css**

# Introdução

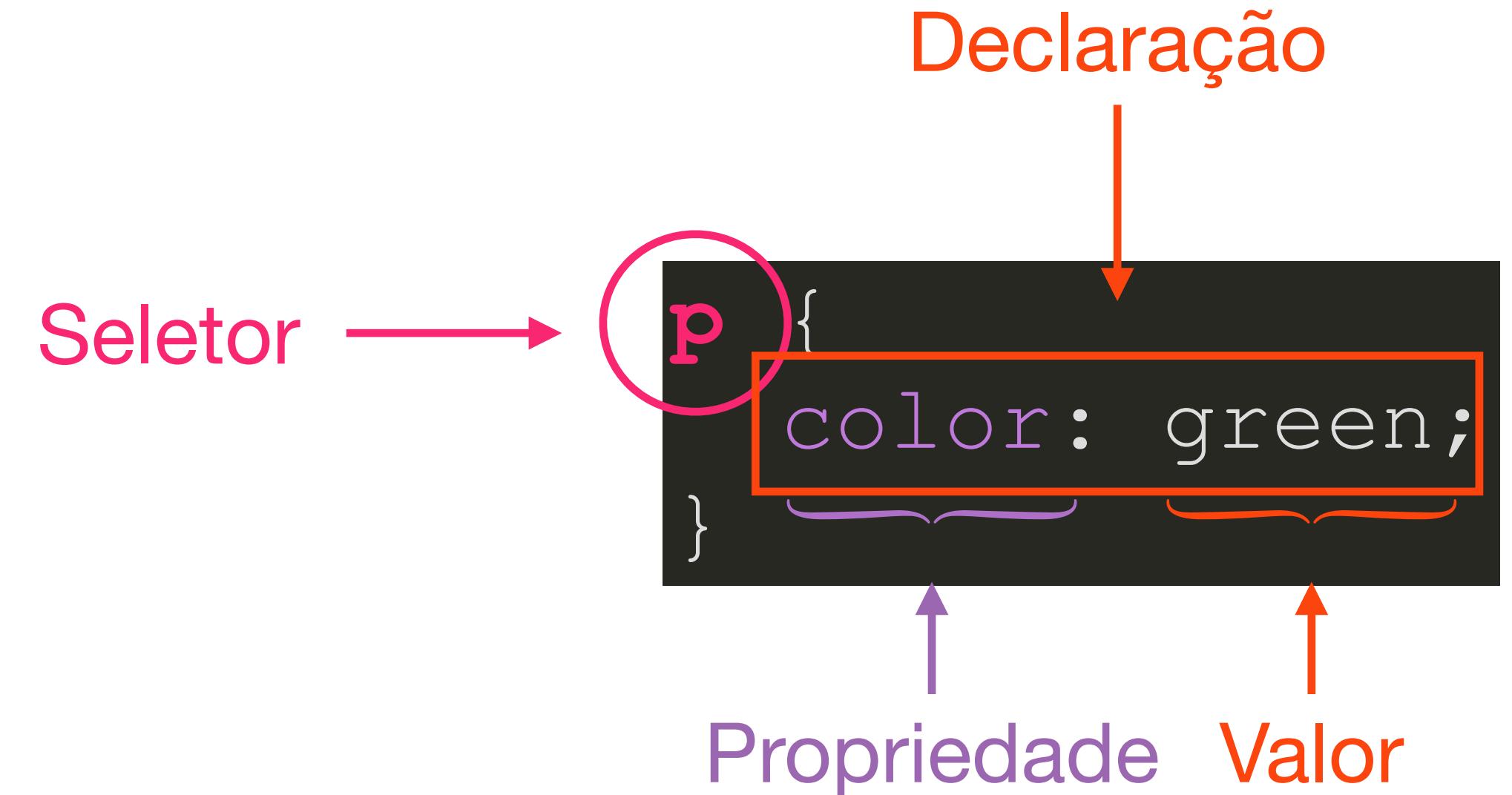
## Vantagens

- Manter o estilo da documento completamente separado do conteúdo
- Melhora o reuso do conteúdo da página web
- Regras de estilo separadas podem ser utilizadas para diferentes tipos de media
- Diminui o consumo de largura de banda, já que as folhas de estilo são armazenadas em cache

# Introdução

## Regra CSS - Sintaxe Básica

- Possui **três** partes:
  - **Seletor**
    - Elemento HTML identificado por sua tag, classe ou ID, e para o qual a regra será válida
  - **Propriedade**
    - Atributo do elemento HTML ao qual será aplicada a regra
  - **Valor**
    - Característica específica a ser assumida pela propriedade



# Introdução

## Tipos de folha de estilo

- Existem três tipos de CSS
  - Folhas de estilo externa
    - As regras são descritas em um arquivo externo (separado do HTML) e este é referenciado através da `<link>` ou utilizando a regra `@import`
  - Folha de estilo interna
    - As regras são separadas do conteúdo em HTML, mas estão contidas no próprio documento web, utilizando a tag `<style>`
  - Folha de estilo em linha (*inline*)
    - Colocar as regras CSS em linha, utilizando o **atributo style** das tags HTML

# Introdução

## Folha de estilo externa

### Folha de estilo externa

- Ideal para ser aplicada a várias páginas
  - Com uma folha de estilo é possível mudar a aparência do site inteiro
  - O arquivo css deve ser linkado ou importado ao documento HTML
    - O elemento link deve estar contido dentro do elemento head
    - O elemento style também deve estar contido no head do documento

# Introdução

## Folha de estilo externa

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="/style.css" media="screen" >
  </head>
</html>
```

- O atributo media especifica os tipos de media onde o css será aplicado
  - Se omitido o valor padrão de atributo é all
  - Para múltiplas medias, deve ser utilizados:
    - Ex: media = “screen, projection”

```
<!DOCTYPE html>
<html>
  <head>
    <style type="text/css" >
      @import url("estilos.css");
    </style>
  </head>
</html>
```

# Introdução

## Folha de estilo interna

### Folha de estilo interna

- As regras CSS são declaradas no próprio documento HTML
  - Ideal para ser aplicada a uma única página
  - São declaradas na seção head do documento
- Assim como nas folhas de estilo externas, temos que definir o atributo type e podemos definir o atributo media

# Introdução

## Folha de estilo interna

```
<!DOCTYPE html>
<html>
  <head>
    <style type="text/css" media="all">
      body {
        background-color:#000 ;
        color: red;
      }
    </style>
  </head>
</html>
```

# Introdução

## Folha de estilo inline

### Folha de estilo inline

- As regras CSS estão declaradas dentro da tag do elemento HTML
- Um estilo inline se aplica a um elemento HTML
  - Apresentam forte acoplamento entre o conteúdo e a camada de apresentação
  - Estilos inline são mais limitados que estilos externos e internos
    - Não é possível utilizar o atributo media
    - Não é aplicável em pseudo-classes e pseudo-elementos

# Introdução

## Folha de estilo inline

```
<!DOCTYPE html>
<html>
<body>
  <p style="color: blue; background-color: yellow;">
    O conteúdo desta tag será exibido em azul com fundo amarelo no navegador!
  </p>
</body>
</html>
```

# Seletores



```
color: green;  
}
```

# Seletores

## Seletor por tag

- Aplica a regra em todos os elementos citados seletor e que estão presentes no arquivo HTML
- Seletores de elementos **não são case sensitive**
  - **Boa prática:** Utilizar letras minúsculas

```
p {  
    color: #5EC85E;  
}
```

Este parágrafo utiliza a regra ao lado

# Seletores

## Seletor de identificador (id)

- Aplica a regra no elemento cujo o **id** possui o valor especificado na regra
  - O atributo **id** deve ser único
  - A regra deve ser aplicada apenas em um elemento
- Maior especificidade entre os seletores

```
#id_do_elemento {  
    color: #5EC85E;  
}
```

# Seletores

## Seletor por id

```
p {  
    color: purple;  
}
```

```
#missao {  
    font-style: italic;  
    color: orange;  
}
```

```
<p> Cidade de Esparta </p>  
<p id="missao" > A nossa missão é  
fornecer mais espartanos </p>
```

Cidade de Esparta  
*A nossa missão é fornecer mais espartanos*

# Seletores

## Seletor por classe (class)

- Aplica a regra nos elementos baseado no **atributo class** do elemento
- Aplicável em elementos que possuem **mais de uma classe**

```
.nome_da_classe{  
    text-align: center;  
}
```

# Seletores

## Seletor por classe

```
.red {  
    color: #f33;  
}
```

```
.yellow-bg {  
    background-color : #ffa;  
}
```

```
.fancy {  
    font-weight: bold;  
    text-shadow: 4px 4px 3px blue;  
}
```

```
<p class="red">This paragraph has red  
text.</p>  
<p class="red yellow-bg">This paragraph has  
red text and a yellow background.</p>  
<p class="red fancy">This paragraph has red  
text and "fancy" styling.</p>  
<p>This is just a regular paragraph.</p>
```

This paragraph has red text.

This paragraph has red text and a yellow background.

This paragraph has red text and "fancy" styling.

This is just a regular paragraph

# Seletores

## Seletor por classe

```
.warning {  
    background-color: red;  
}  
  
p.warning {  
    color: white;  
}  
  
.special {  
    background-color: yellow;  
    font-weight: bold;  
}  
  
.warning.special {  
    color: blue;  
}
```

```
<p class="warning"> Texto 1 </p>  
<h1 class="warning"> Texto 2 </h1>  
<p class="warning special"> Texto 3 </p>
```

Texto 1

Texto 2

Texto 3

# Seletores

## Seletor por atributo

- Irá combinar com elementos baseado na presença de um atributo ou de acordo com o seu valor

```
tag_name[atributo] {  
    text-align: center;  
}
```

# Seletores

## Seletor por atributo

```
a[href^="#"] {  
    background-color: gold;  
}  
a[href*="example"] {  
    background-color: silver;  
}
```

```
a[href*="insensitive" i] {  
    color: cyan;  
}
```

```
a[href$=".org"] {  
    color: red;  
}
```

```
a[href^="https"] [href$=".org"] {  
    color: green;  
}
```

```
<ul>  
    <li><a href="#internal">Interno</a></li>  
    <li><a href="http://example.com">Exemplo</a></li>  
    <li><a href="#InSensitive">Insensitive internal</a></li>  
    <li><a href="http://example.org">Exemplo org</a></li>  
    <li><a href="https://example.org">Exemplo https org</a></li>  
</ul>
```

Interno

Exemplo

Insensitive Interno

Exemplo org

Exemplo https org

# Seletores

## Seletor universal

- Este seletor básico irá escolher todos os nós
- Pode ser omitido

```
* {  
    text-align: center;  
}
```

# Seletores

## Seletor universal

```
* [lang^="en"] {  
    color: green;  
}
```

```
* .warning {  
    color: red;  
}
```

```
* #maincontent {  
    border: 1px solid blue;  
}
```

```
<p class="warning">  
    <span lang="en-us">Um span verde</span>  
no parágrafo vermelho  
</p>
```

```
<p id="maincontent" lang="en-gb">  
    <span class="warning">A red span</span>  
in a green paragraph.  
</p>
```

Um span verde no parágrafo vermelho

A red span in a green paragraph

Além dos seletores mostrados, é possível agrupar e combinar seletores

# Seletores

## Seletor de grupo

- Aplica a regra em cada seletor especificado no seletor de grupo

```
elemento1, elemento2 {  
    text-align: center;  
}
```

# Seletores

## Seletor de grupo

```
h1, h2, h3, span.titulo {  
    color:red;  
}
```

```
<h1> Cabeçalho 1</h1>  
<h3> Cabeçalho 3</h3>  
<p> Meu  
    <span class="titulo">subtitulo</span>  
</p>
```

Cabeçalho 1

Cabeçalho 3

Meu **subtitulo**

# Seletores

## Seletor de descendência

- É uma combinação de dois ou mais seletores simples separados por um espaço em branco
- Combina elementos que sejam descendentes do primeiro elemento simples declarado

```
elemento1 elemento2 {  
    text-align: center;  
}
```

# Seletores

## Seletor de descendência

```
#ad li.important strong {  
    text-decoration: underline;  
    color: oragen;  
}
```

```
<div id="ad">  
    <p> Loja<strong> Hardware Hardwick </strong></p>  
    <ul>  
        <li class="important"> Os <strong> melhores </strong> preços da cidade! </li>  
        <li> Enquanto <strong> durar o estoque</strong>! </li>  
    </ul>  
</div>
```

**Loja Hardware Hardwick**

Os melhores preços da cidade

Enquanto durar o estoque

# Seletores

## Seletor de filhos

- Tem como alvo apenas os filhos de um elemento
- Um ou mais seletores simples separados por >

```
elemento1 > elemento2 {  
    text-align: center;  
}
```

# Seletores

## Seletor de filhos

```
strong {  
    color: green;  
}  
  
div > strong {  
    color: purple;  
}
```

```
<div>  
    Eu gosto de uva <strong>roxa</strong>  
    <p>Eu prefiro uva<strong>verde</strong></p>  
</div>
```

Eu gosto de uva **roxa**

Eu prefiro uva **verde**

# Seletores

## Seletor de irmãos

- Tem como alvo os elementos que são irmãos do elemento especificado anteriormente e que aparecem após ele
- Um ou mais seletores simples separados por ~

```
elemento1 ~ elemento2 {  
    text-align: center;  
}
```

# Seletores

## Seletor de irmãos

```
span {  
    color: blue;  
}  
  
p ~ span {  
    color: red;  
}
```

<span>Irmão que vem antes.</span>  
<p>Irmão principal.</p>  
<span>Irmão que vem depois!</span>  
<code>Irmão que vem depois!</code>  
<span>Outro irmão</span>  
<h2>Mais um irmão. </h2>  
<span>Último irmão.</span>

Irmão que vem antes.

Irmão principal.

Irmão que vem depois! Irmão que vem depois! Outro irmão

Mais um irmão.

Último irmão.

# Seletores

## Seletor de irmãos adjacentes

- Tem como alvo um elemento que seja irmão e adjacente ao primeiro
- Um ou mais seletores simples separados por `+`

```
elemento1 + elemento2 {  
    text-align: center;  
}
```

# Seletores

## Seletor de irmãos adjacentes

```
span {  
    color: blue;  
}  
  
p + span {  
    color: red;  
}
```

<span>Irmão que vem antes.</span>  
<p>Irmão principal.</p>  
<span>Irmão que vem depois!</span>  
<code>Irmão que vem depois!</code>  
<span>Outro irmão</span>  
<h2>Mais um irmão. </h2>  
<span>Último irmão.</span>

Irmão que vem antes.

Irmão principal.

Irmão que vem depois! Irmão que vem depois! Outro irmão

Mais um irmão.

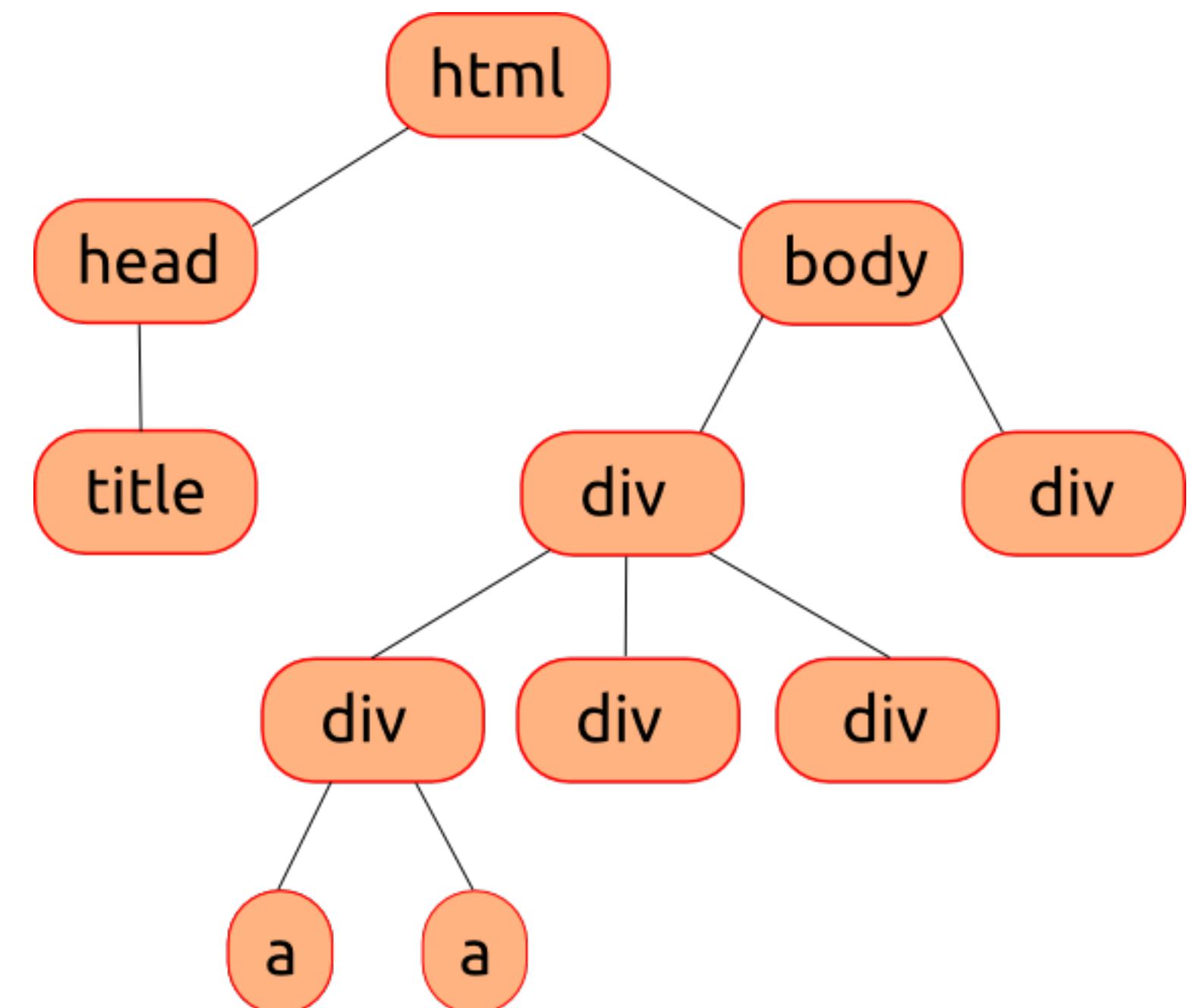
Último irmão.

# Pseudo-classes

```
a:hover {  
    color: green;  
}
```

# Pseudo-classes

- São similares a uma classe CSS, **porém não são especificadas na marcação**
- Exemplo:
  - Não existe um elemento que marque que um link já foi visitado
- Podem ser dinâmicas
  - São aplicadas de acordo com a interação do usuário com o documento
- Um seletor de pseudo-classe começam com :



# Pseudo-classes

Pseudo-classe	Descrição
<b>:active</b>	Link no momento em que ele é clicado
<b>:hover</b>	Quando usuário põe o cursor em cima de um elemento
<b>:visited</b>	Link já visitado
<b>:focus</b>	Quando um elemento recebe o foco

# Pseudo-classes

```
li:first-child {  
    color: red;  
}  
  
li:nth-child(2n) {  
    background-color: green;  
}  
  
li:nth-child(3) {  
    color: orange;  
}  
  
li:last-child {  
color: blue;  
}
```

```
<ul>  
    <li> Primeiro item </li>  
    <li> Segundo item </li>  
    <li> Terceiro item </li>  
    <li> Quarto item </li>  
</ul>
```

**Primeiro item**

**Segundo item**

**Terceiro item**

**Quarto item**

# Pseudo-classes

- Lista completa de todas as pseudos-classes disponíveis
  - <https://developer.mozilla.org/pt-BR/docs/Web/CSS/Pseudo-classes>

# Pseudo- elementos

```
a::after {  
    color: green;  
}
```

# Pseudo-elementos

- Assim como as pseudo-classes, combinam com elementos virtuais não especificados na marcação
- Permitem selecionar certo tipo de conteúdo que não possuem elemento específico para ele

# Pseudo-elementos

```
p::first-letter {  
    font-size: 200%;  
}  
  
p::first-line {  
    font-style: italic;  
}
```

**<p>**Você **<br>** pode utilizar apenas um pseudo-elemento em um seletor.**</p>**

**<p>** Ele **<br>** deve aparecer depois da declaração de um elemento simples**</p>**.

**V**ocê

**pode utilizar apenas um pseudo-elemento em um seletor**

**E**le

**deve aparecer depois da declaração de um elemento simples**

# Pseudo-elementos

```
a::before {  
    content: ' [ ';  
}  
  
a::after {  
    content: ' ] ';  
}  
  
a[href^="http://"]::after {  
    content: ' ] (externo) ';  
}
```

```
<a href="#">Link 1</a><br>  
<a href="#">Link 2</a><br>  
<a href="http://">Link 3</a><br>
```

[Link 1]

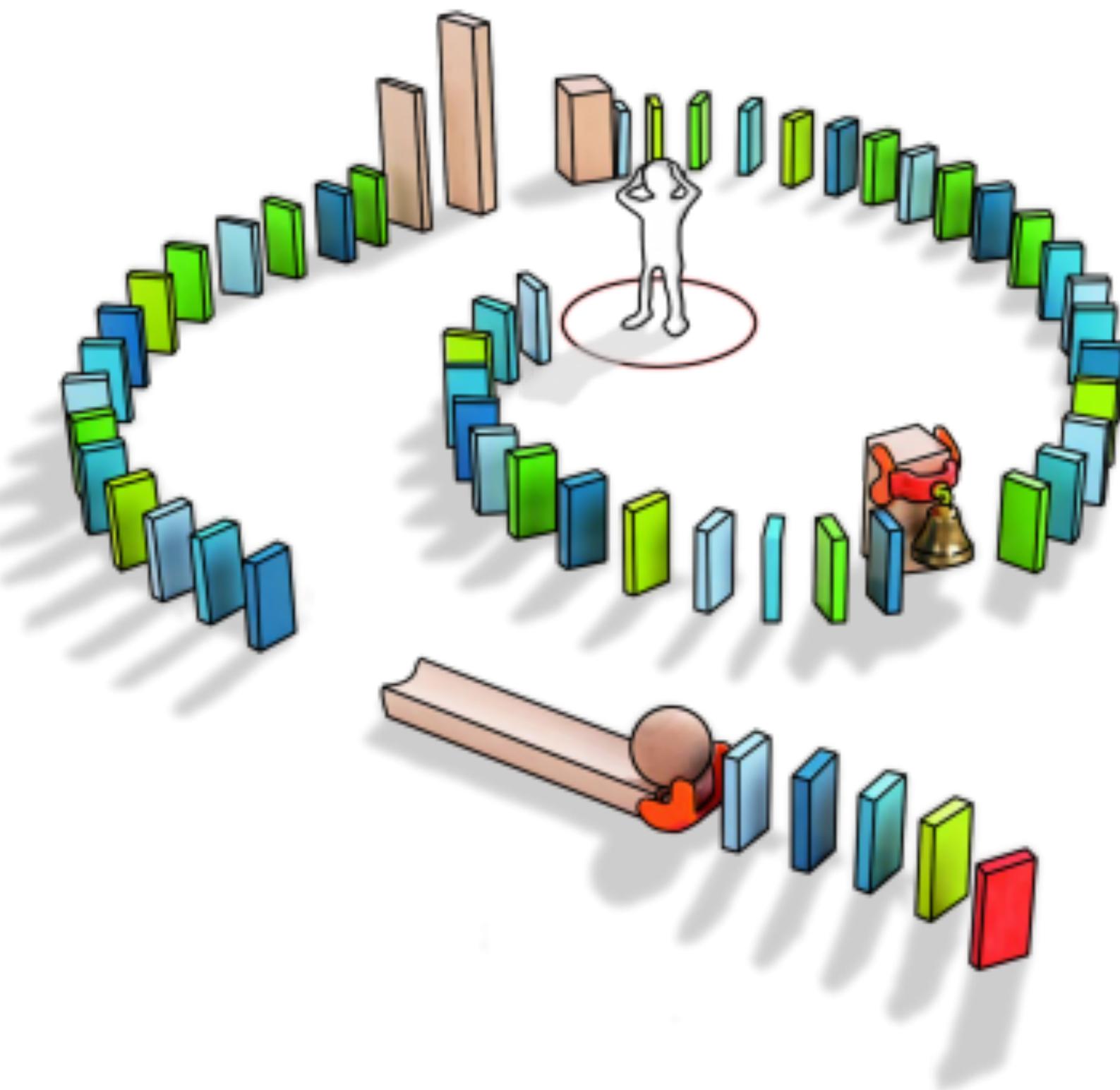
[Link 2]

[Link 3](externo)

# Pseudo-elementos

- Lista completa de todas as pseudos-elementos disponíveis
  - <https://developer.mozilla.org/pt-BR/docs/Web/CSS/Pseudo-elements>

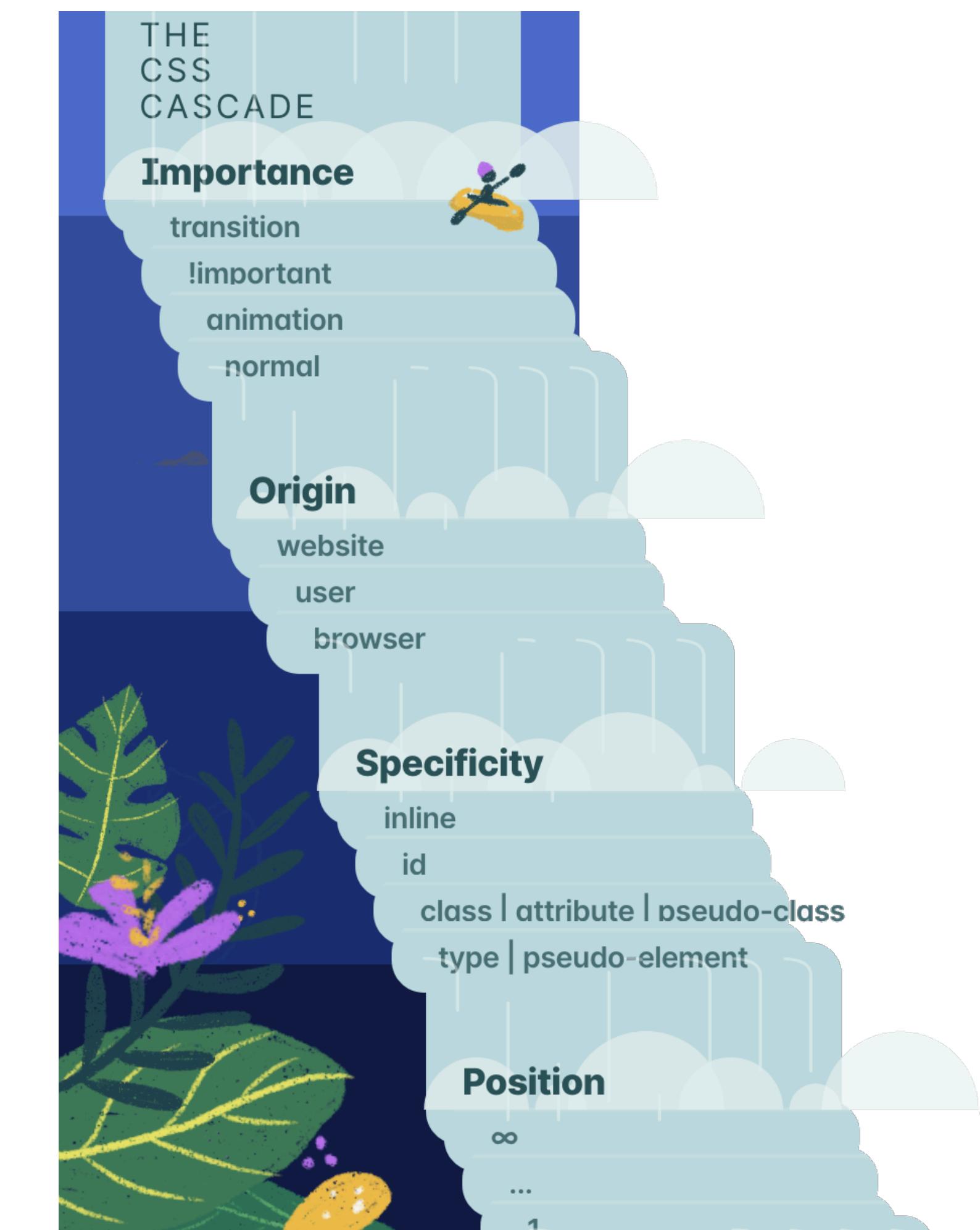
# Efeito em cascata



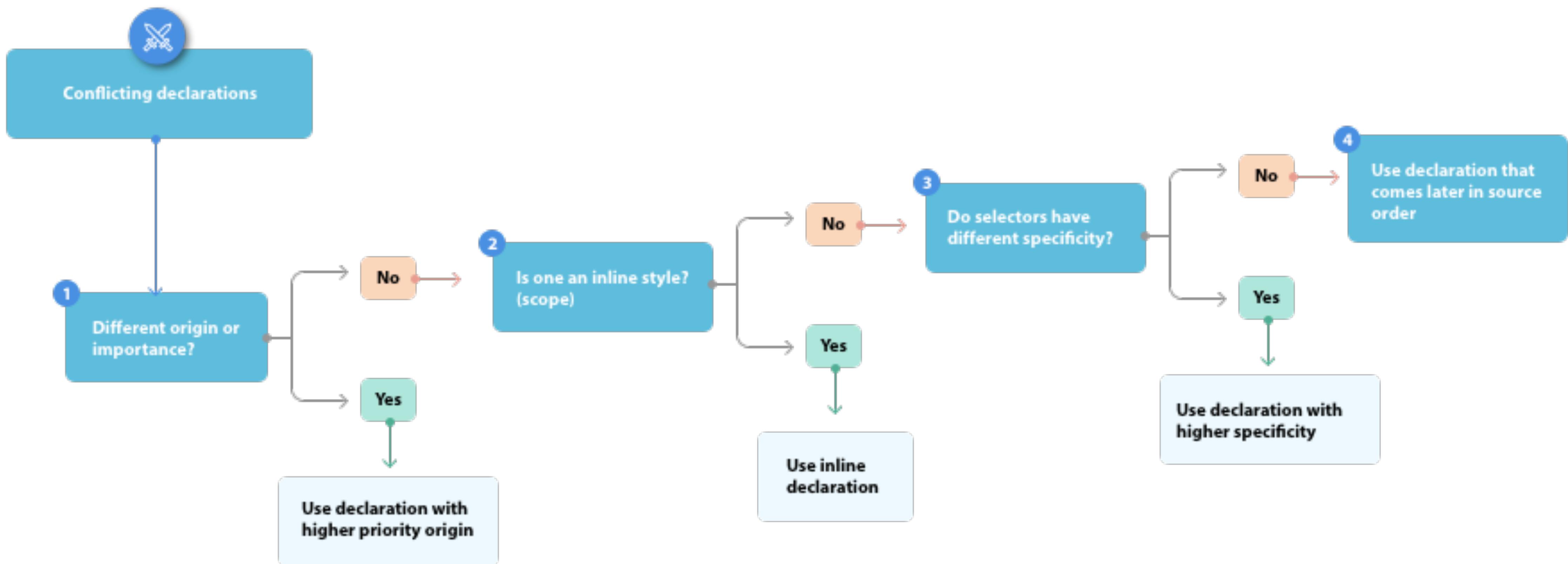
# Efeito em cascata

## Efeito em Cascata

- É o estabelecimento de uma prioridade para aplicação da regra de estilo ao elemento
- É a maneira do navegador resolver efeitos conflitantes



# Efeito em cascata

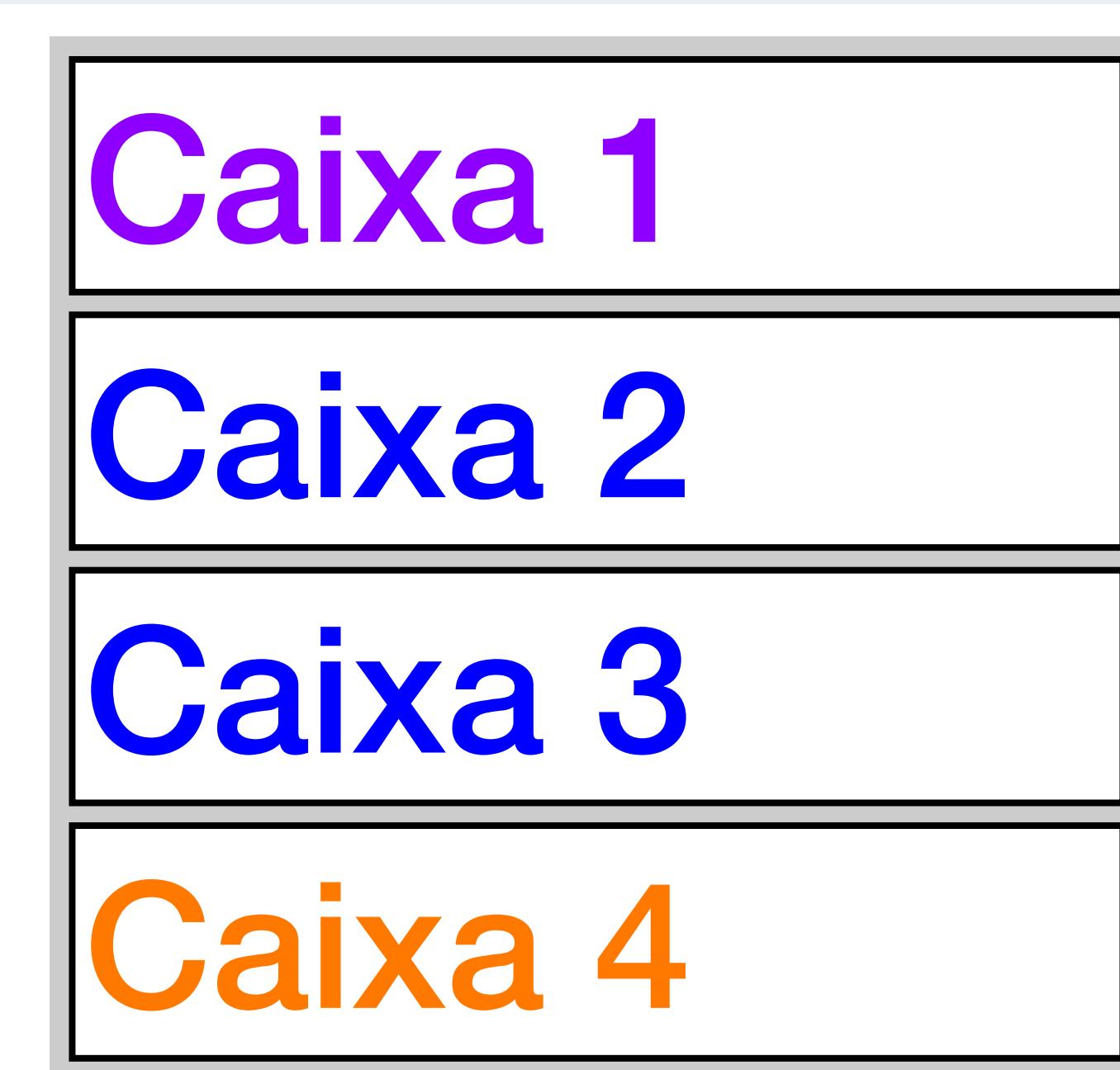


# Efeito em cascata

```
.parent {  
    color: blue;  
    background-color: gray;  
    width: 500px;  
    height: 500px;  
}  
  
.special.box {  
    color: red;  
}  
  
#b {  
    color: orange;  
}  
  
div > .box {  

```

```
<div class="parent">  
    <div class="box" style="color: purple;">Caixa 1</div>  
    <div class="box" id="a">Caixa 2</div>  
    <div class="box" >Caixa 3</div>  
    <div class="box special" id="b">Caixa 4</div>  
</div>
```



# Efeito em cascata

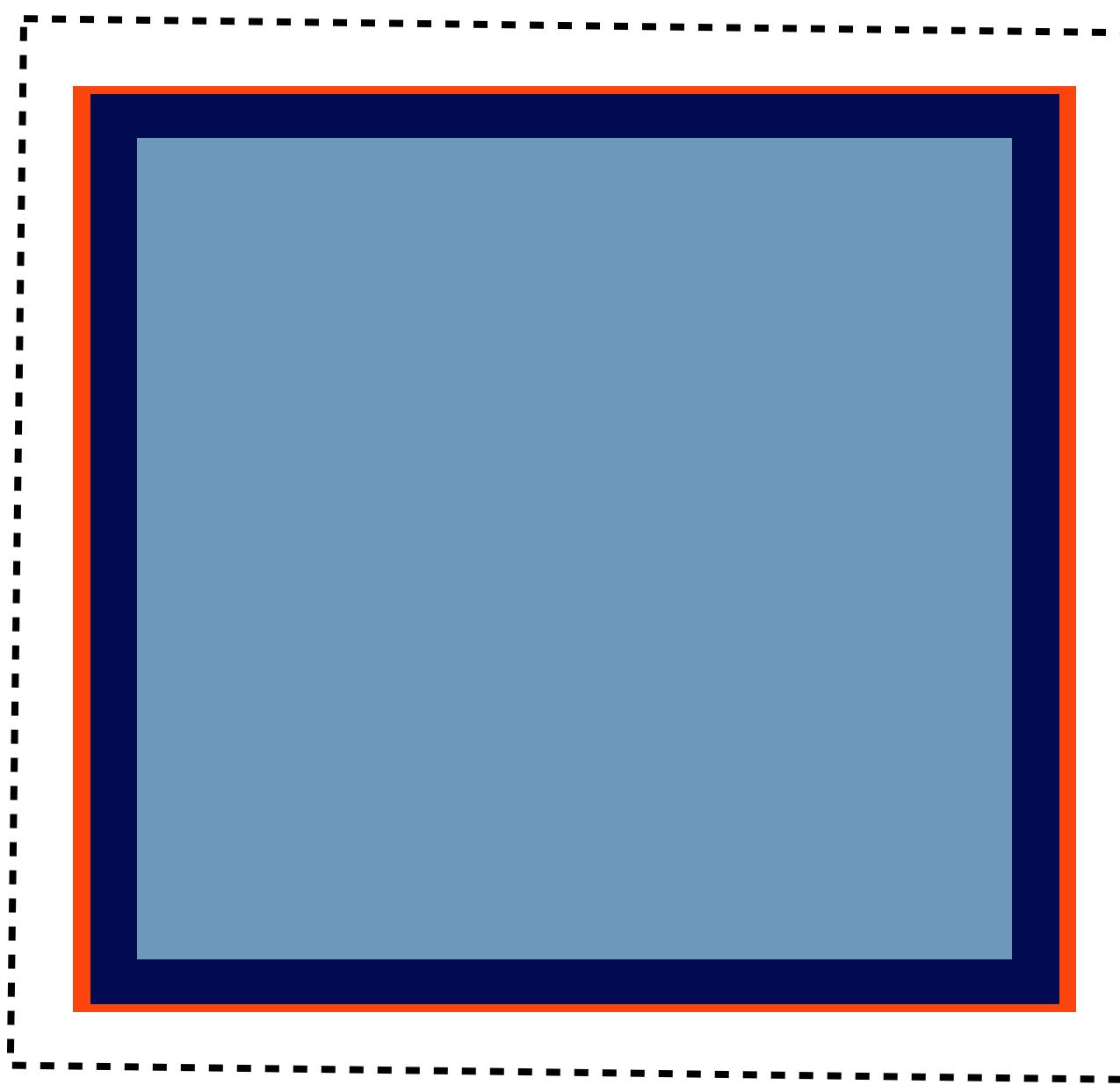
## Herança em CSS

- Quando vários estilos são aplicados a um elemento, eles são herdados
- Uma regra mais específica de correspondência pode substituir uma regra mais geral

### Propriedades não herdadas

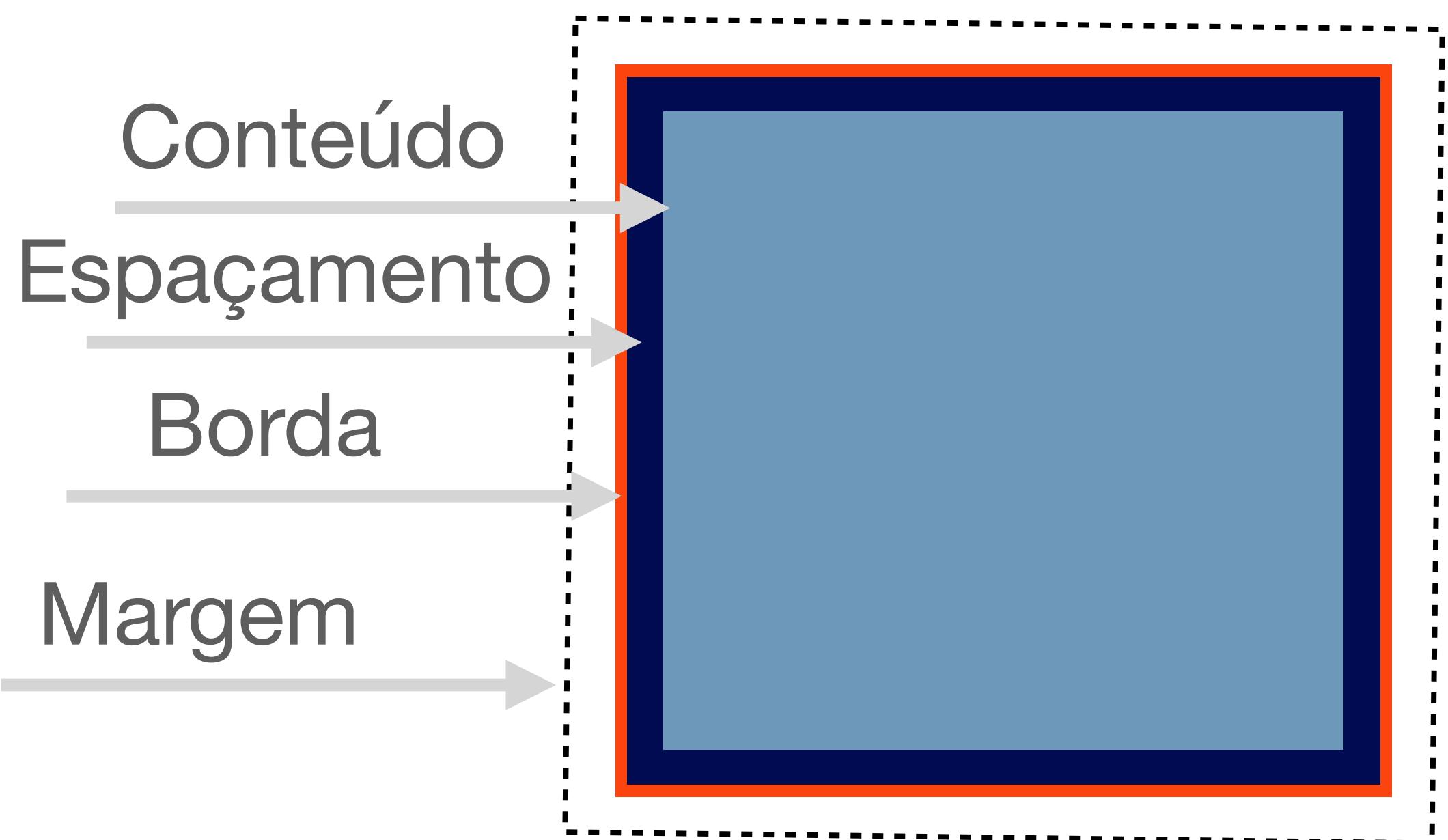
```
background, border (exceto: border-collapse e border-spacing),  
clip, content, counter, cue, display, float, height, left, margin,  
outline, overflow, padding, page-break, pause, play-during, position, right, table-  
layout, text-decoration, top, unicode-bidi, vertical-align, width, z-index.
```

# CSS Box Model



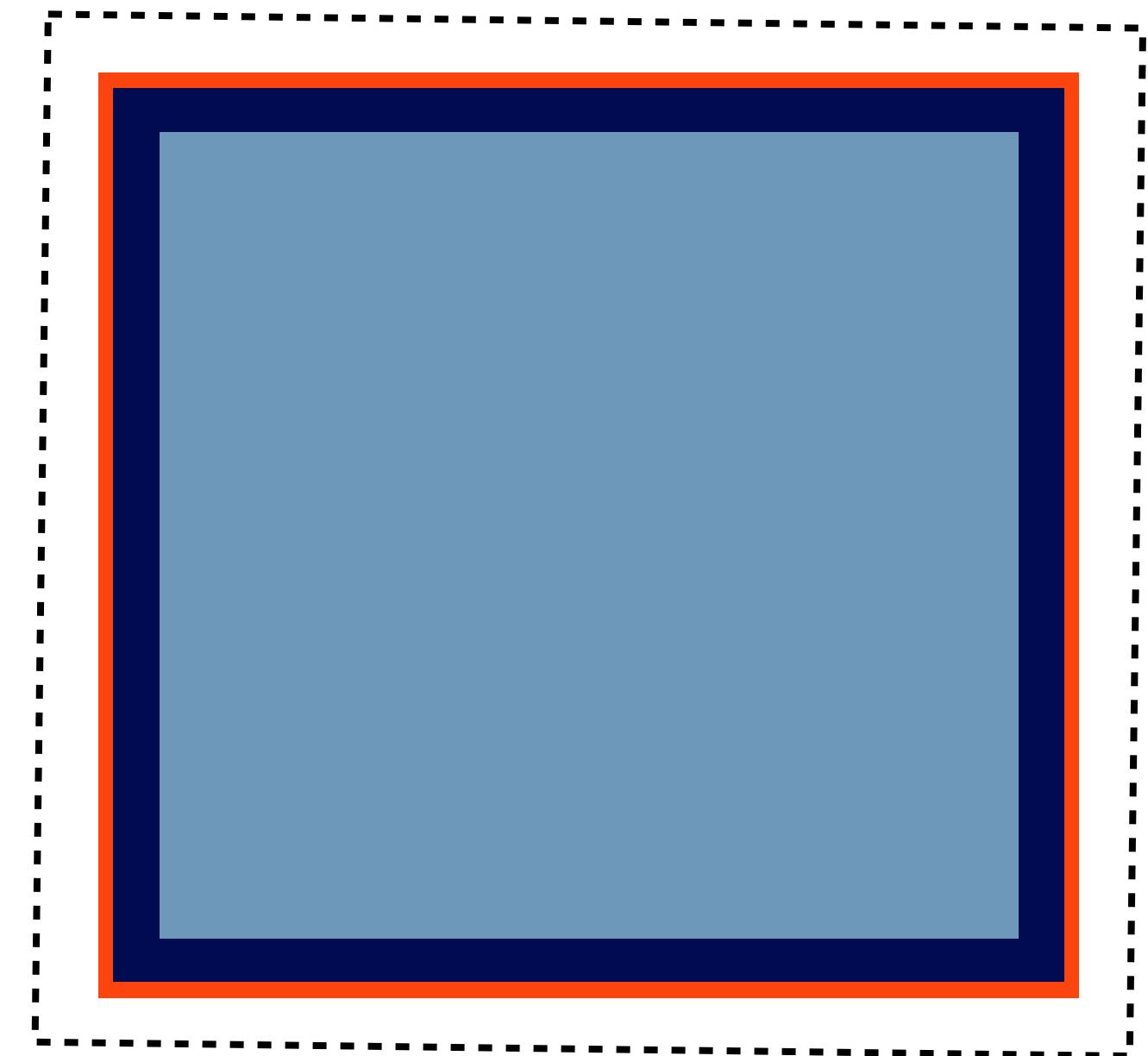
# CSS Box Model

- A caixa é constituída por:
  - Conteúdo:
    - Aquilo que vai ser exibido
  - Borda (*border*):
    - Uma caixa pode possuir até 4 bordas
  - Espaçamento (*padding*):
    - Distância entre a borda e o conteúdo
  - Margem (*margin*):
    - Distância que separa uma caixa de outra



# CSS Box Model

- Margem:
  - São sempre transparentes
  - Cada elemento possui uma margem determinada por padrão
- Borda:
  - São sempre transparentes e de espessura iguais a zero
  - A sua aparência pode ser alterada: tipos, cores e dimensão
- Espaçamento:
  - São sempre transparentes e de espessura iguais a zero



# CSS Box Model

## Dimensionando os elementos

Propriedade	Descrição
width, height	Quão largo ou alto deve ser o elemento <i>(altura não afeta elementos inline)</i>
max-width, max-height,	Tamanho máximo para largura e altura
min-width, min-height	Tamanho mínimo para largura e altura

# CSS Box Model

## Modificando as bordas

Propriedade	Descrição
border-width	Define a espessura da borda
border-style	Define o estilo da borda
border-color	Define a cor da borda
border-{top/right/left/bottom}	Regra abreviada para um lado da borda
border	Regra abreviada para os quatro cantos de borda

# CSS Box Model

## Modificando as bordas

Gerador de Bordas em CSS

# CSS Box Model

## Modificando o espaçamento

Propriedade	Descrição	Valores
padding-top	Define o espaçamento superior	<b>auto</b> : valor padrão <b>dimensão</b> : qualquer medida aceita em CSS %: dimensão relativa a largura do elemento pai
padding-right	Define o espaçamento a direita	
padding-left	Define o espaçamento a esquerda	
padding-bottom	Define o espaçamento inferior	
padding	Regra abreviada para os quatro espaçamentos	

# CSS Box Model

## Modificando o espaçamento

Propriedade	Descrição	Valores
margin-top	Define a margem superior	<b>auto</b> : valor padrão <b>dimensão</b> : qualquer medida aceita em CSS %: dimensão relativa a largura elemento pai
margin-right	Define a margem a direita	
margin-left	Define a margem a esquerda	
margin-bottom	Define a margem inferior	
margin	Regra abreviada para as quatro margens	

# CSS Box Model

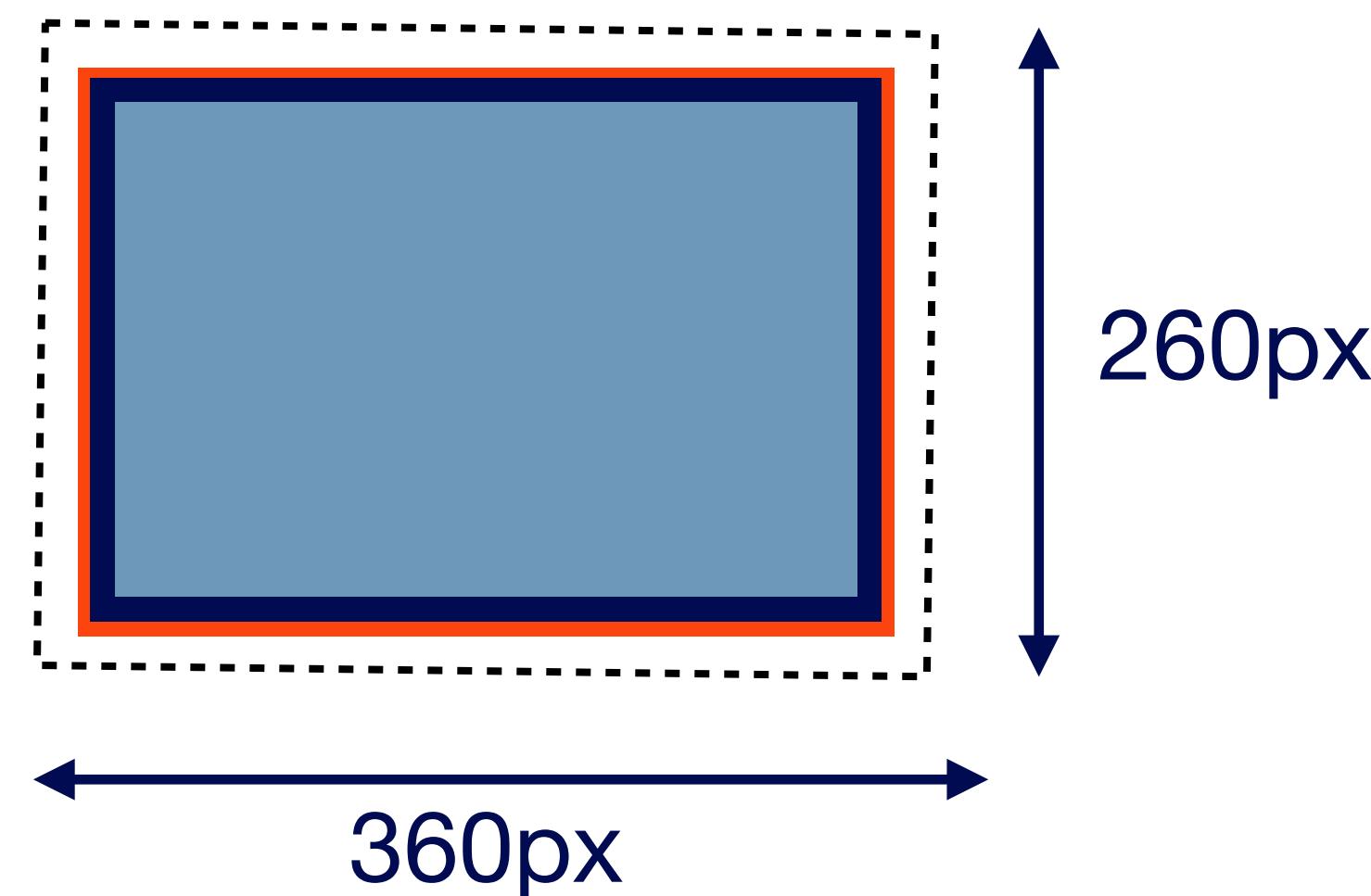
Quanto espaço ocupa uma caixa?

- Largura:
  - $margem_{esq} + borda_{esq} + espacamento_{esq} + largura + espacamento_{dir} + borda_{dir} + margem_{dir}$
- Altura
  - $margem_{sup} + borda_{sup} + espacamento_{sup} + altura + espacamento_{inf} + borda_{inf} + margem_{inf}$

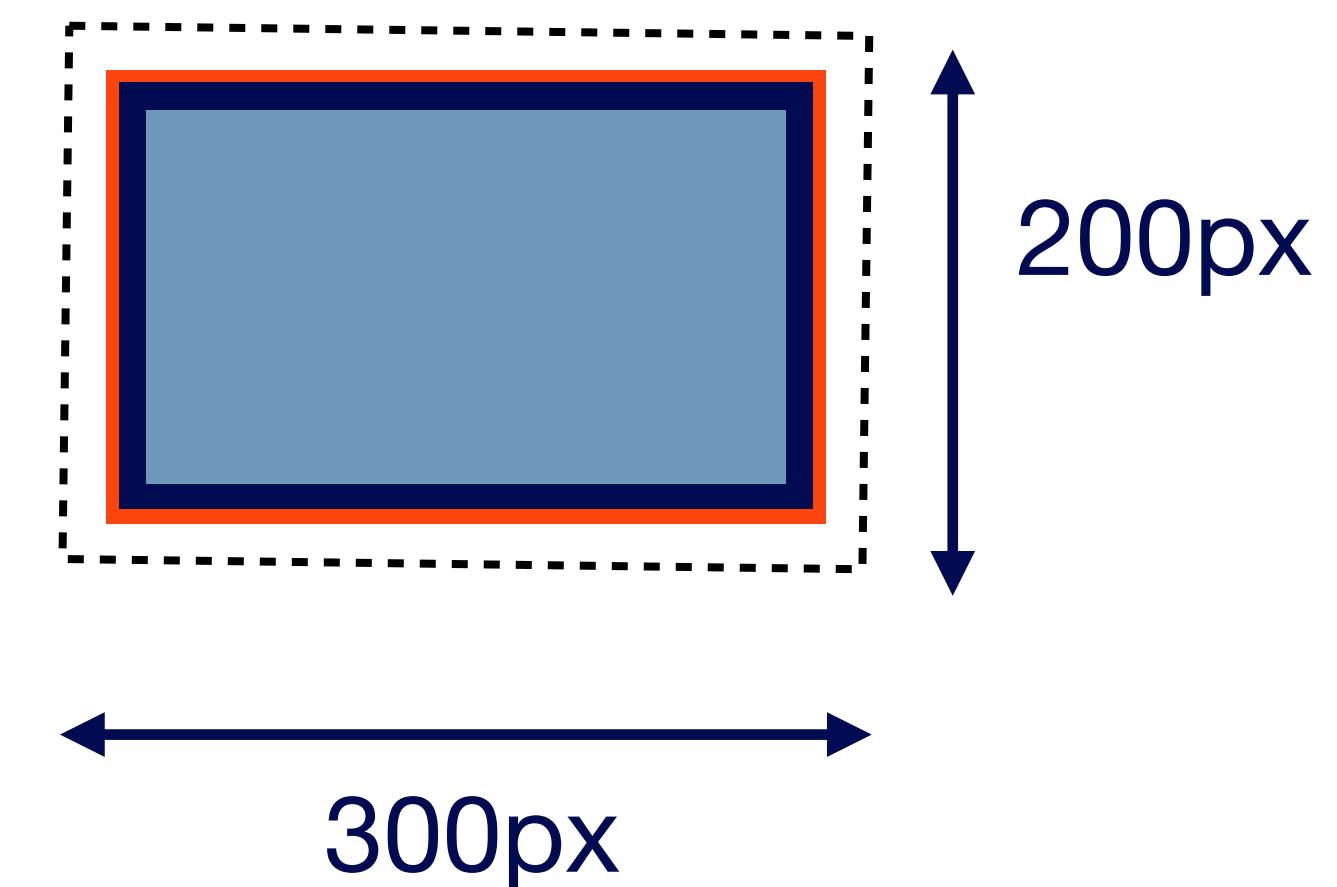
# CSS Box Model

## content-box vs border-box

```
.box {  
    box-sizing: content-box;  
    width: 300px;  
    height: 200px;  
    padding: 10px;  
    border: 5px solid #FC440F;  
    margin: 15px;  
}
```



```
.box {  
    box-sizing: border-box;  
    width: 300px;  
    height: 200px;  
    padding: 10px;  
    border: 5px solid #FC440F;  
    margin: 15px;  
}
```



# Posicionamiento

STATIC

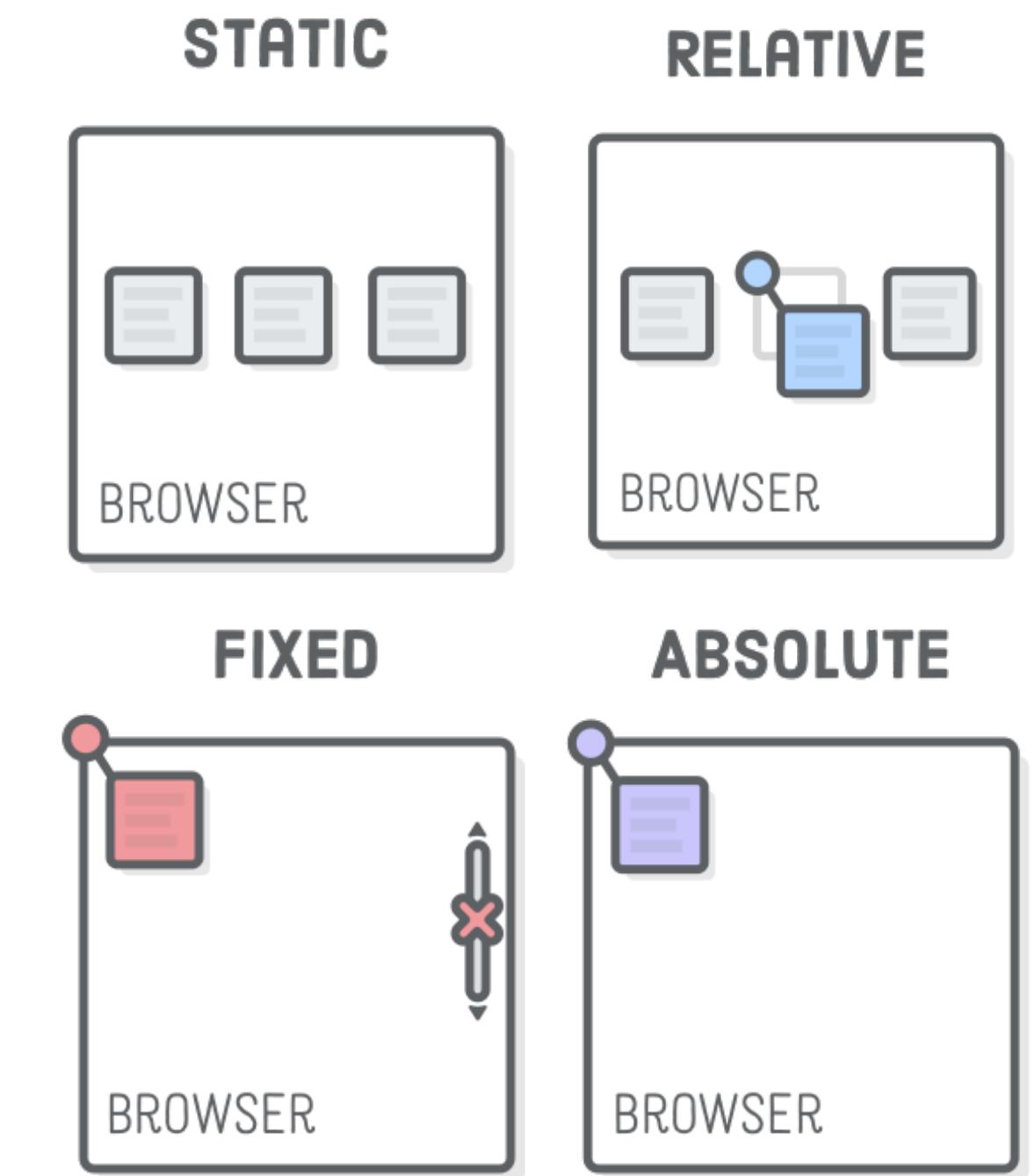


BROWSER

# Posicionamento

- Para possibilitar a construção de **layouts complexos** precisamos ter conhecimento sobre a propriedade **position**

Propriedade	Descrição
static	Padrão
relative	Calcula a posição em relação ao canto superior esquerdo da sua posição normal ( <b>static</b> )
fixed	Define a posição em relação a janela do navegador
absolute	Calcula a posição do elemento em relação ao elemento superior do primeiro ancestral <b>não static</b>

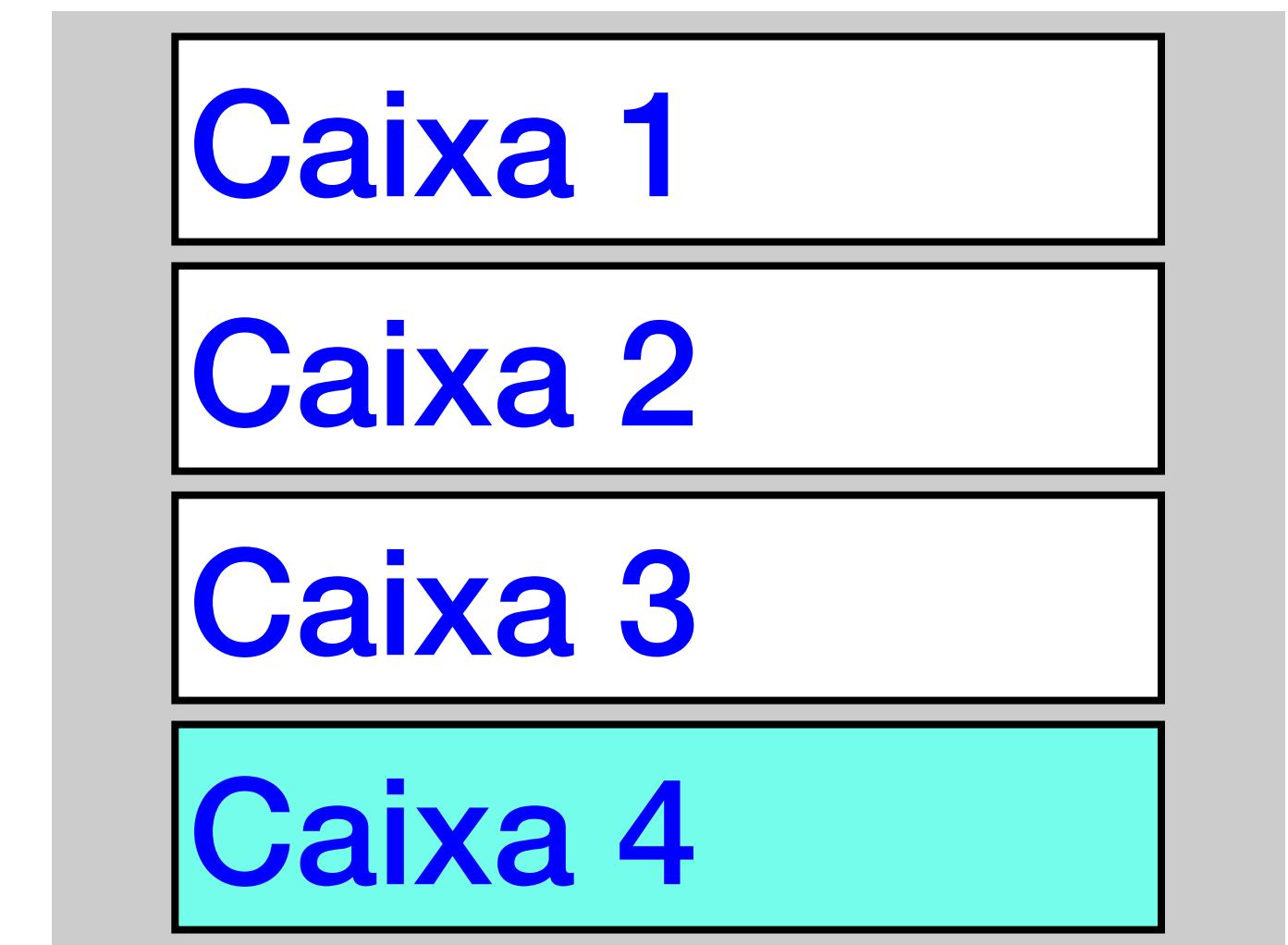
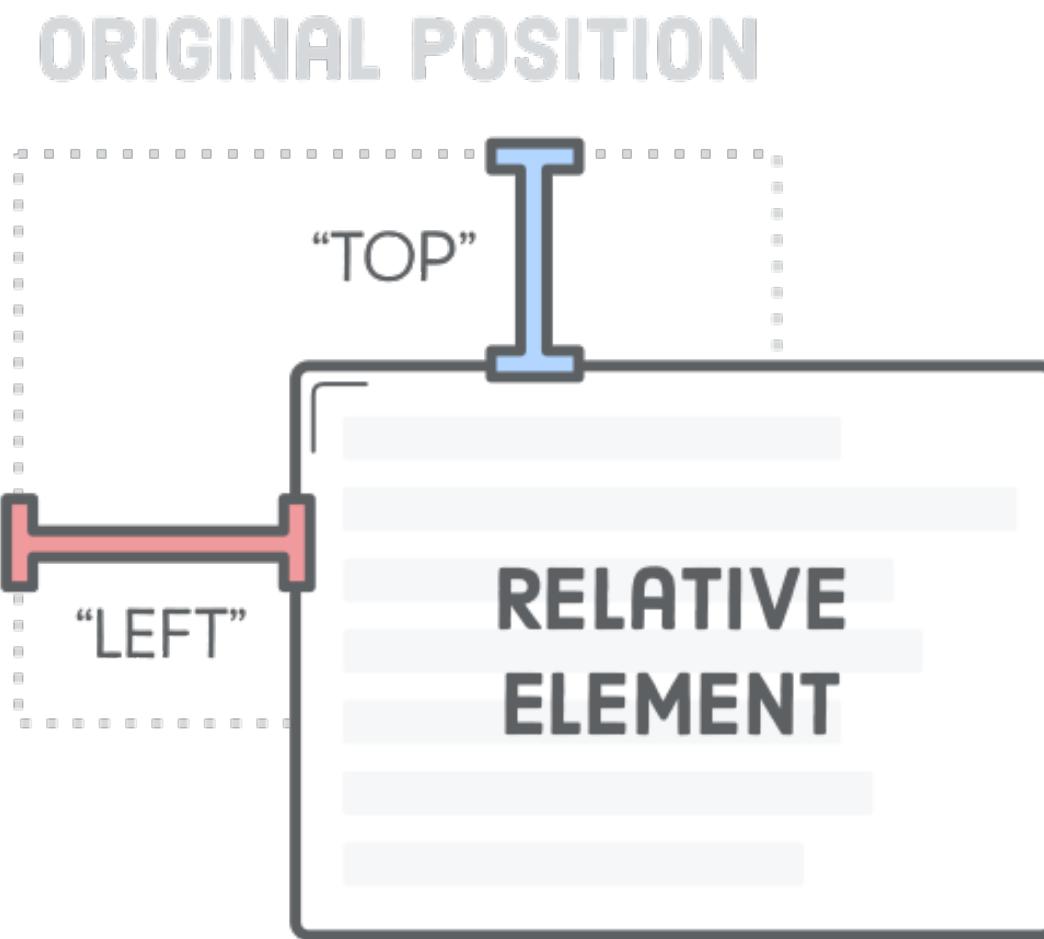


# Posicionamento

## static vs relative

```
.parent {  
    color: blue;  
    background-color: gray;  
    width: 500px;  
    height: 500px;  
}  
  
.special.box {  
    position: relative;  
    top: 20px;  
    left: 20px;  
    background-color: cyan;  
}  
  
.box {  
    font-size: 2em;  
    background-color: white;  
    width: 400px;  
    border: 1px solid black;  
    margin: 10px auto;  
    box-sizing: border-box;  
}
```

```
<div class="parent">  
    <div class="box">Caixa 1</div>  
    <div class="box">Caixa 2</div>  
    <div class="box">Caixa 3</div>  
    <div class="box special">Caixa 4</div>  
</div>
```

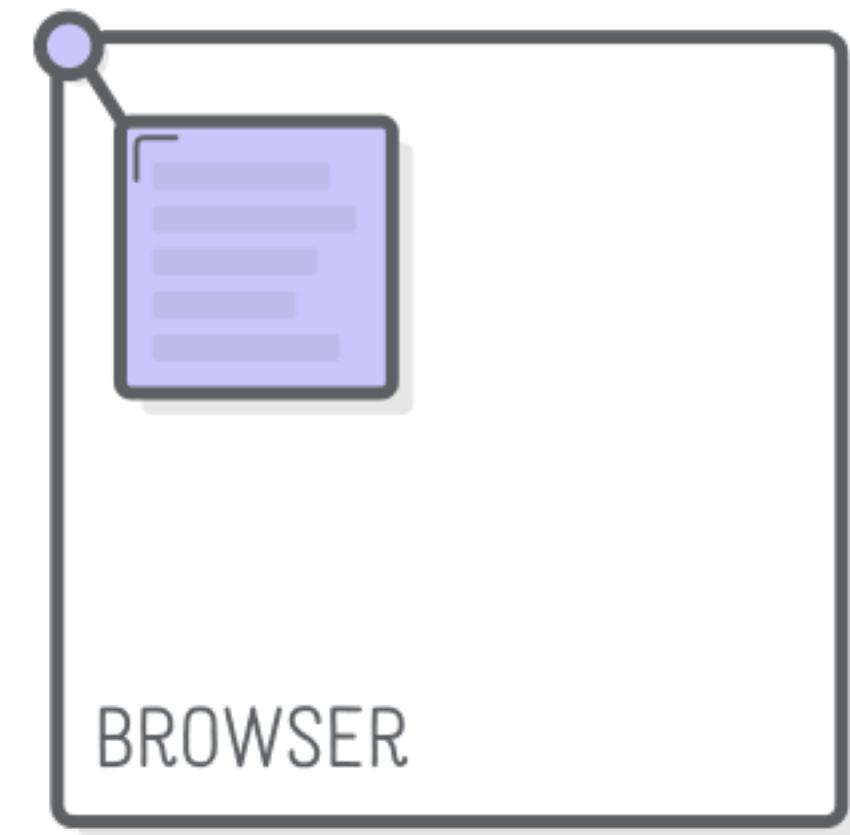


# Posicionamento

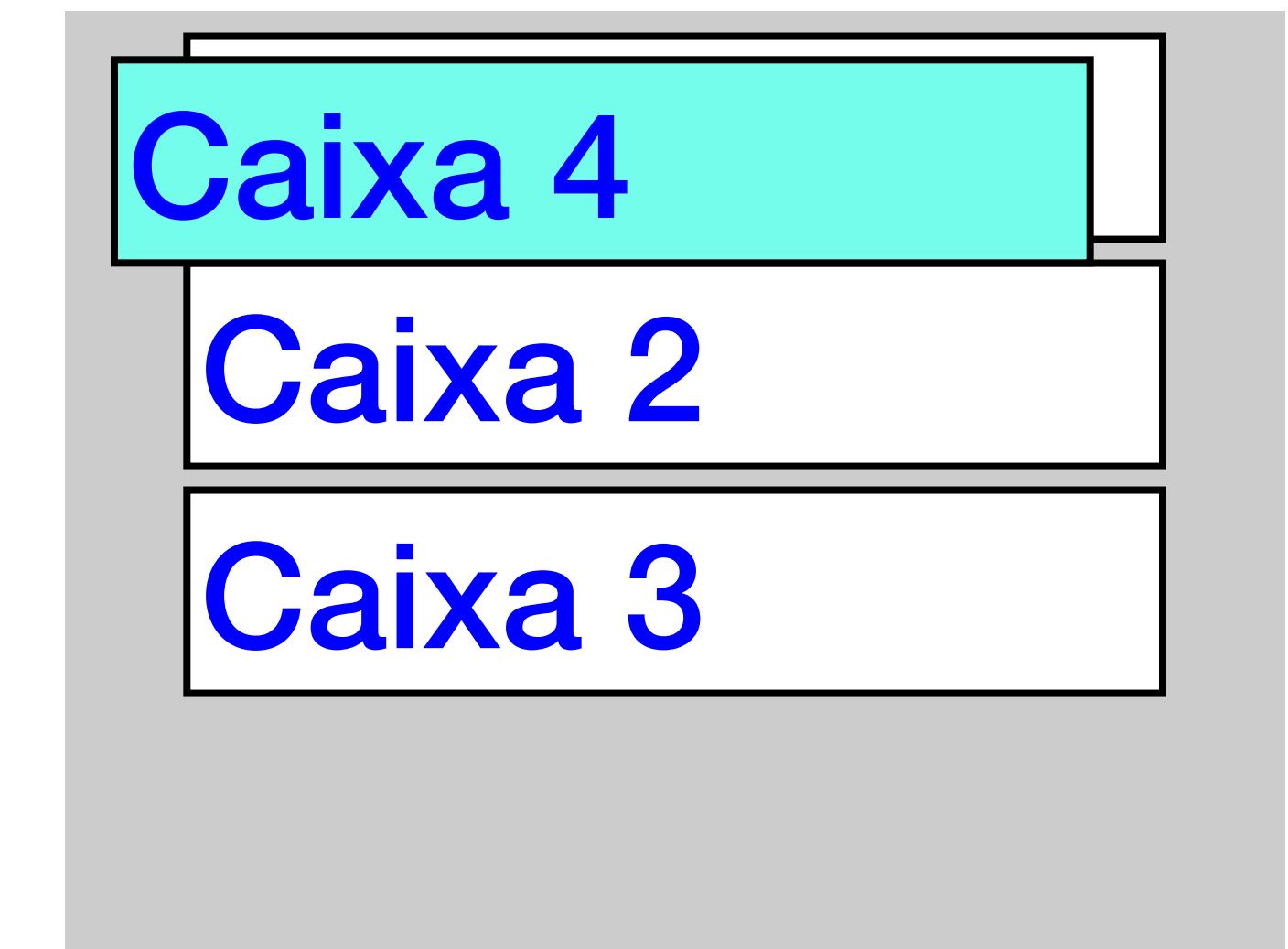
## Absolute

```
.parent {  
    color: blue;  
    background-color: gray;  
    width: 500px;  
    height: 500px;  
}  
  
.special.box {  
    position: absolute;  
    top: 20px;  
    left: 20px;  
    background-color: cyan;  
}  
  
.box {  
    font-size: 2em;  
    background-color: white;  
    width: 400px;  
    border: 1px solid black;  
    margin: 10px auto;  
    box-sizing: border-box;  
}
```

```
<div class="parent">  
    <div class="box">Caixa 1</div>  
    <div class="box">Caixa 2</div>  
    <div class="box">Caixa 3</div>  
    <div class="box special">Caixa 4</div>  
</div>
```



ABSOLUTE POSITIONING



# Posicionamento

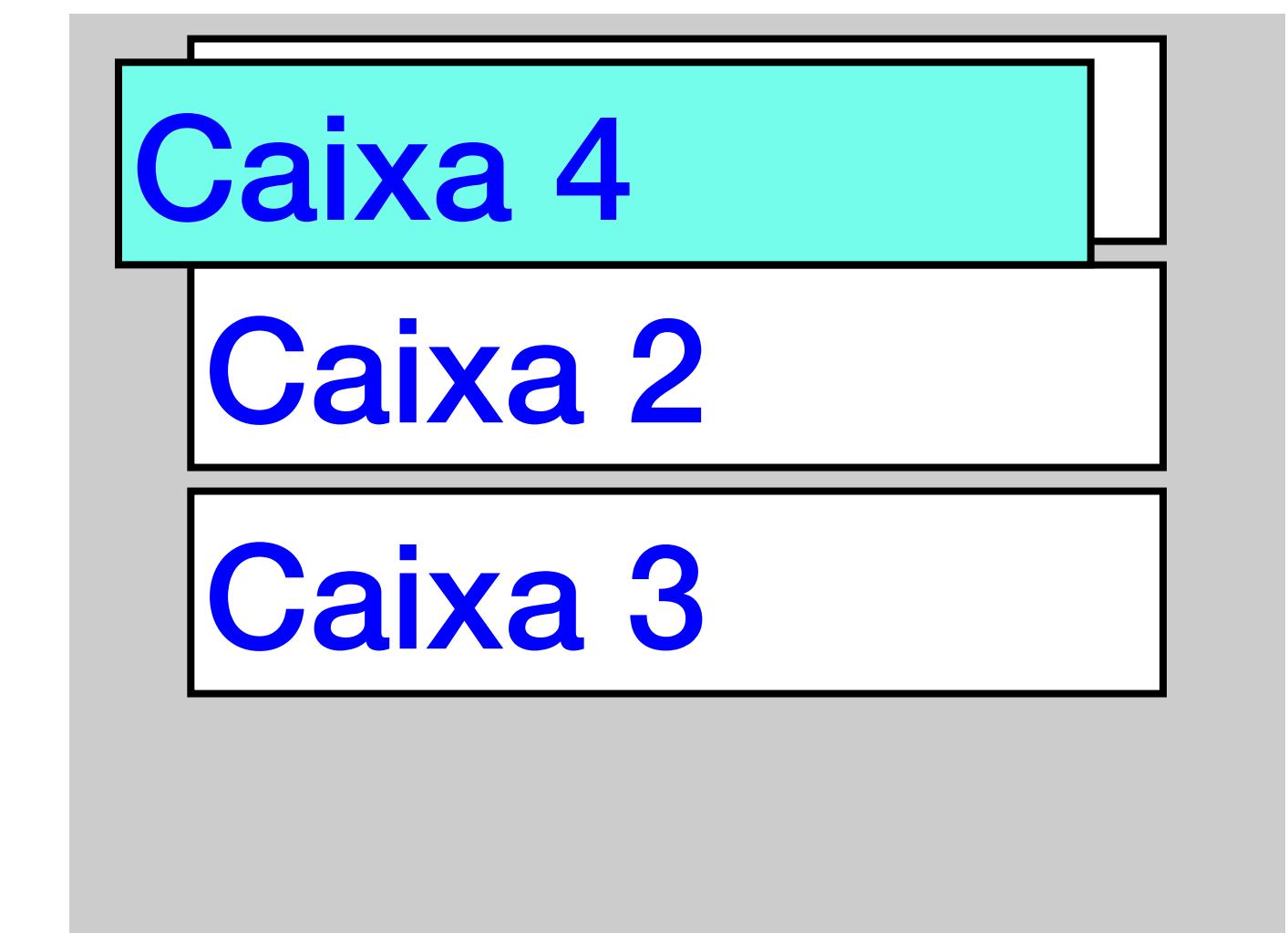
## fixed

```
.parent {  
    color: blue;  
    background-color: gray;  
    width: 500px;  
    height: 500px;  
}  
  
.special.box {  
    position: fixed;  
    top: 20px;  
    left: 20px;  
    background-color: cyan;  
}  
  
.box {  
    font-size: 2em;  
    background-color: white;  
    width: 400px;  
    border: 1px solid black;  
    margin: 10px auto;  
    box-sizing: border-box;  
}
```

```
<div class="parent">  
    <div class="box">Caixa 1</div>  
    <div class="box">Caixa 2</div>  
    <div class="box">Caixa 3</div>  
    <div class="box special">Caixa 4</div>  
</div>
```

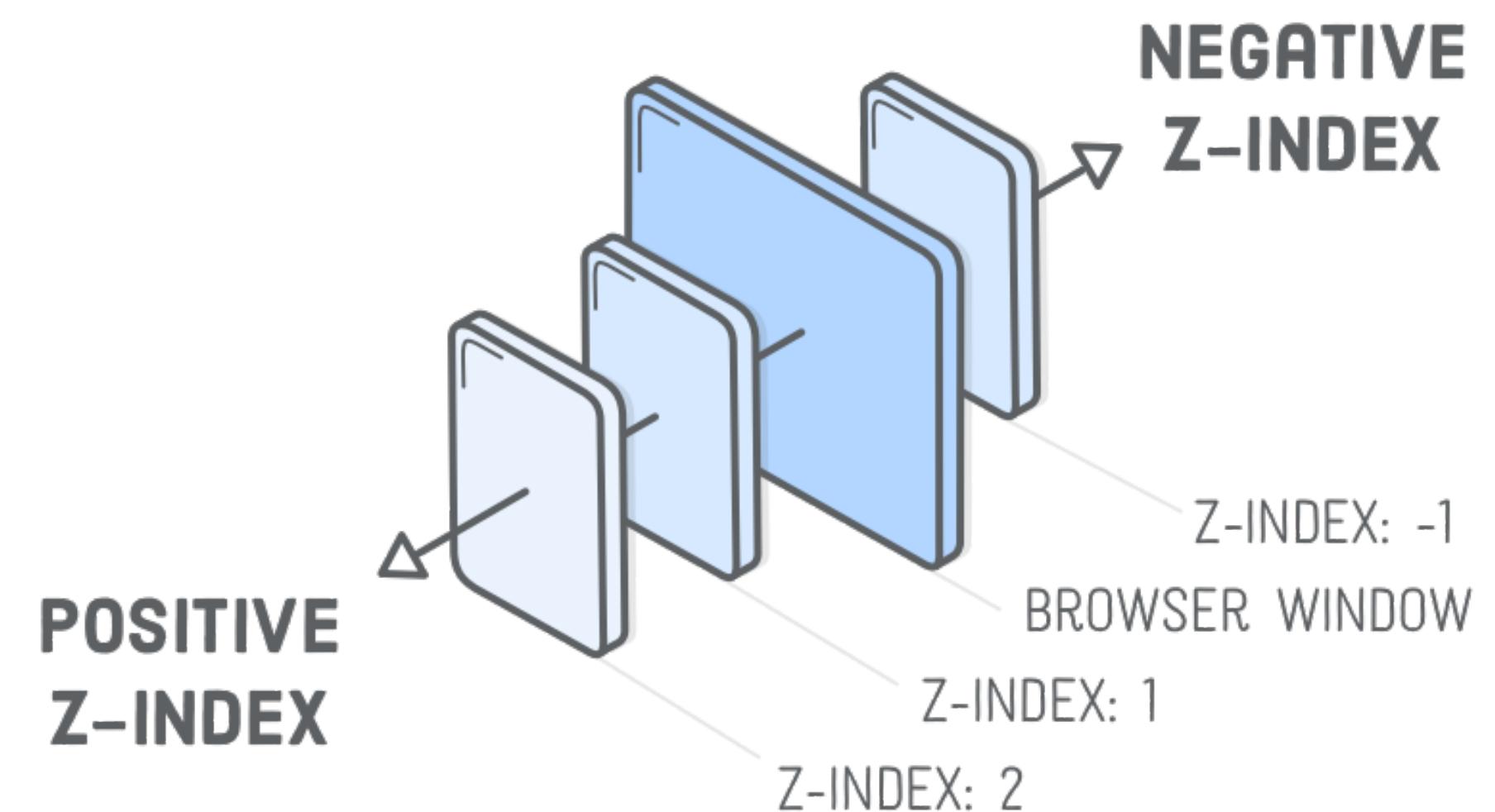


FIXED POSITIONING

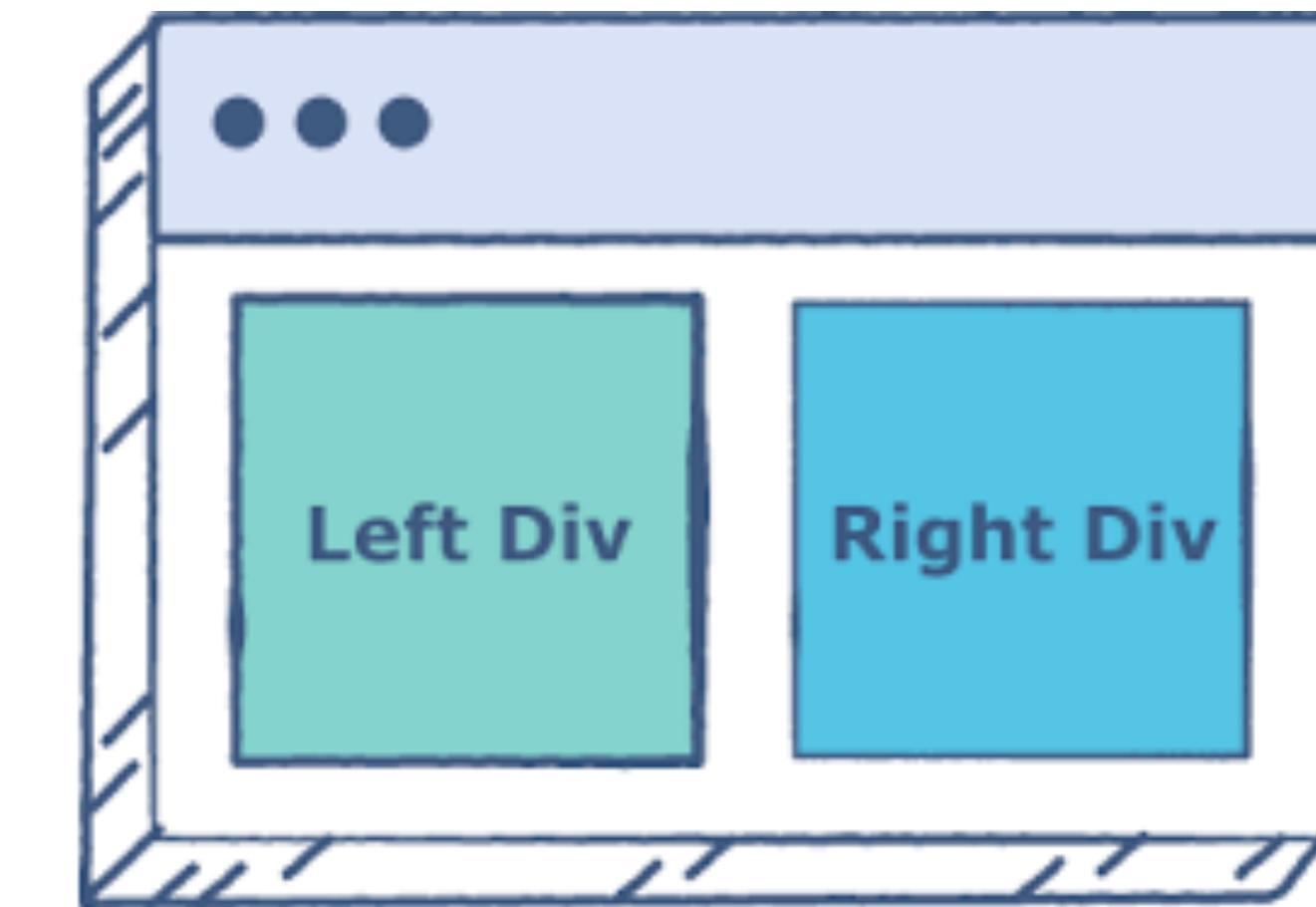


# Posicionamento

- Elementos que usam **relative**, **absolute** e **fixed** são também chamados *positioned elements*
- Único que analisam a propriedade **z-index**
- O posicionamento absolute e fixed não obedecem o **fluxo normal** de posicionamento
- Exemplo prático completo:
  - <https://www.internetingishard.com/html-and-css/advanced-positioning/>



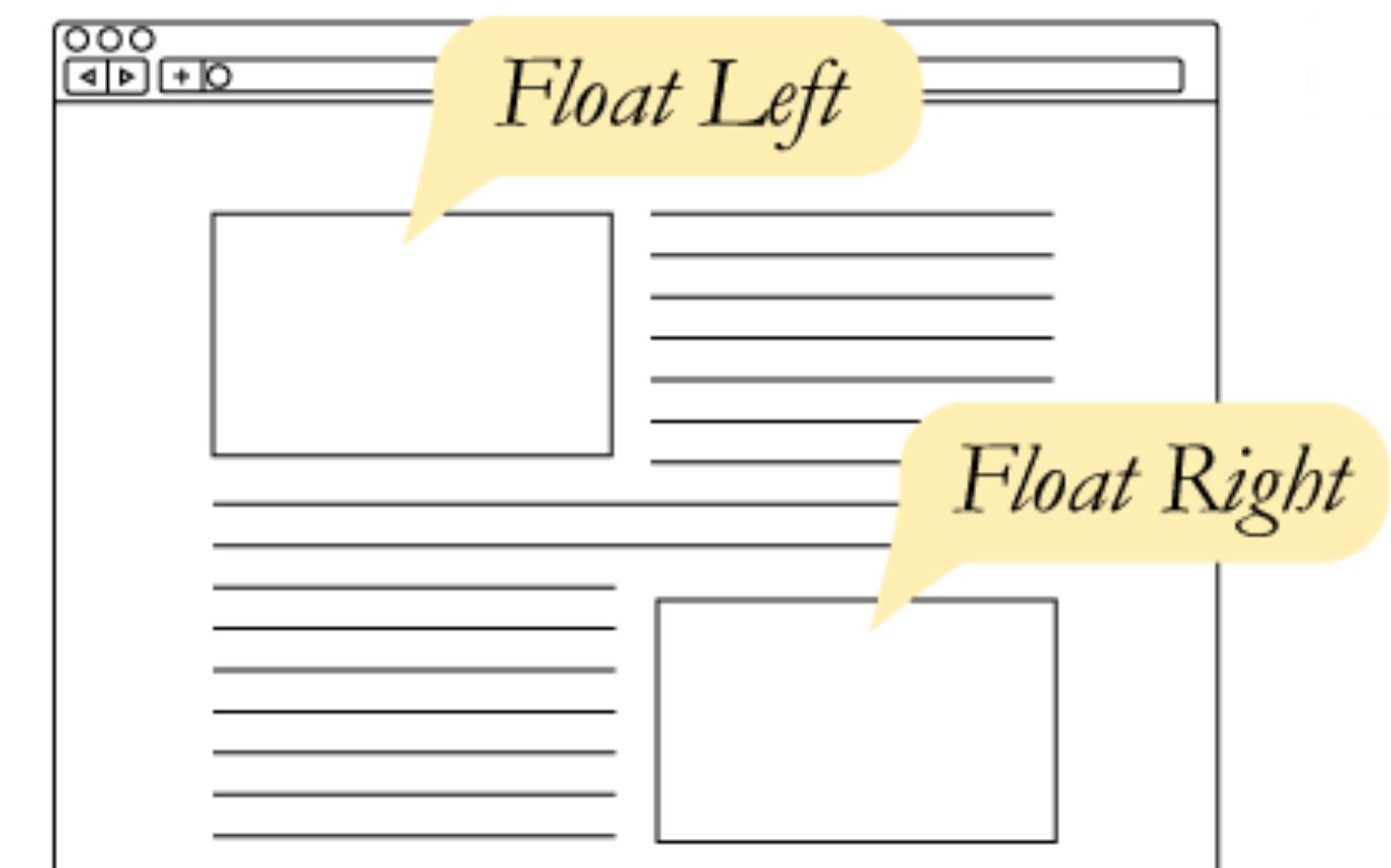
# Elementos flutuantes



# Elementos flutuantes

## Elemento flutuante

- Determinado a partir da propriedade **float**
  - Não segue o fluxo normal de posicionamento
  - O text adjacente o envolve da maneira



# Elementos flutuantes

## Float

```
p {  
    border-color: 1px solid black;  
    background-color: gray;  
    margin: 0;  
}  
  
#content {  
    float: none;  
    text-align: right;  
    border-color: 1px solid red;  
    background-color: purple;  
    color: white;  
}
```

```
<p id="content">Lorem ipsum dolor sit amet,  
consectetur adipiscing elit fusce.</p>  
<p id="content2">Bla bla bla bla bla bla  
bla bla bla  
Bla bla bla bla bla bla bla bla bla Bla  
bla ... </p>
```

**Lore  
m ipsum dolor sit amet,  
consectetur adipiscing elit fusce.**

**Bla bla bla bla bla bla bla bla  
Bla bla bla bla bla bla bla bla Bla bla Bla bla  
bla bla bla bla bla bla bla**

# Elementos flutuantes

## Float

```
p {  
    border-color: 1px solid black;  
    background-color: gray;  
    margin: 0;  
}  
  
#content {  
    float:none;  
    text-align:right;  
    border-color: 1px solid red;  
    background-color: purple;  
    color: white;  
    width: 50%;  
}
```

```
<p id="content">Lorem ipsum dolor sit amet,  
consectetur adipiscing elit fusce.</p>  
<p id="content2">Bla bla bla bla bla bla  
bla bla bla  
Bla bla bla bla bla bla bla bla Bla  
bla ... </p>
```

**Lore  
m ipsum dolor sit  
amet,  
consectetur adipiscin  
g  
elit fusce.**

Bla bla bla bla bla bla bla bla  
Bla bla bla bla bla bla bla Bla bla Bla bla  
bla bla bla bla bla bla bla

# Elementos flutuantes

## Float

```
p {  
    border-color: 1px solid black;  
    background-color: gray;  
    margin: 0;  
}  
  
#content {  
    float:left;  
    text-align:left;  
    border-color: 1px solid red;  
    background-color: purple;  
    color: white;  
    width: 50%;  
}
```

```
<p id="content">Lorem ipsum dolor sit amet,  
consectetur adipiscing elit fusce.</p>  
<p id="content2">Bla bla bla bla bla bla  
bla bla bla  
Bla bla bla bla bla bla bla bla Bla  
bla ... </p>
```

**Lorem ipsum dolor sit  
amet,  
consectetur adipiscing  
elit fusce.  
bla bla bla**

**Bla bla bla bla bla bla bla  
bla bla Bla bla bla bla bla  
bla bla bla bla Bla bla  
bla bla bla bla bla bla bla**

# Elementos flutuantes

## Float

```
p {  
    border-color: 1px solid black;  
    background-color: gray;  
    margin: 0;  
}  
  
#content {  
    float: left;  
    text-align: center;  
    border-color: 1px solid red;  
    background-color: purple;  
    color: white;  
    width: 50%;  
}
```

```
<p id="content">Lorem ipsum dolor sit amet,  
consectetur adipiscing elit fusce.</p>  
<p id="content2">Bla bla bla bla bla bla  
bla bla bla  
Bla bla bla bla bla bla bla bla Bla  
bla ... </p>
```

**Lore** ipsum dolor sit  
amet,  
consectetur adipiscing  
elit fusce.  
bla bla bla

Bla bla bla bla bla bla bla  
bla bla Bla bla bla bla bla  
bla bla bla bla Bla bla  
bla bla bla bla bla bla bla

# Elementos flutuantes

## clear

- Propriedade irmã da propriedade float
- Desabilita a sobreposição de um elemento flutuante sobre esse elemento
  - left, right, both ou none

# Elementos flutuantes

## Clear

```
p {  
background-color: aqua;  
border: 1px solid red;  
}  
  
img { float: right; }  
  
h5 {  
clear: none;  
border: 6px solid blue;  
background-color: silver;  
}
```

<p> Sonic é um ouriço azul bastante veloz</p>  
  
<h5> Sonic fã site</h5>

Sonic é um ouriço bastante veloz

Sonic fã site



# Elementos flutuantes

## Clear

```
p {  
background-color: aqua;  
border: 1px solid red;  
}  
  
img { float: right; }  
  
h5 {  
clear: right;  
border: 6px solid blue;  
background-color: silver;  
}
```

<p> Sonic é um ouriço azul bastante veloz</p>  
  
<h5> Sonic fã site</h5>

Sonic é um ouriço bastante veloz



Sonic fã site

# Elementos flutuantes

## overflow

- Utilizada quando o elemento flutuante é maior que a área ocupada pelo conteúdo do seu elemento pai
- Especifica o que deve acontecer se um elemento filho for muito grande
  - **hidden** ou **visible**

# Elementos flutuantes

## Overflow

```
p {  
background-color: aqua;  
border: 1px solid red;  
overflow: visible;  
}  
  
img { float: right; }
```

```
<p> Sonic é um  
ouriço azul bastante veloz</p>
```

**Sonic é um ouriço bastante veloz**



# Elementos flutuantes

## Overflow

```
p {  
background-color: aqua;  
border: 1px solid red;  
overflow: visible;  
height: 111px;  
}  
  
img { float: right; }
```

```
<p> Sonic é um  
ouriço azul bastante veloz</p>
```

**Sonic é um ouriço bastante veloz**



# Elementos flutuantes

## Overflow

```
p {  
background-color: aqua;  
border: 1px solid red;  
overflow: hidden;  
  
}  
  
img { float: right; }
```

```
<p> Sonic é um  
ouriço azul bastante veloz</p>
```

**Sonic é um ouriço bastante veloz**



# Elementos flutuantes

## Overflow

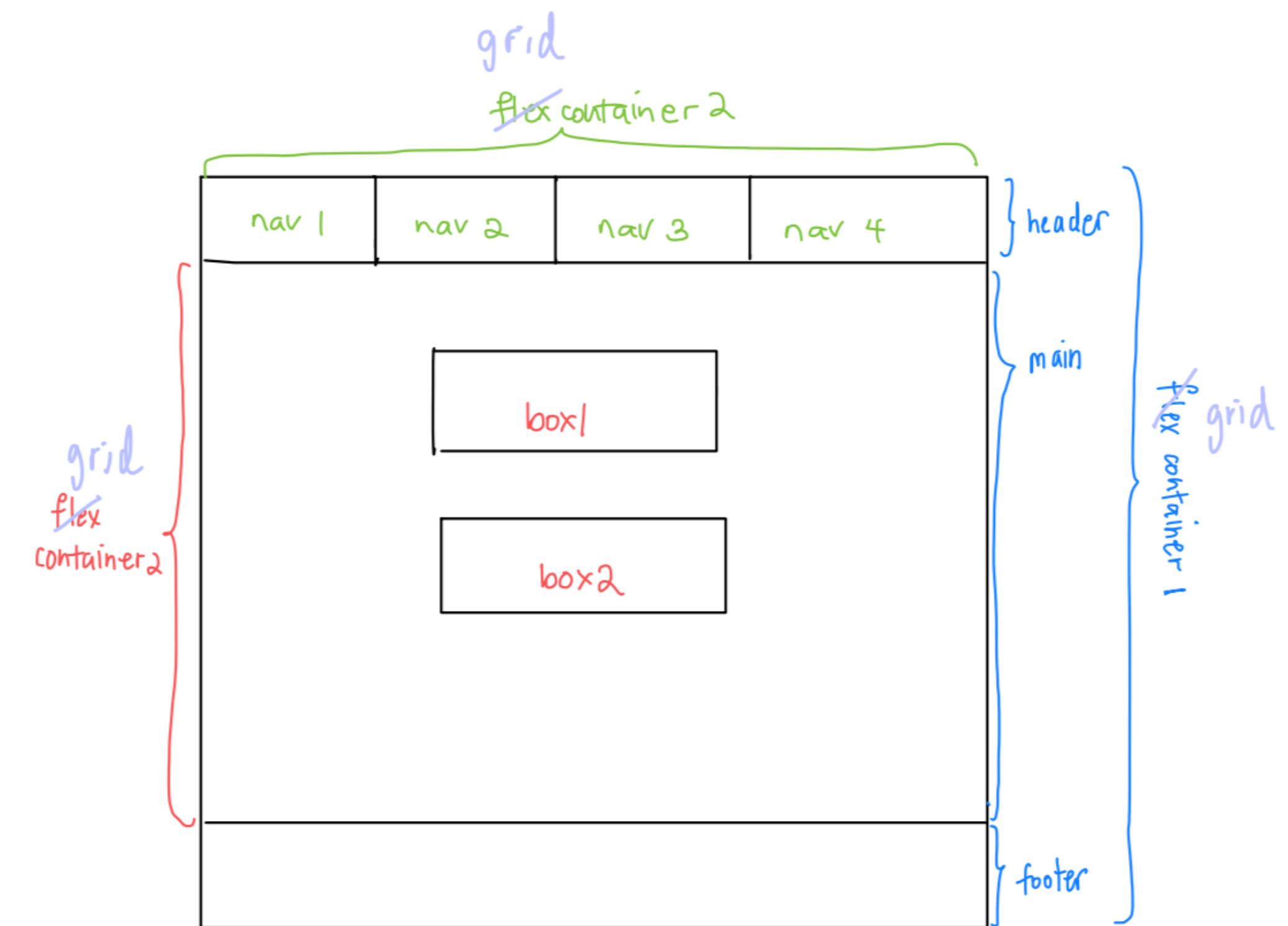
```
p {  
background-color: aqua;  
border: 1px solid red;  
overflow: hidden;  
height: 111px;  
}  
  
img { float: right; }
```

```
<p> Sonic é um  
ouriço azul bastante veloz</p>
```

**Sonic é um ouriço bastante veloz**



# Layouts



# Layouts

## Construindo o layout do seu site

- Uma maneira de pensar sobre o layout é imaginar as divisões da página
  - Até o **HTML 4**, isso era feito utilizando a tag **div**
  - Onde cada **div** deveria ser identificada com um nome significativo
- No **HTML 5** existem tags mais apropriadas

# Layouts

## Construindo layouts

```
<body>
  <div id="header">
    </div>

    <div id="nav">
    </div>

    <div id="section">
    </div>

    <div id="footer">
    </div>
</body>
```

HTML4 → HTML5

```
<body>
  <header>
    </header>

    <nav>
    </nav>

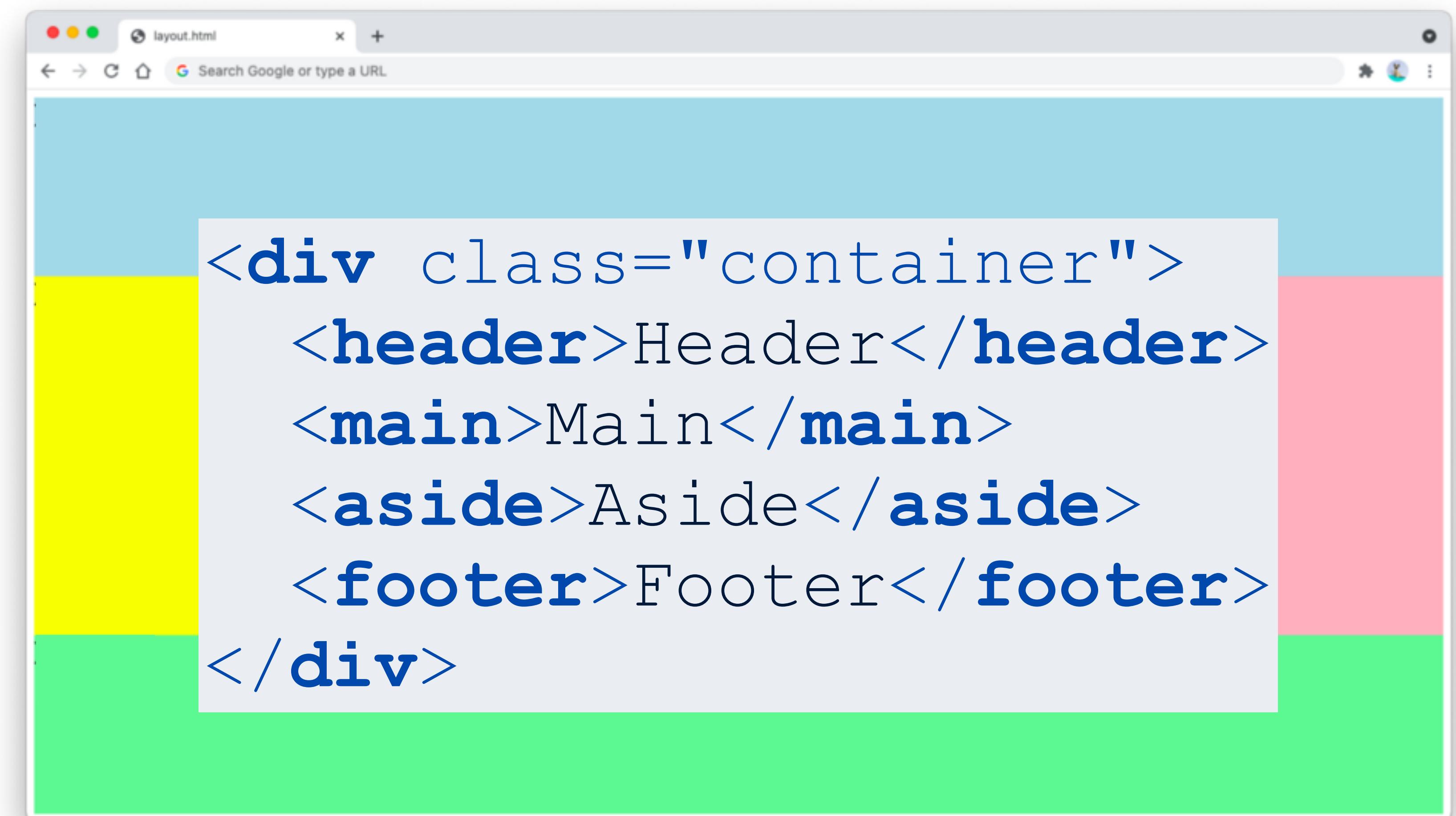
    <section>
    </section>

    <footer>
    </footer>
</body>
```

# Layouts

## Construindo layouts

1. Escolher os contêineres do layout
2. Nomeá-los
3. Escrever HTML
4. Preencher com conteúdo mínimo
5. Posicioná-los



# Layouts

## Layouts responsivos

Layouts responsivo vêm da ideia que um website deve se adaptar a diferentes condições em especial em dispositivos com diferentes resoluções.

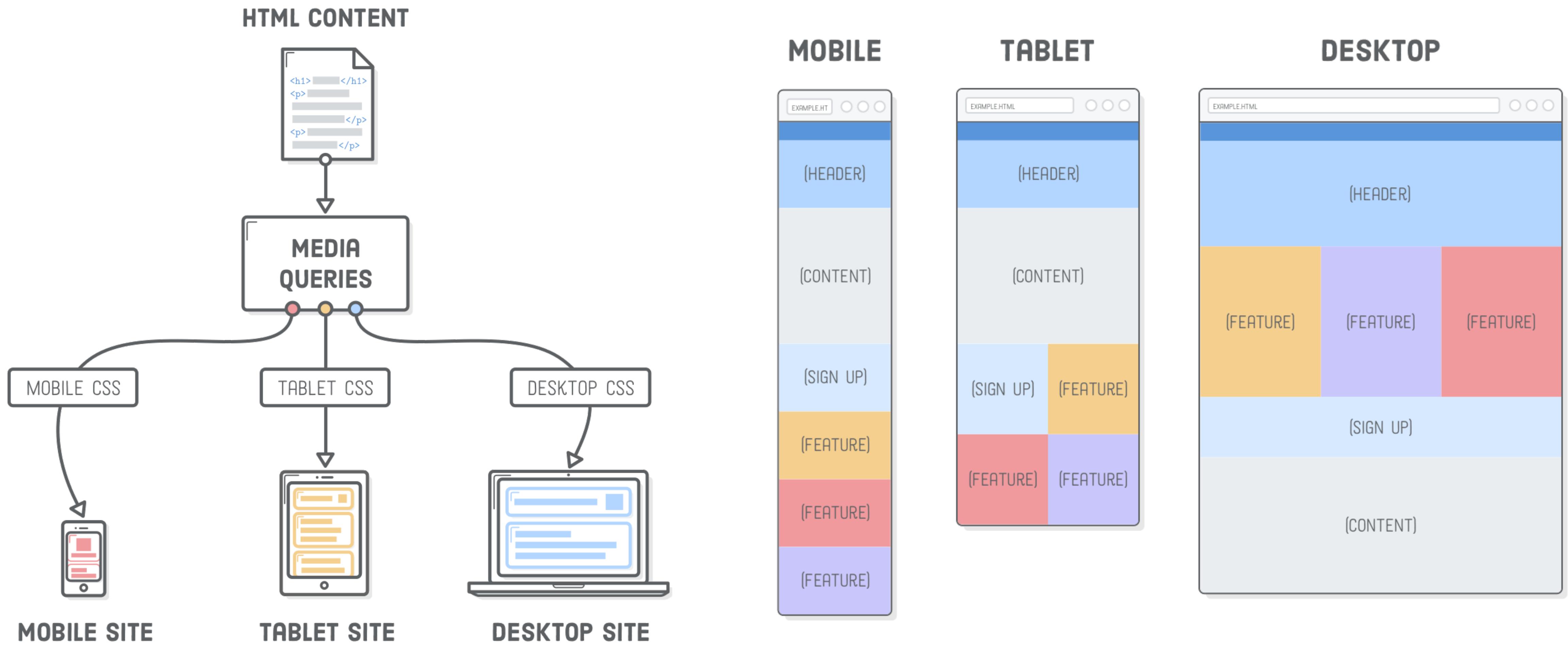
# Como ?

Resoluções:

- Media queries
- Layout fluído
- Imagens flexíveis

# Layouts

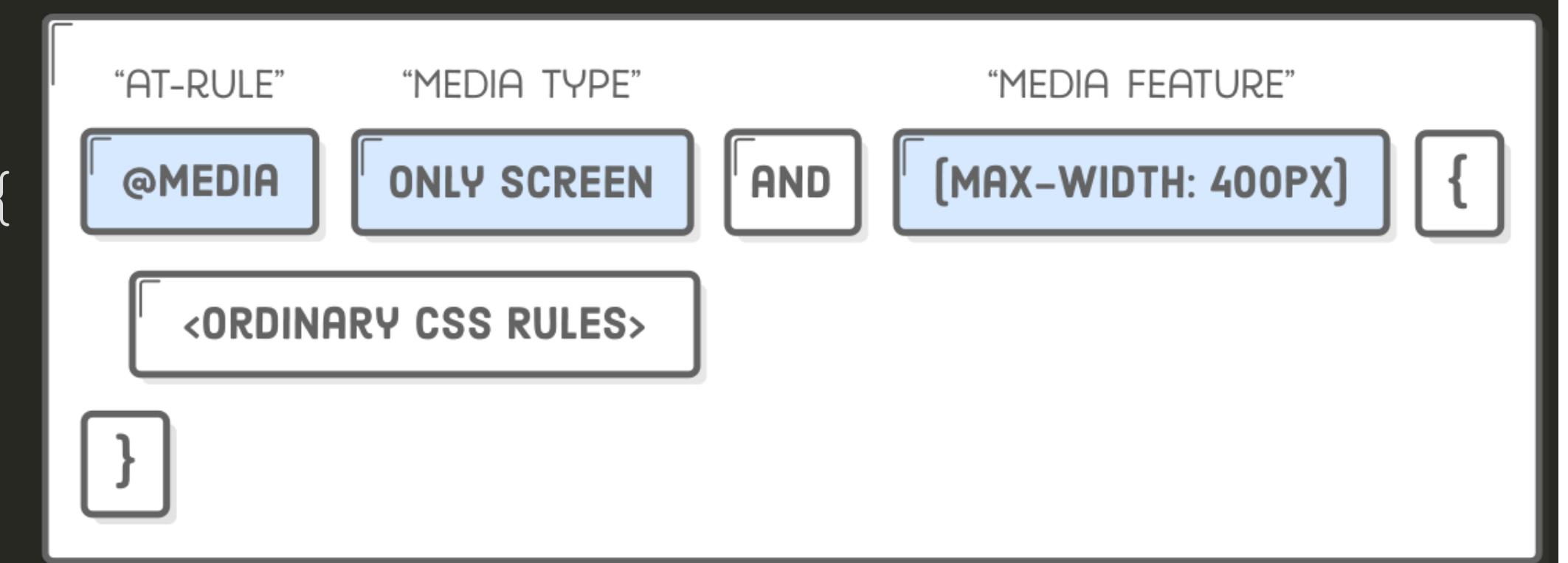
## Media queries



# Layouts

## Media queries

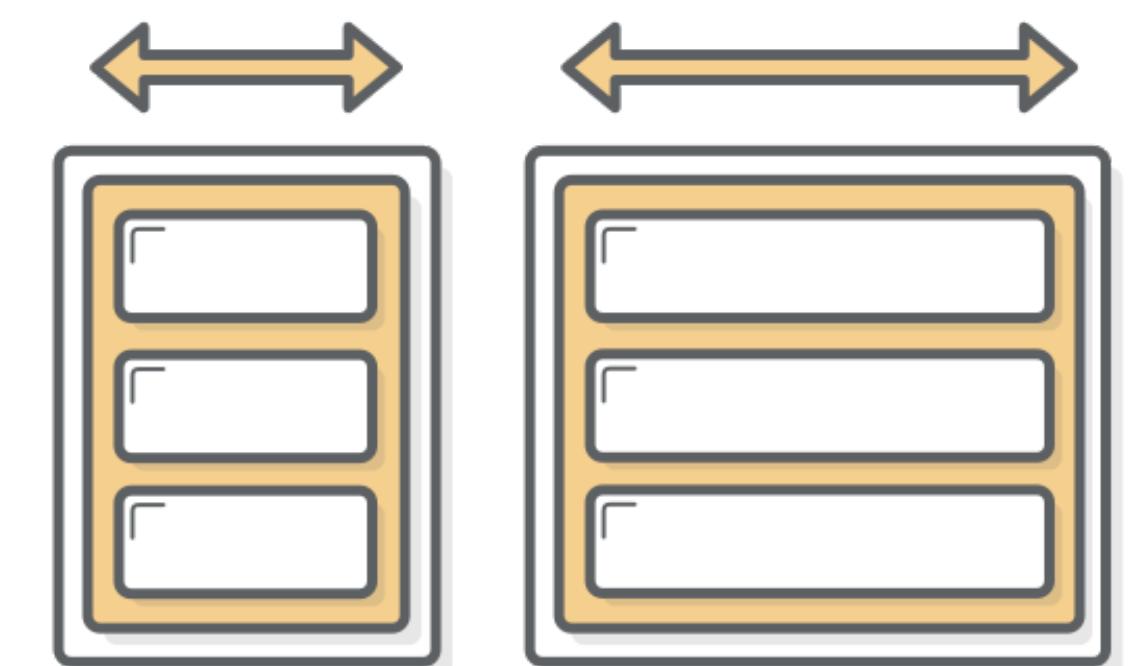
```
/* Mobile Styles */  
@media only screen and (max-width: 400px) {  
    body {  
        background-color: #F09A9D; /* Red */  
    }  
}  
  
/* Tablet Styles */  
@media only screen and (min-width: 401px) and (max-width: 960px) {  
    body {  
        background-color: #F5CF8E; /* Yellow */  
    }  
}  
  
/* Desktop Styles */  
@media only screen and (min-width: 961px) {  
    body {  
        background-color: #B2D6FF; /* Blue */  
    }  
}
```



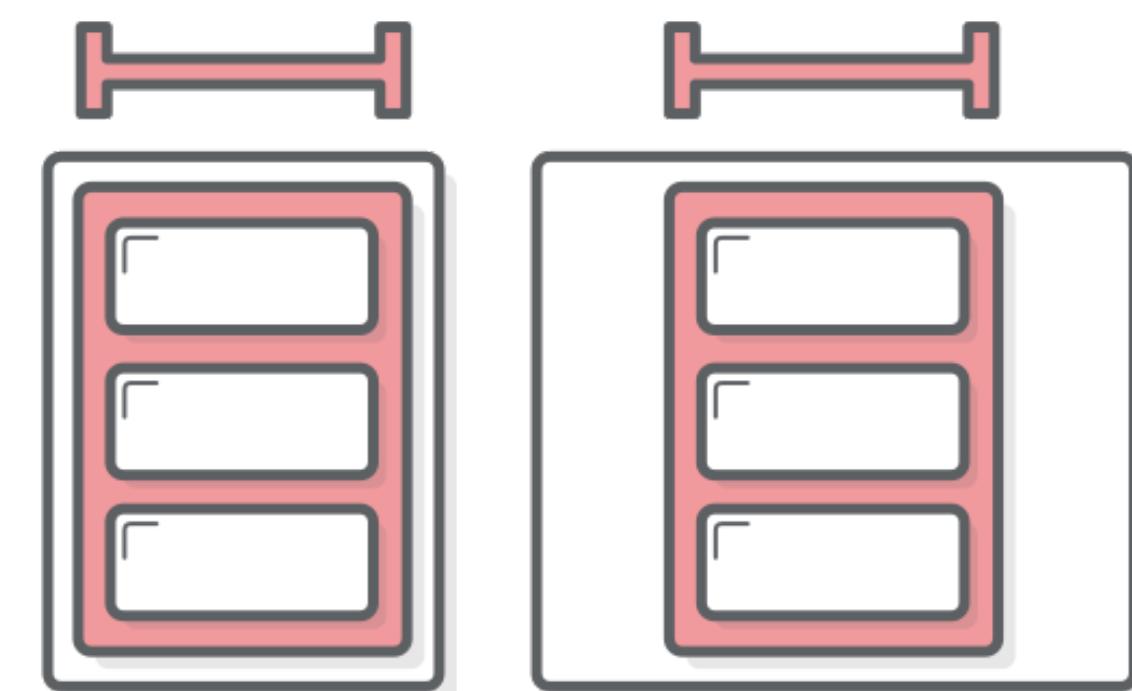
# Layouts

## Tipos de layout

- Fluído
  - Estica e encolhe de acordo com a largura da tela
- Largura-fixa
  - Possui a mesma largura independente das dimensões da tela



FLUID LAYOUT



FIXED-WIDTH LAYOUT

# Layouts

## Mobile first

- Abordagem simples e evolutiva.
- Evita desfazer trabalho já realizado.
- Começar o desenvolvimento pelo mobile e, depois, adicionar suporte a layouts desktop.
- Uso de media queries min-width ao invés de max-width, mais comum em códigos desktop-first

# Layouts

## CSS Layout models

- Flexbox ou Flexible Box
  - Layout unidimensional
  - Ajusta o layout da página de acordo com o conteúdo automaticamente
- Grid
  - Layout de duas dimensões
  - Contrói um layout mais rígido que não muda baseado no conteúdo

# Layouts

## Flexbox

- Basicamente é necessário definir duas propriedades

- **flex-direction**

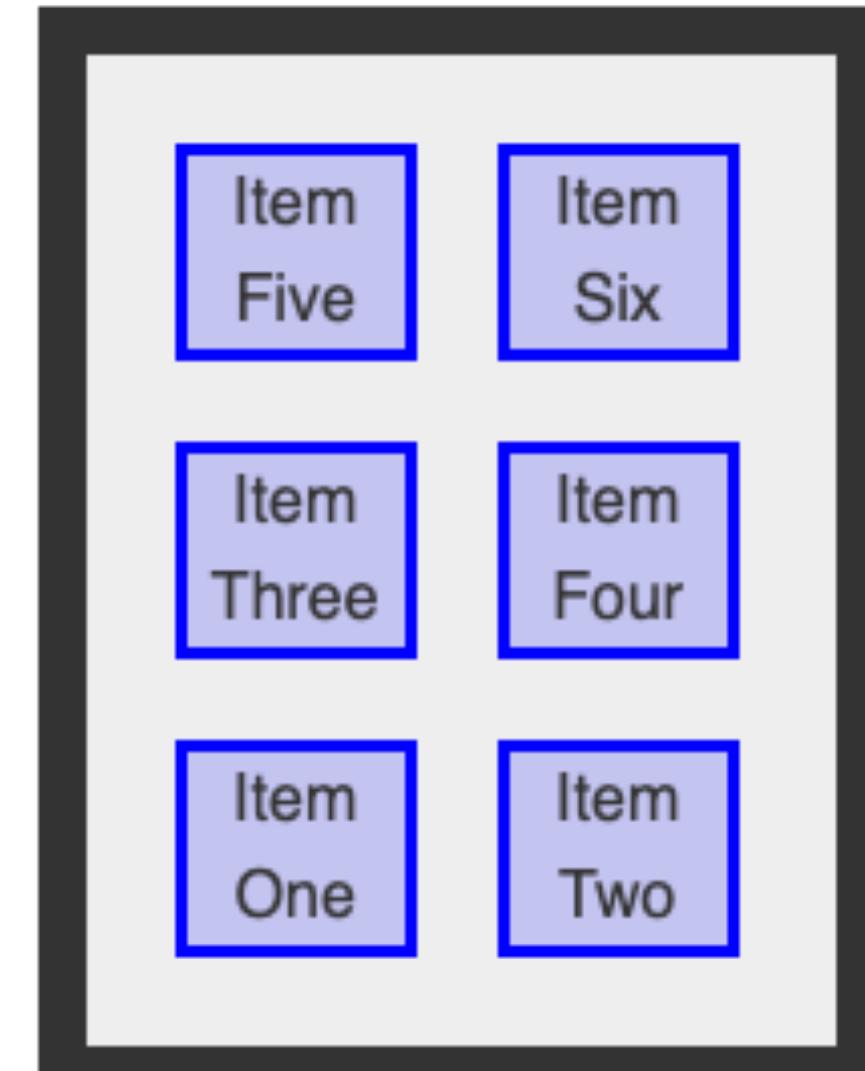
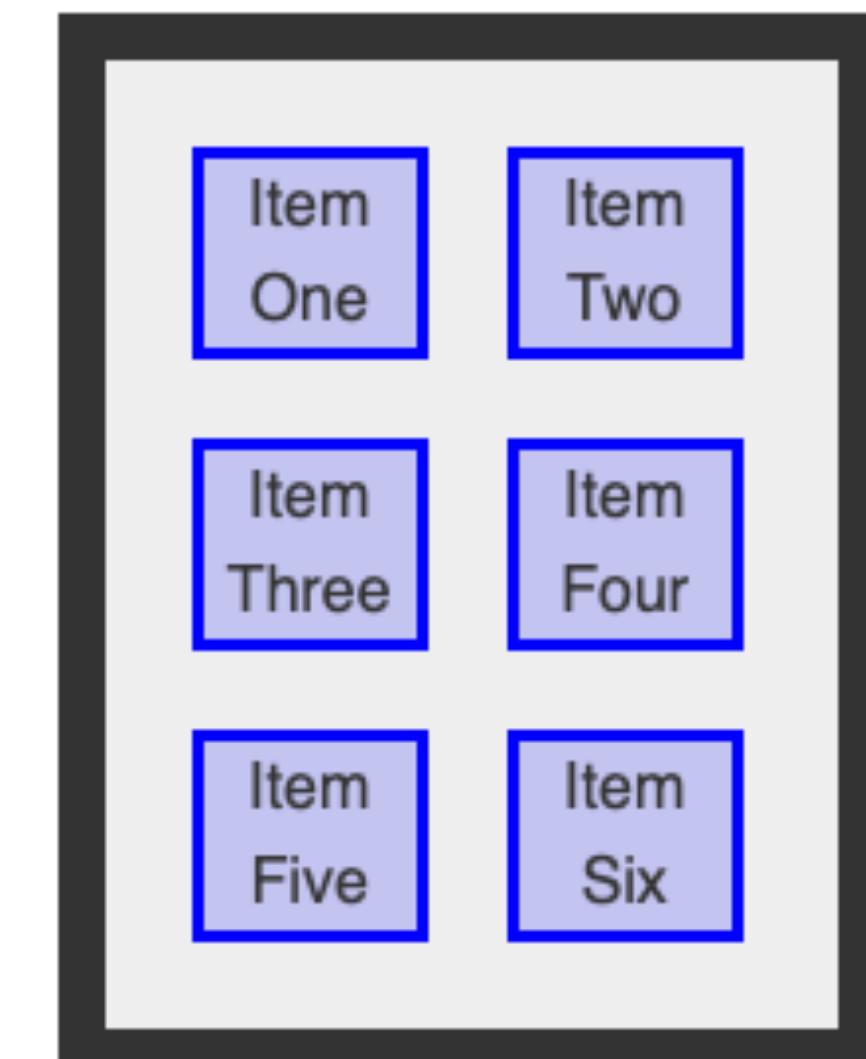
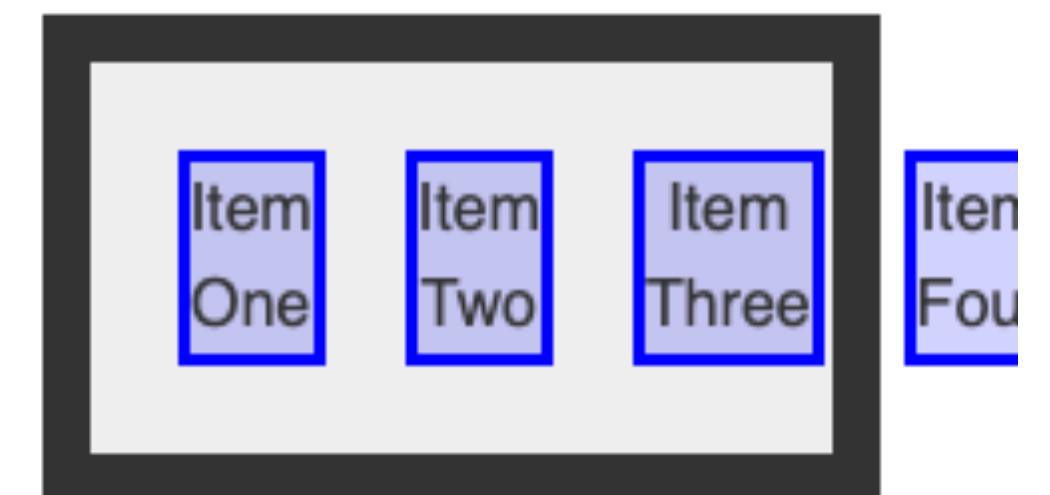
- Linha (*row*) ou coluna (*column*)

- **flex-wrap**

- nowrap

- wrap

- wrap-reverse



# Layouts

## Flexbox

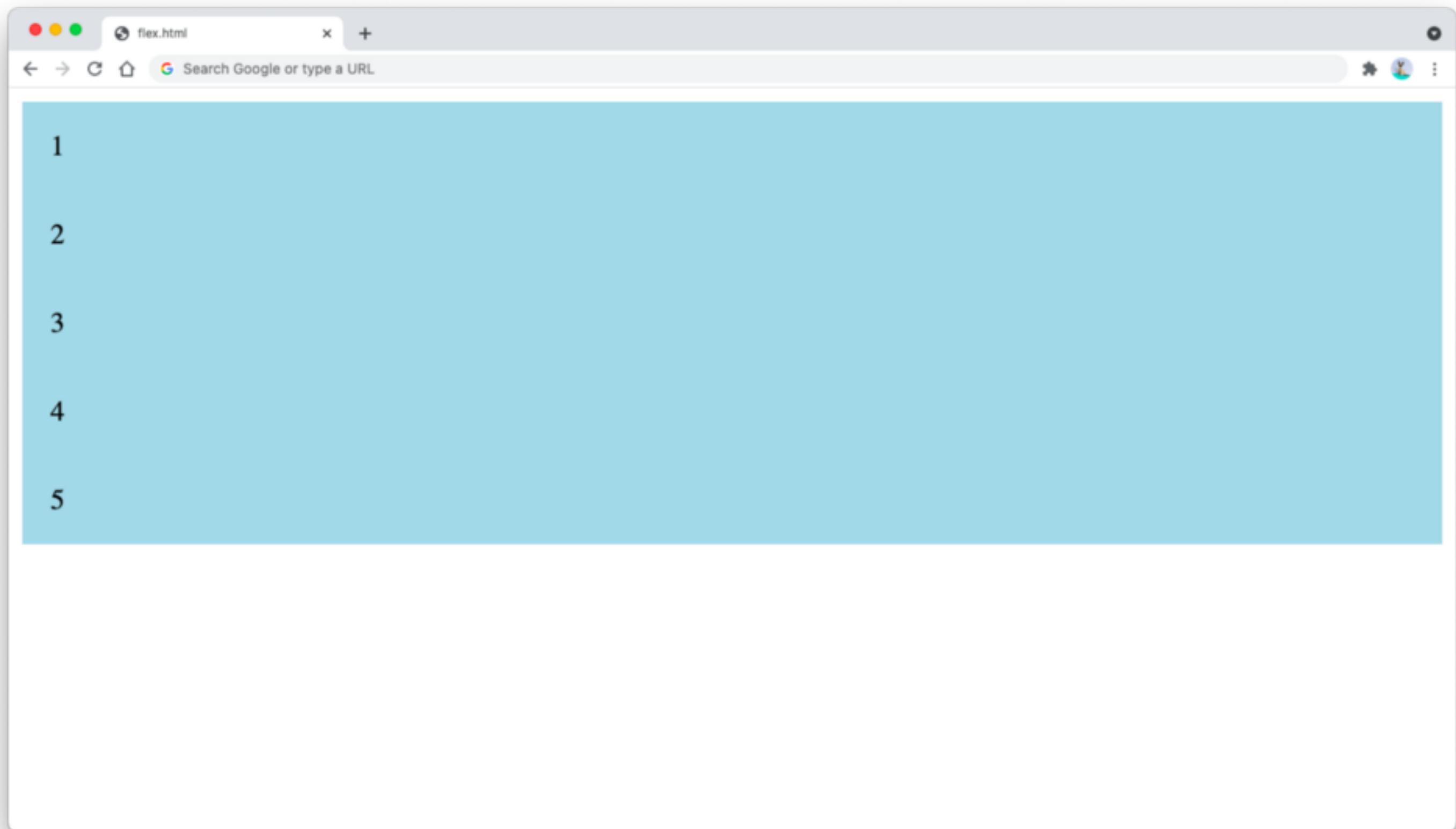
- gap
  - Espaçamento entre itens
- flex-grow
  - Determina o fator de crescimento
  - Usado para fazer com que os itens se expandam e ocupem o espaço vazio

# Layouts

## Flexbox

```
<div class="container">
  <div class="item">1</div>
  <div class="item">2</div>
  <div class="item">3</div>
  <div class="item">4</div>
  <div class="item">5</div>
</div>
```

```
.item {
  background-color: lightblue;
  padding: 1rem;
}
```



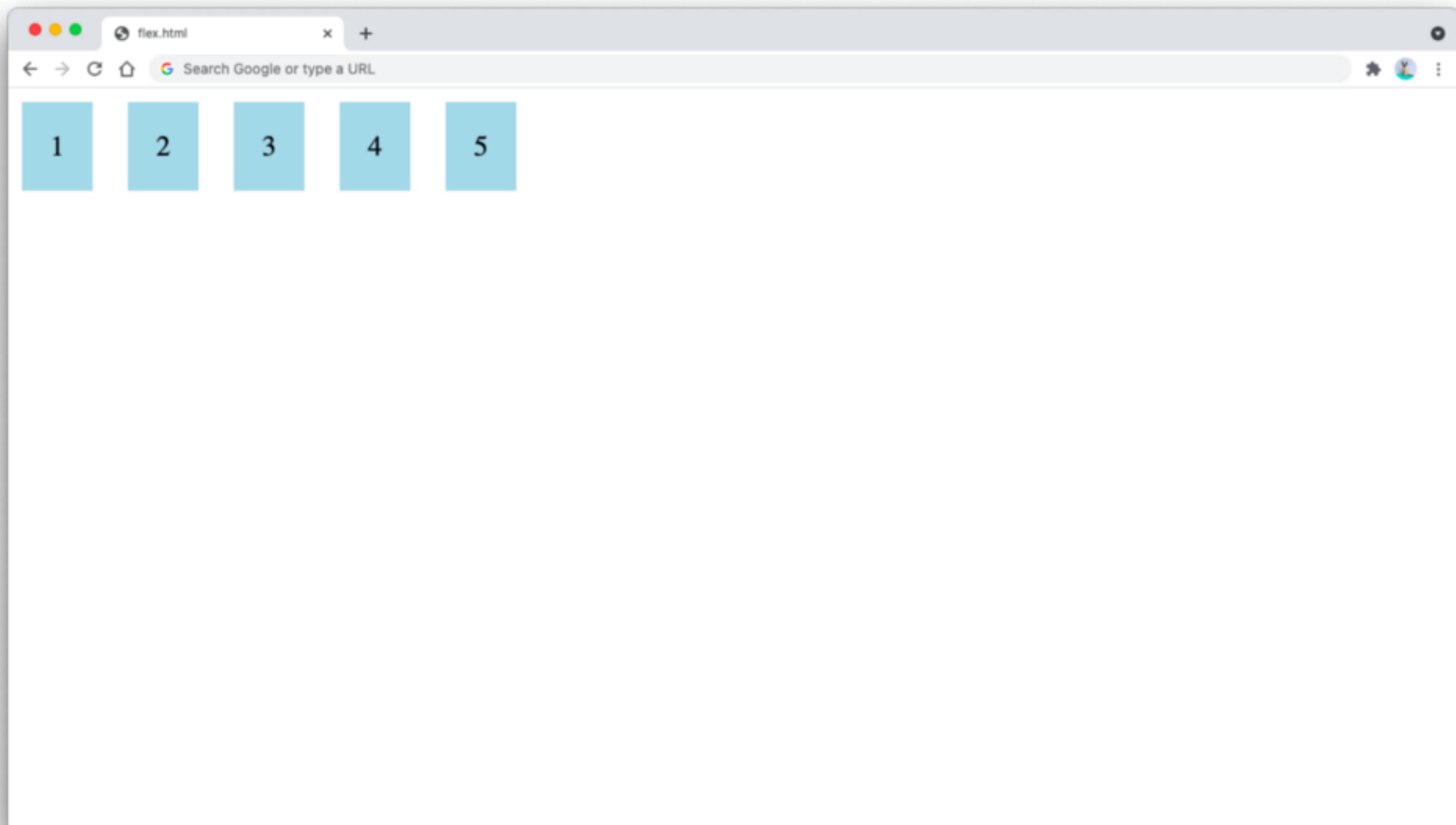
# Layouts

## Flexbox

```
<div class="container">
  <div class="item">1</div>
  <div class="item">2</div>
  <div class="item">3</div>
  <div class="item">4</div>
  <div class="item">5</div>
</div>
```

```
.container {
  display: flex;
  gap: 20px;
  flex-direction: row;
}

.item {
  background-color: lightblue;
  padding: 1rem;
}
```



# Layouts

## Flexbox

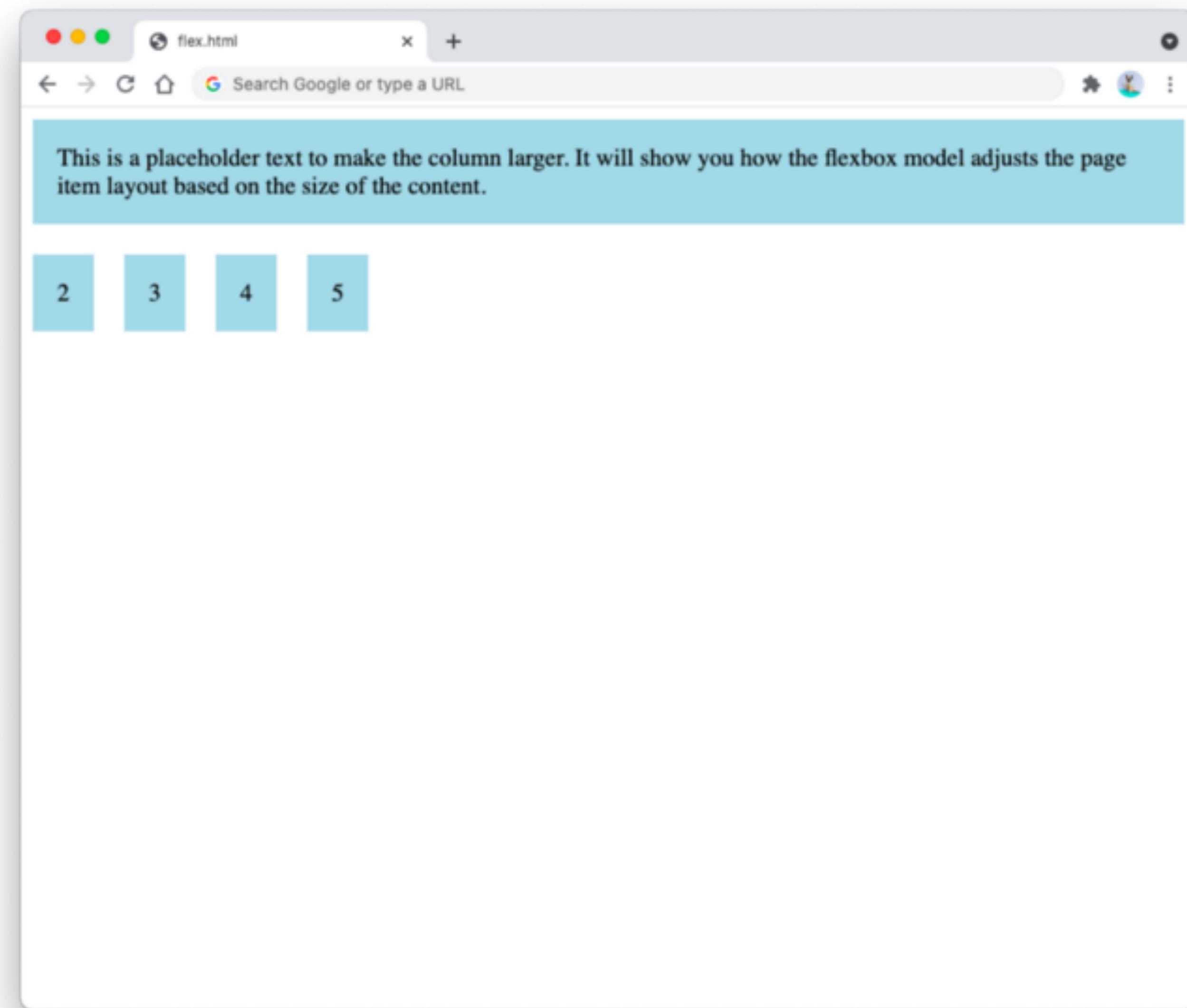
```
<div class="container">
  <div class="item">
    This is a placeholder text to make the
    column larger. It will show you how the
    flexbox model adjusts the page item layout
    based on the size of the content.
  </div>
  <div class="item">2</div>
  <div class="item">3</div>
  <div class="item">4</div>
  <div class="item">5</div>
</div>
```

```
.container {
  display: flex;
  gap: 20px;
  flex-direction: row;
  flex-wrap: wrap;
}

.item {
  background-color: lightblue;
  padding: 1rem;
}
```

# Layouts

## Flexbox



# Layouts

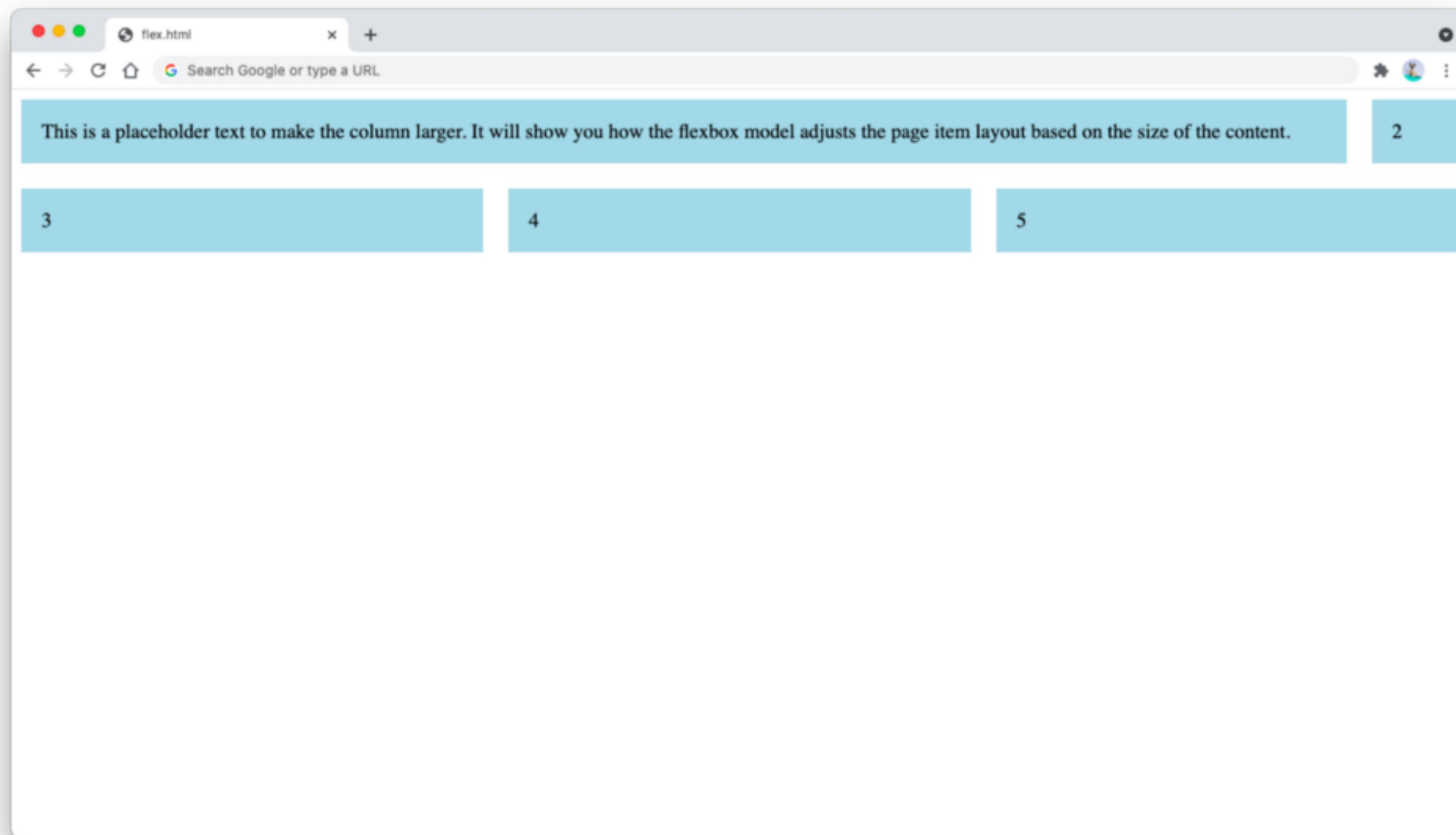
## Flexbox

```
<div class="container">
  <div class="item">
    This is a placeholder text to make the
    column larger. It will show you how the
    flexbox model adjusts the page item layout
    based on the size of the content.
  </div>
  <div class="item">2</div>
  <div class="item">3</div>
  <div class="item">4</div>
  <div class="item">5</div>
</div>
```

```
.container {
  display: flex;
  gap: 20px;
  flex-direction: row;
  flex-wrap: wrap;
}
.item {
  background-color: lightblue;
  padding: 1rem;
  flex-grow: 1;
}
```

# Layouts

## Flexbox



# Layouts

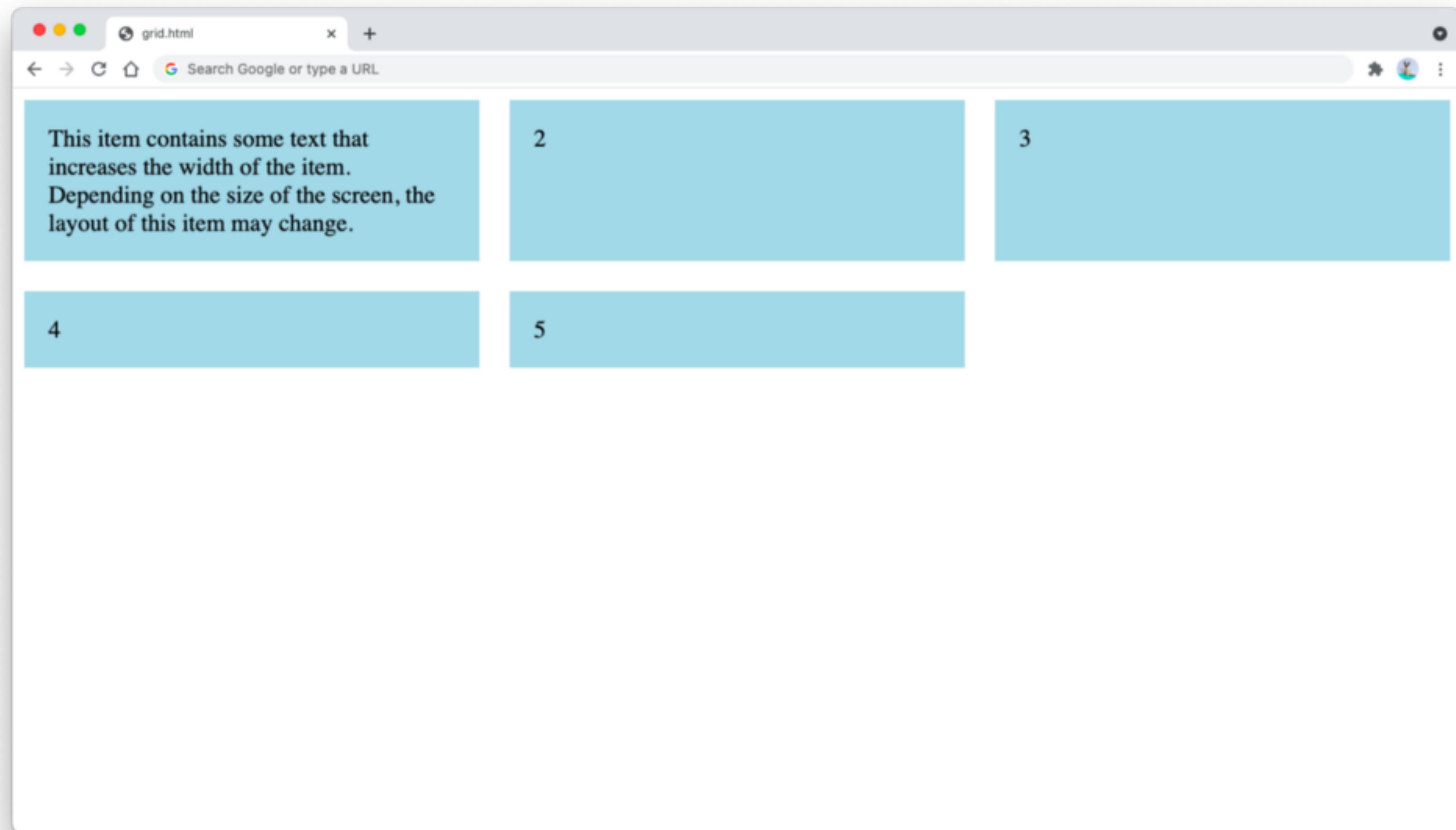
## Grid

```
<div class="container">
  <div class="item">
    This is a placeholder text to make the
    column larger. It will show you how the
    flexbox model adjusts the page item layout
    based on the size of the content.
  </div>
  <div class="item">2</div>
  <div class="item">3</div>
  <div class="item">4</div>
  <div class="item">5</div>
</div>
```

```
.grid {
  display: grid;
  grid-template-columns: 1fr
  1fr 1fr;
  gap: 20px;
}
.item {
  background-color: lightblue;
  padding: 1rem;
}
```

# Layouts

## Grid



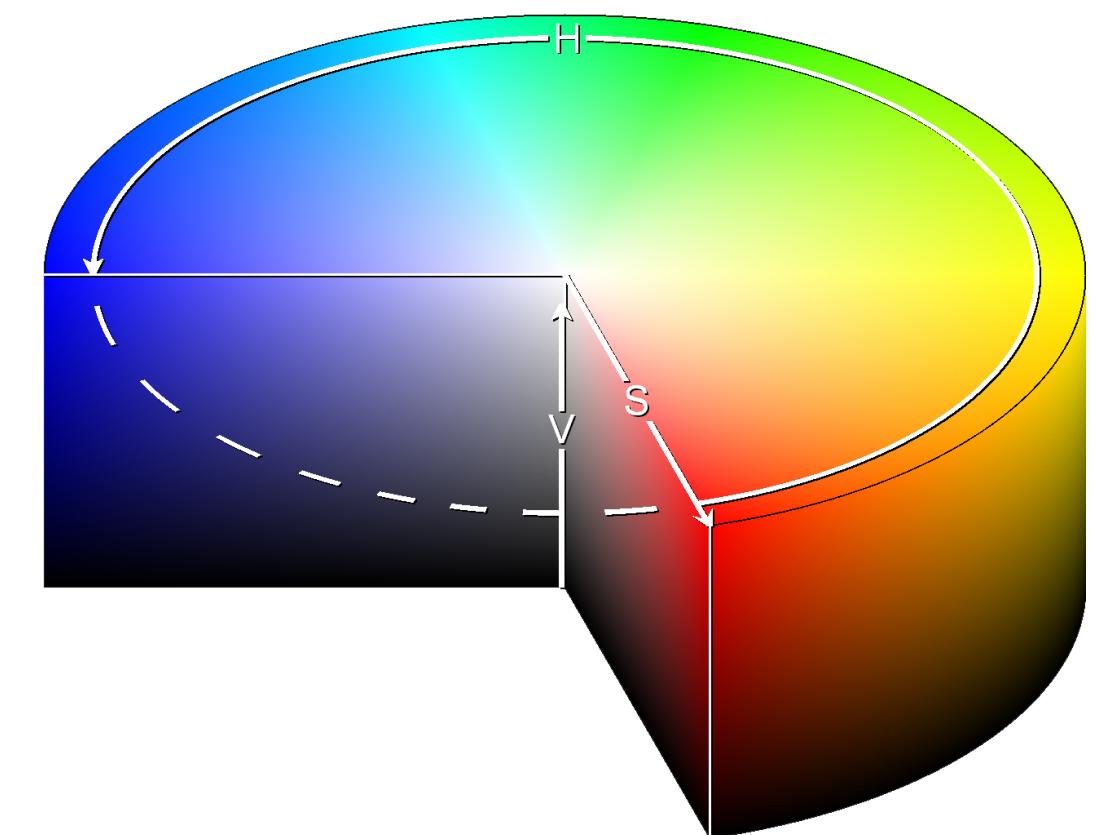
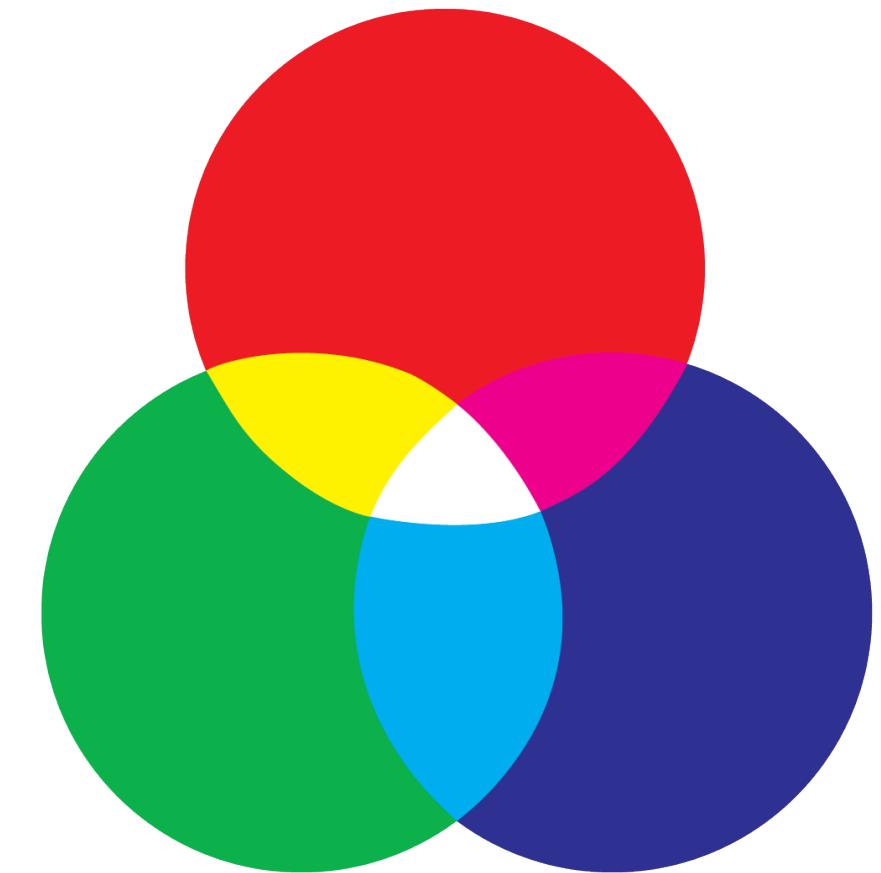
# Cores em CSS



# Cores em CSS

## Especificando cores

- O tipo cor pode ser descritos de diferentes maneiras
- Palavra-Chave (Cores pré definidas):
  - CSS 2.1: **aqua**, **black**, **blue**, **fuchsia**, **gray**, **lime**, **green**, **maroon**, **navy**, **olive**, **orange**, **purple**, **red**, **teal**, **silver**, **white\***, **yellow**
  - CSS3: novas cores foram adicionadas. **Confira todas aqui.**
- Código RGB ou RGBA: Os valores de vermelho, verde e azul de 0 até 255
- Código hexadecimal: valores RGB em base 16 00 até FF
- Coordenada cilíndrica: hsl ou hsla



# Cores em CSS

```
p {  
    color: rgba(0, 0, 255, 1.0);  
    background-color: yellow;  
    border: 10px dashed #008000;  
}
```

<p>Olhem essa combinação de cores</p>



# Cores em CSS

- Ferramentas Online
  - <https://colorschemedesigner.com/csd-3.5/>
  - <https://coolors.co>

# Fontes em CSS

AA  
—

# Fontes em CSS

Propriedade	Descrição
font-family	Que fonte será utilizada
font-size	Tamanho das letras
font-style	Usada para habilitar ou não o estilo itálico
font-weight	Usada para habilitar ou não o estilo bold

# font-family

```
p {  
    font-family: Garamond, "Times New Roman", serif;  
    font-size: 2em;  
    font-weight: bold; //normal|bold|bolder|lighter|number|initial|inherit  
    font-style: italic; //normal|italic|oblique|initial|inherit  
}
```

Maior prioridade      Mais de um palavra -> “”      Genérica

- Unidade absoluta
  - Mesmo tamanho em todos dispositivos
  - Pré definidos: xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger
  - **pt** - point: 1/72 in
  - **pc** - pica: 12 points ou 1/6 in
  - **mm** - milímetro: 1/10 cm
  - **cm** - centímetro: 1/100 m
  - **in** - polegada: 2,54 cm
- Unidade relativa
  - Adapta-se ao uso de diferentes tipos de mídia
  - **em**: ...ao tamanho da fonte ('font-size') herdada
  - **ex**: ...a altura da letra x (xis) da fonte herdada
  - **px** - pixel: ...ao dispositivo (mídia) de exibição
  - **%**: ...à percentagem de uma medida previamente definida

# Propriedades de Texto

T

# Propriedades de Texto

Propriedade	Descrição
text-align	Alinhamento do texto dentro do elemento
text-decoration	Decorações, ex: sublinhado
line-height word-spacing letter-spacing	Espaçamento entre as várias linhas, palavras e letras respectivamente
text-indent	Margem antes da primeira letra do parágrafo

# Propriedades de Texto

```
p {  
    text-align: left;  
    text-decoration: underline; //none|underline|overline|line-through|blink  
    word-spacing: -1px;  
    letter-spacing: -1px;  
    text-indent: 50px;  
}
```

# Propriedade CSS para listas



# Propriedade CSS para listas

## list-style-type

- Define o tipo de marcador da lista
- Só é aplicada no caso da propriedade `list-style-image` estiver com o valor `none` ou se a imagem estiver indisponível
- A cor do marcador é definido pela propriedade `color`

# Propriedade CSS para listas

- none: sem marcador
- disc: círculo (bolinha cheia)
- circle: circunferência (bolinha vazia)
- square: quadrado cheio
- decimal: números 1, 2, 3, 4, ...
- decimal-leading-zero
- lower-roman: romano minúsculo i, ii, iii, iv, ...
- upper-roman: romano maiúsculo I, II, III, IV, ...
- lower-alpha: letra minúscula a, b, c, d, ...
- upper-alpha: letra maiúscula A, B, C, D, ...
- lower-greek
- lower-latin
- upper-latin
- hebrew
- armenian
- georgian
- cjk-ideographic
- hiragana
- katakana
- hiragana-iroha
- katakana-iroha

# list-style-image

## list-style-image

- Especifica uma imagem para ser o marcador da lista
- **None:**
  - Nenhum marcador em forma de imagem
  - Será exibido o que for determinado na propriedade list-style-type
- **URI:**
  - Se a imagem estiver disponível ela será exibida no local reservado para o marcador

# Boas práticas

BEST  
PRACTICE



# Boas práticas

## Tipos de folha de estilo

- Usar uma ou mais folhas de arquivo externa
  - Reforça a separação entre conteúdo e apresentação
- Folhas de estilos internas
  - São justificadas se usadas em páginas que tenham uma aparência muito diferente do resto do site
- Folhas de estilos internas também pode ser usadas combinando com folhas externas
  - Para estilizar um elemento específico que ocorre em apenas uma página
- Folhas de estilo em linhas devem ser **evitadas**
  - Acabam com independência entre conteúdo e sua apresentação

# Boas práticas

## Conteúdo vs apresentação

- HTML é para **conteúdo**
  - O que está na página (posição; lista; código, etc)
- CSS é para **apresentação**
  - como exibir a página (em negrito, centralizado; 20px margem, etc)
- Manter o conteúdo separado da apresentação é um princípio muito importante
- Se o HTML não contém estilos, sua aparência inteira pode ser alterado por troca .css arquivos
  - Exemplo: [CSS Zen Garden](#)

# Boas práticas

## CSS Reset

- Cada navegador tem estilos definidos por padrão, diferentes entre os navegadores.
- CSS Reset seta um valor básico para todas as características do CSS, sobrescrevendo totalmente os estilos padrão do navegador.
- Exemplos:
  - [HTML5 Boilerplate](#)
  - [YUI3 CSS Reset](#)
  - [Eric Meyer CSS Reset](#)

# Referências

- <https://developer.mozilla.org/pt-BR/docs/Web>
- <https://www.internetingishard.com/html-and-css/>
- <https://blog.bitsrc.io/css-flexbox-vs-grid-layout-5ae41e28d1d2>

# Referências

## Links úteis

- Cores e esquema de cores
  - <https://colorschemedesigner.com/csd-3.5/>
  - <https://coolors.co>
- Fontes
  - <https://fonts.google.com>
- CheatSheet
  - <https://htmlcheatsheet.com>
  - <https://htmlcheatsheet.com/css/>

# Referências

## Links úteis

- Efeito em cascata
  - <https://wattenberger.com/blog/css-cascade>
- Flex e Grid
  - <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
  - <https://css-tricks.com/snippets/css/complete-guide-grid/>
  - <https://www.origamid.com/projetos/css-grid-layout-guia-completo/>

Por hoje é só