



UNIVERSIDADE
FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

Fundamentos de CSS

QXD0020 - Desenvolvimento de Software para Web

Prof. Bruno Góis Mateus (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**

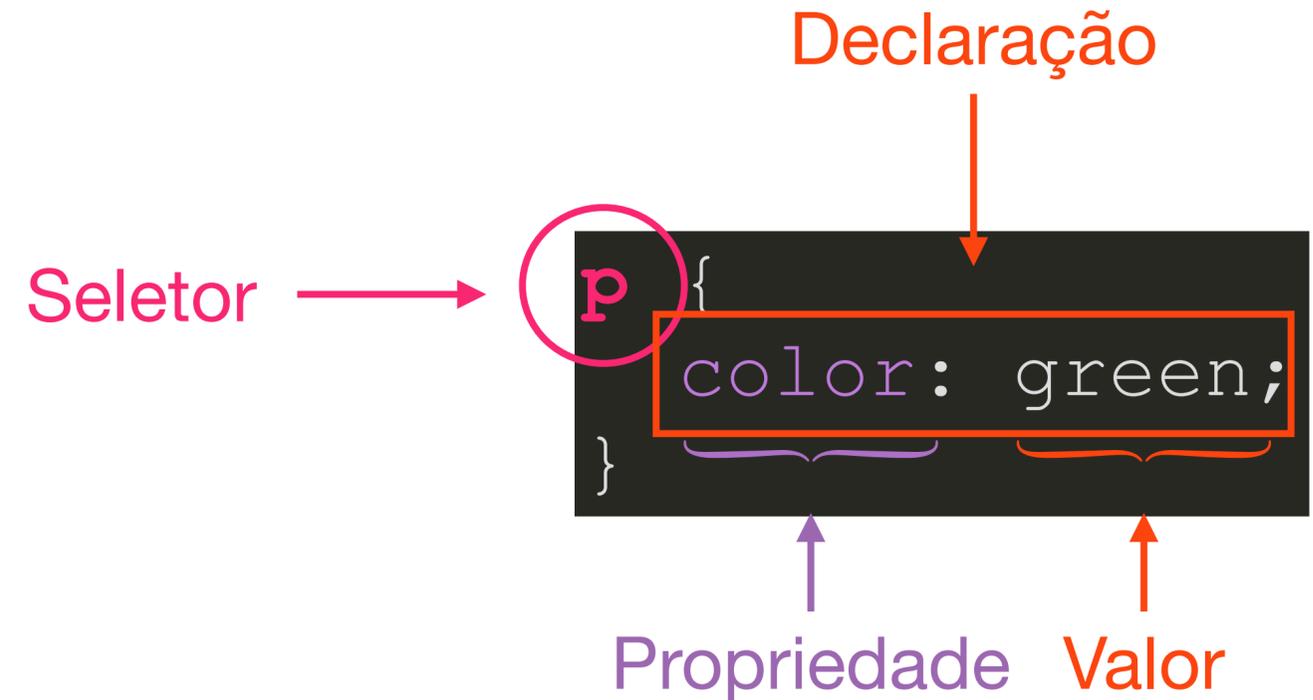
- Elemento HTML identificado por sua tag, classe ou ID, e para o qual a regra será válida

- **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" href="/style.css" media="screen" >
  </head>
</html>
```

- O atributo **rel** é obrigatório
 - Adiciona mais valor semântico por meio de metadados
 - Usando com css sempre terá o valor “**stylesheet**”, conteúdo diversos valores são aceitos como pode ser visto em [The Road to HTML 5: Link Relations](#)

Introdução

Folha de estilo externa

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

Introdução

Atributo media

Nome	Valor
<code>all</code>	Valor padrão. Usado em qualquer tipo de dispositivo.
<code>print</code>	Usado no modo de pré-visualização de impressão e em páginas impressas
<code>screen</code>	Usado em dispositivos com tela, computadores, tablets, smart-phones
<code>speech</code>	Usado para leitores de tela

Para mais opções envolvendo o atributo media confira a página: [HTML <link> media attribute](#)

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, podemos definir o atributo **media**

Introdução

Folha de estilo interna

```
<!DOCTYPE html>
<html>
  <head>
    <style 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>
```

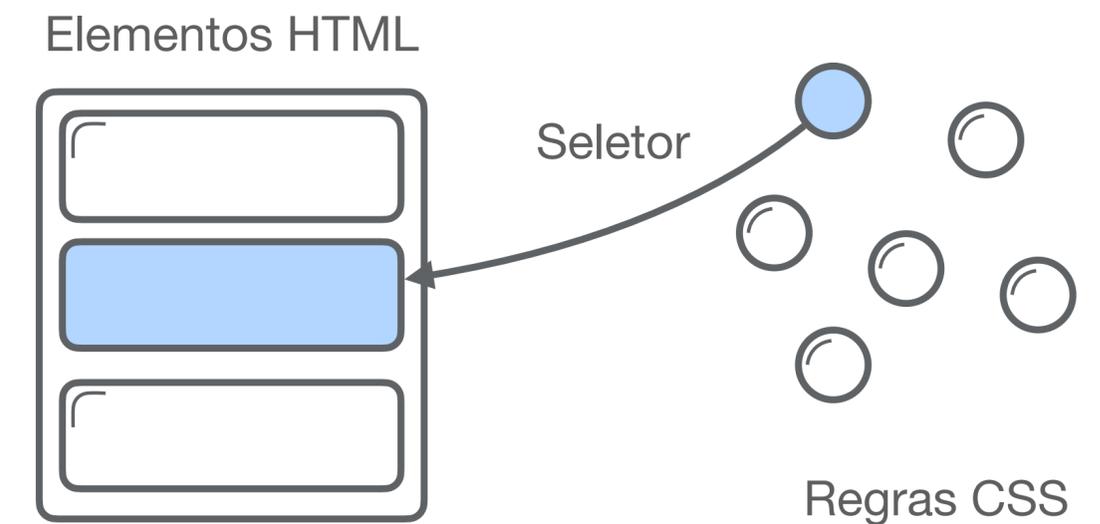
Seletores

```
p {  
  color: green;  
}
```

Seletores

Seletor de elemento

- 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



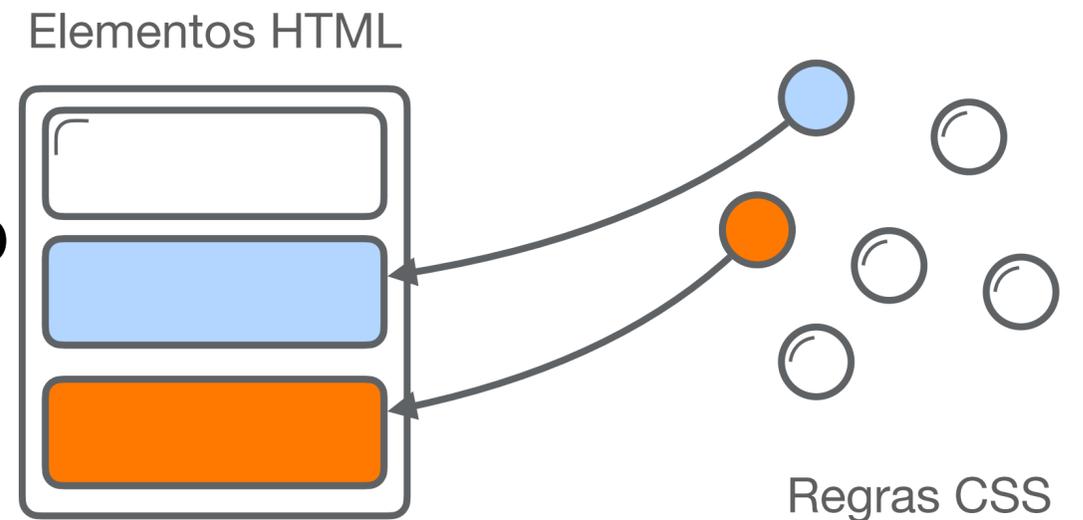
```
p {  
  color: #5EC85E;  
}
```

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 é 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

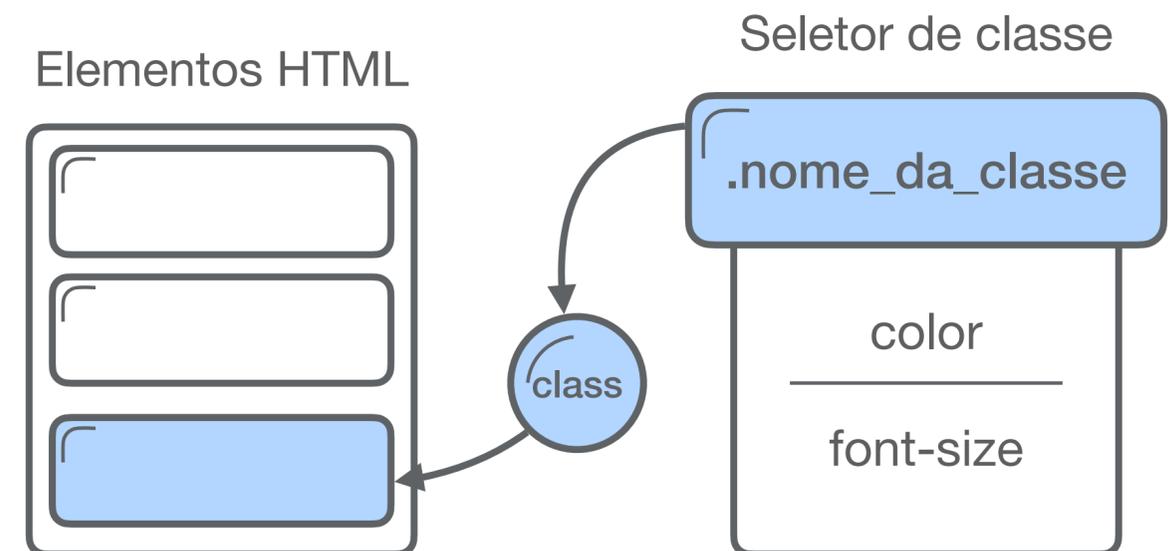
A nossa missão é fornecer mais espartanos

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

- Aplica a regra de acordo com cada seletor especificado 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

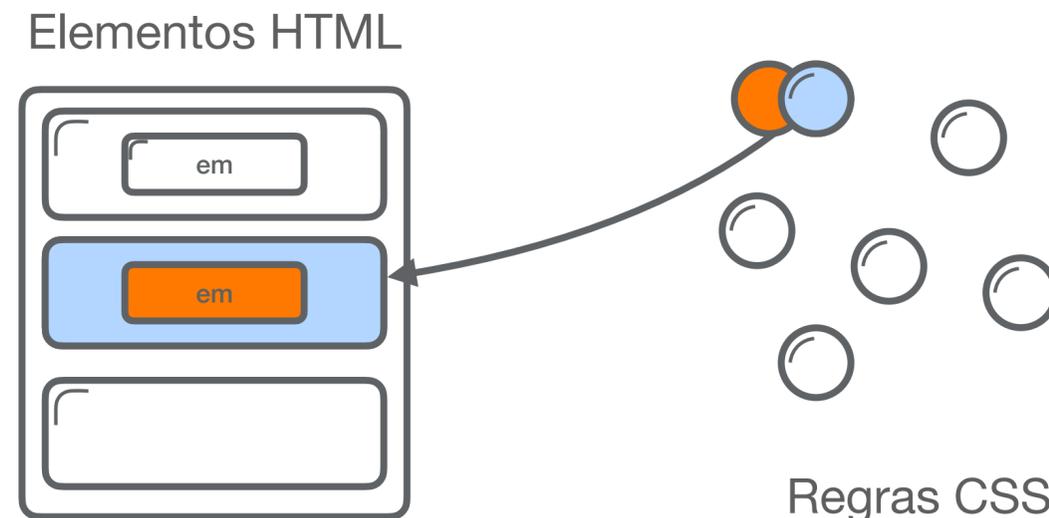
Meu **subtitulo**

Seletores

Seletor de descendência

- Permite utilizar como alvo elementos contidos em outro elemento
- 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: orange;  
}
```

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

Mercadinho Compre aqui
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;  
}
```

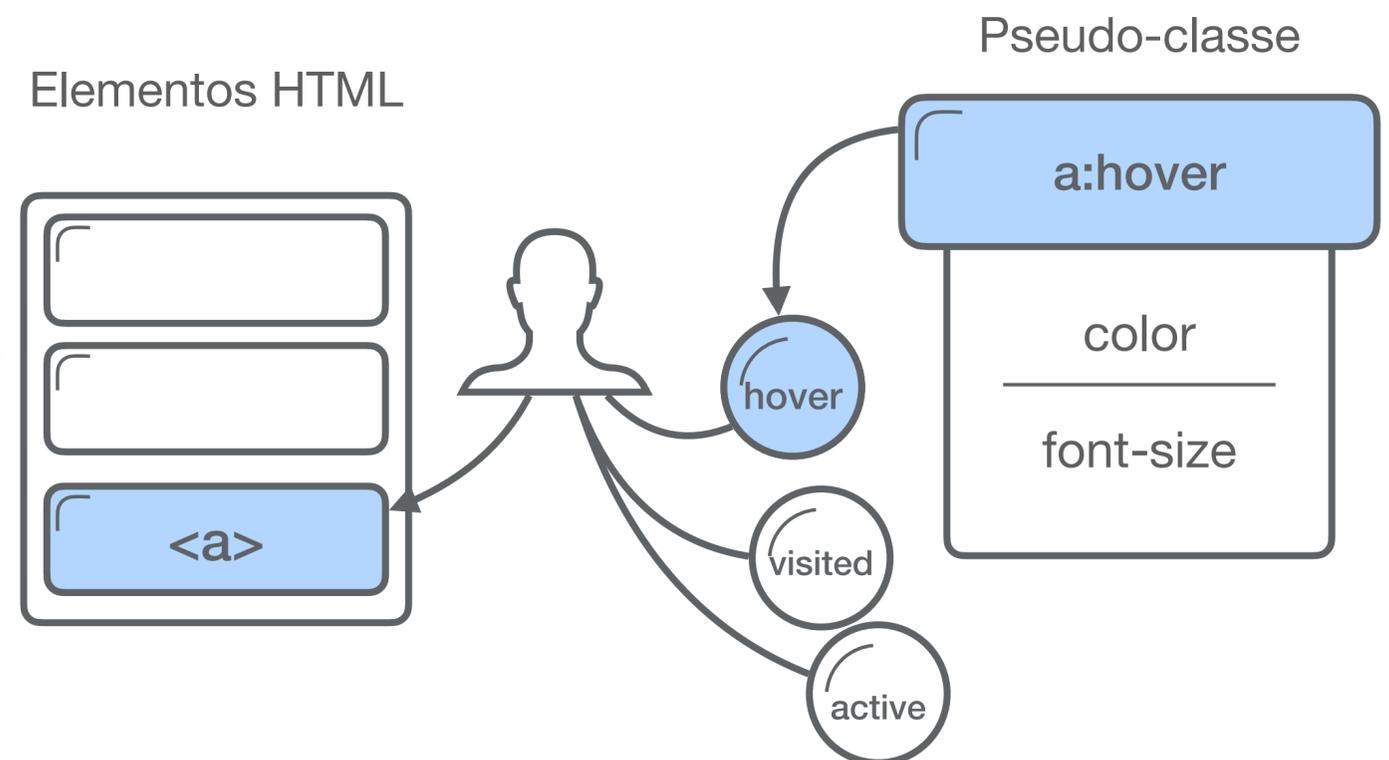
Pseudo-classes

- São similares a uma classe CSS, porém não são especificadas na marcação
- Seleciona elementos HTML baseados em seu estado
 - Ex: se um link foi visitado
- Sintaxe:
 - Um seletor com **:** seguido do nome da pseudo-classe

```
seletor:pseudo-class {  
    ...  
}
```

Pseudo-classes

- Como são associadas aos estados dos elemento, elas podem ser dinâmicas
 - São aplicadas de acordo com a interação do usuário com o documento
- Podem classificá-las em 9 categorias:
 - Estados de visualização (*Element display state*),
Entrada de dados (*input*), Linguística,
Localização, Estado de Recursos (*resource state*),
Dimensão temporal (*time-dimensional*),
Estruturais (*tree-structural*),
Ações do usuário e Funcionais.



Pseudo-classes

Pseudo-classe	Categoria	Descrição
<code>:disabled</code>	Entrada de dados	É aplicada em elementos de entrada desabilitados
<code>:invalid</code>	Entrada de dados	É aplicada em elementos de entrada que inválidos de acordo com validação
<code>:lang()</code>	Linguística	Seleciona um elemento de acordo com o seu idioma
<code>:link</code>	Localização	É aplicada em links não visitados
<code>:visited</code>	Localização	É aplicada em links já visitados
<code>:playing</code>	Estado de recursos	É aplicado em um elemento de media quando seu conteúdo está em reprodução
<code>:empty</code>	Estrutural	É aplicado em elementos que não possuem filhos
<code>:active</code>	Ações de usuário	Link no momento em que ele é clicado
<code>:hover</code>	Ações de usuário	Quando usuário põe o cursor em cima de um elemento
<code>:focus</code>	Ações de usuário	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

```
.button {  
  border-radius: 5px;  
  border: 1px solid gray;  
  padding: 5px;  
}  
  
a.button:link {  
  color: gray;  
  text-decoration:none;  
}  
  
a.button:visited {  
  color: black;  
}  
  
a:hover {  
  background-color: blue;  
}
```

```
<a href="#1" class="button">Ação 1</a>  
<a href="#2" class="button">Ação 2</a>  
<a href="#3">Ação 3</a>
```



Pseudo- elementos

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

Pseudo-elementos

- Assim como as pseudo-classes, combinam com elementos virtuais não especificados na marcação
- Um pseudo-elemento permite a estilização de uma parte específica do elemento selecionado pelo seletor
 - Permitem selecionar um conteúdo que não possui um elemento específico para ele

```
seletor::pseudo-element {  
  ...  
}
```

Pseudo-elementos

Pseudo-classe	Descrição
<code>::after</code>	Insere alguma coisa depois do conteúdo do elemento selecionado
<code>::before</code>	Insere alguma coisa antes do conteúdo do elemento selecionado
<code>::first-letter</code>	Aplica a regra na primeira letra do conteúdo de texto de elemento selecionado
<code>::first-line</code>	Aplica a regra na primeira linha do conteúdo de texto de elemento selecionado
<code>::marker</code>	Aplica a regra nos marcadores dos itens de listas
<code>::selection</code>	Aplica a regra na porção do conteúdo selecionado de um elemento

Pseudo-elementos

```
p::first-letter {  
  font-size: 200%;  
}  
  
p::first-line {  
  color: blue;  
}
```

<p>Você
 pode utilizar apenas um pseudo-elemento em um seletor.</p>
<p> Ele
 deve aparecer depois da declaração de um elemento simples</p>.

Você

pode utilizar apenas um pseudo-elemento em um seletor

Ele

deve aparecer depois da declaração de um elemento simples

Pseudo-elementos

```
a::before {  
  content: '[';  
}
```

```
a::after {  
  content: ']';  
}
```

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

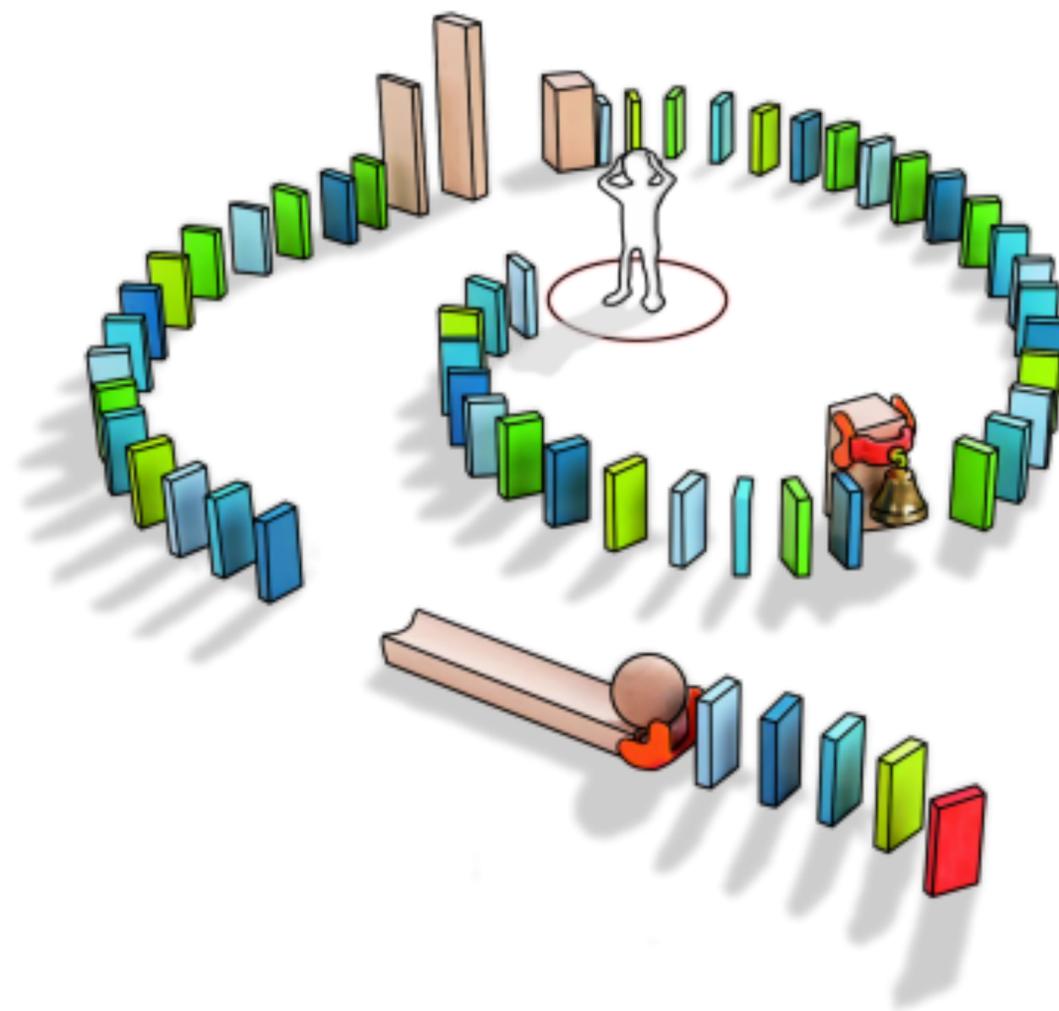
```
<a href="#">Link 1</a><br>  
<a href="#">Link 2</a><br>  
<a href="http://">Link 3</a><br>
```

[Link 1]

[Link 2]

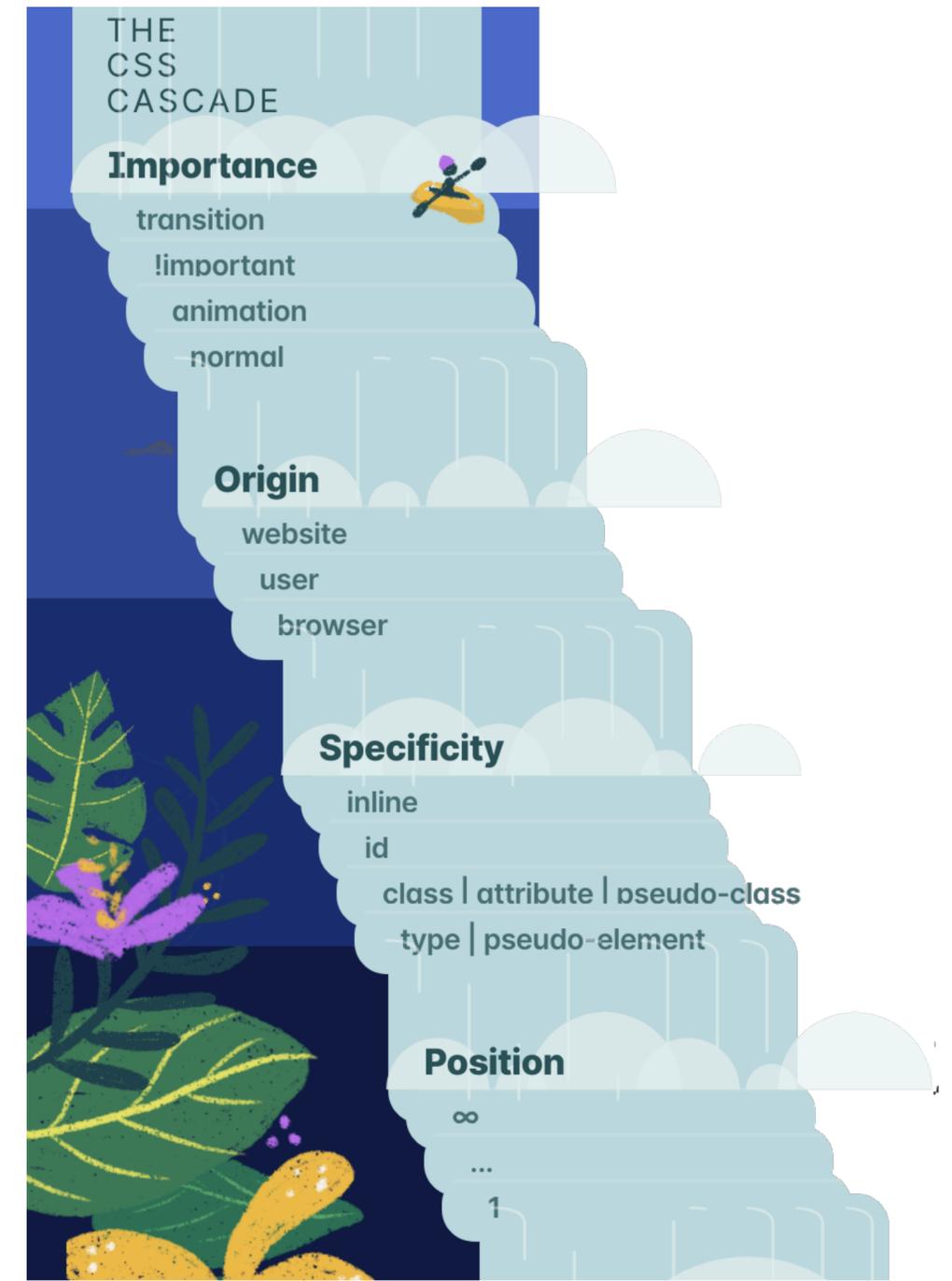
[Link 3](externo)

Efeito em cascata

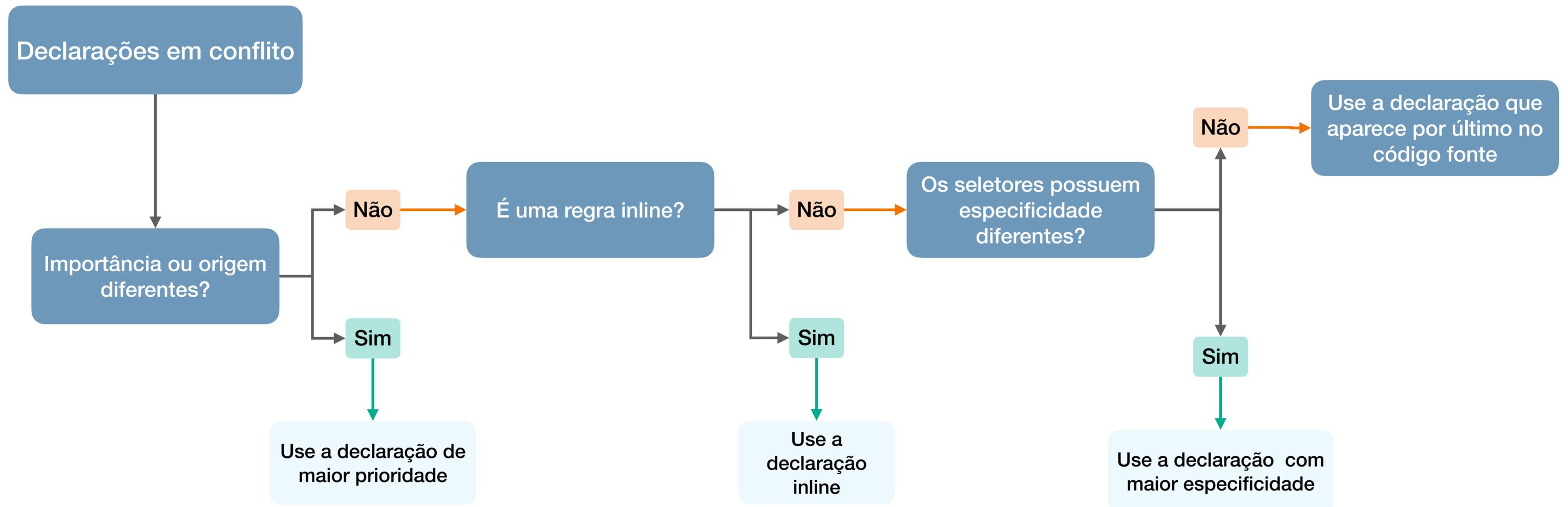


Efeito em cascata

- 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

Especificidade

- O cálculo da especificidade considera três colunas
 - Coluna de ID
 - Considera apenas seletores de ID
 - Coluna de classe
 - Considera seletores de classe, atributos, e pseudo-classes, como `:hover` and `:first-of-type`
 - Coluna de tipo
 - Considera o seletor de elemento (*tag*) e pseudo-elementos



Efeito em cascata

```
.parent {  
  color: blue;  
  background-color: gray;  
  width: 500px;  
  height: 500px;  
}
```

```
.special.box {  
  color: red;  
}
```

```
#b {  
  color: orange;  
}
```

```
div > .box {  
  font-size: 2em;  
  background-color: white;  
  border: 1px solid  
  margin: 10px;  
}
```

```
<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>
```

0 - 2 - 0

Caixa 1

1 - 0 - 0

Caixa 2

Caixa 3

Caixa 4

Efeito em cascata

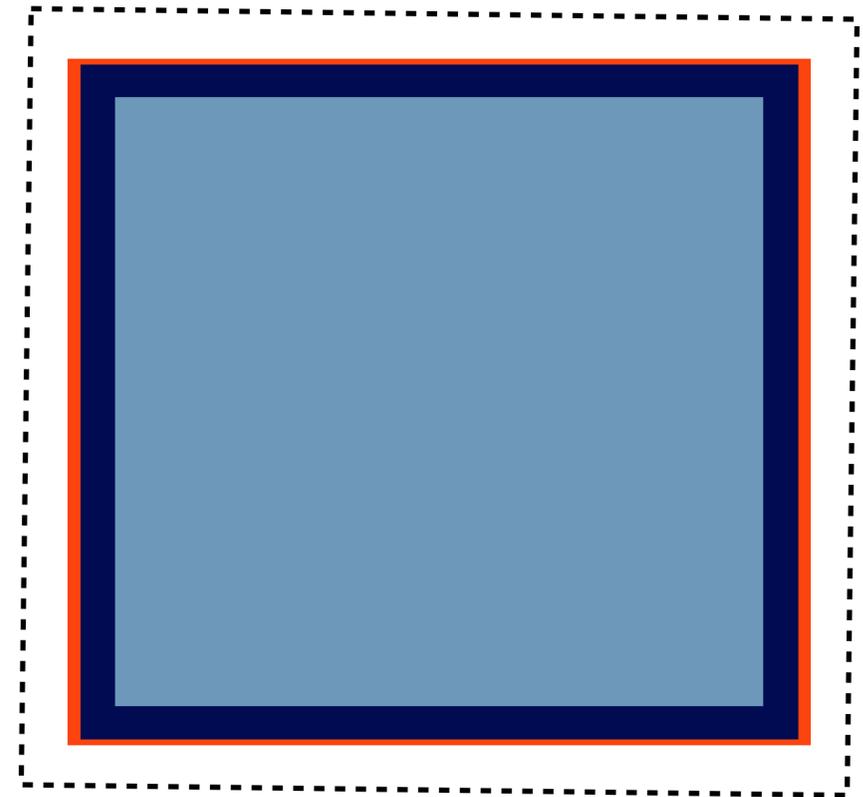
Herança em CSS

- 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

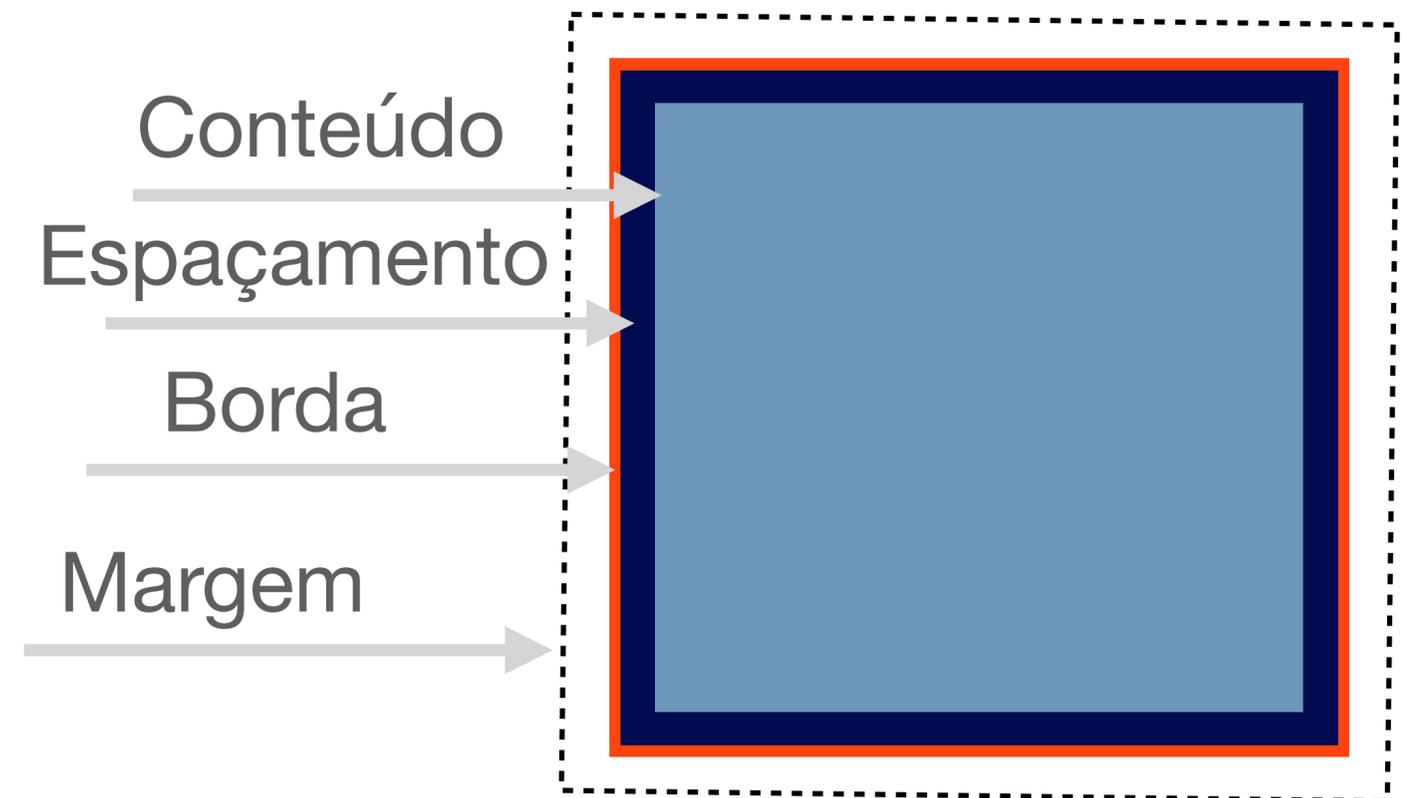
```
background, border (exceto: border-collapse e border-spacing),  
clip, content, counter, cue, display, float, height, left, margin,  
outline, overflow, padding, page-break, pause, play-during, position, right, table-  
layout, text-decoration, top, unicode-bidi, vertical-align, width, z-index.
```

CSS Box Model



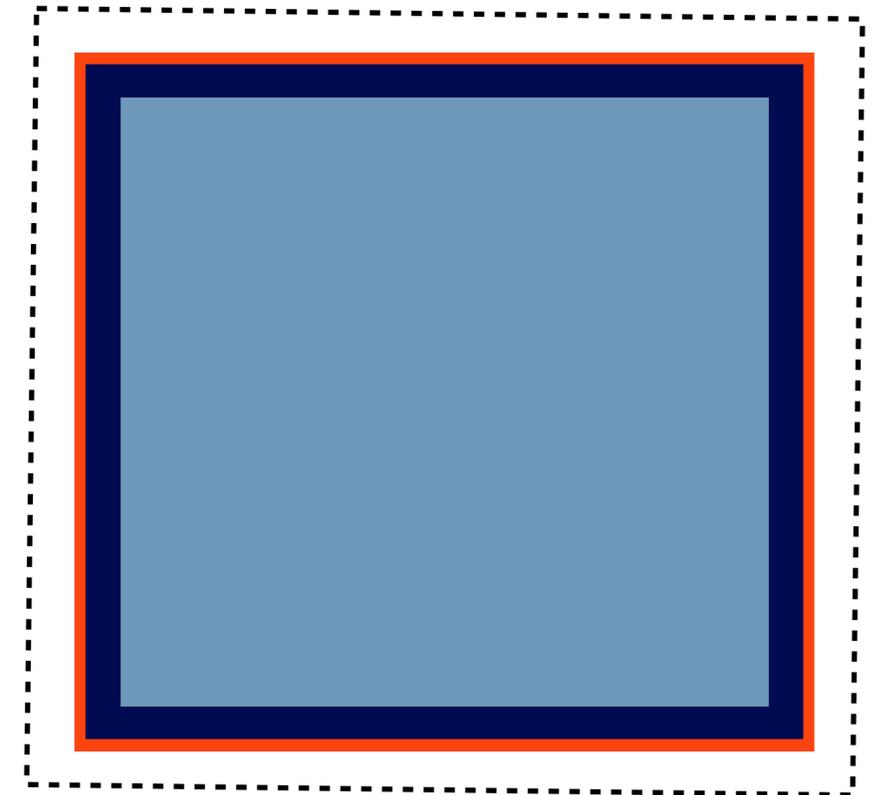
CSS Box Model

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

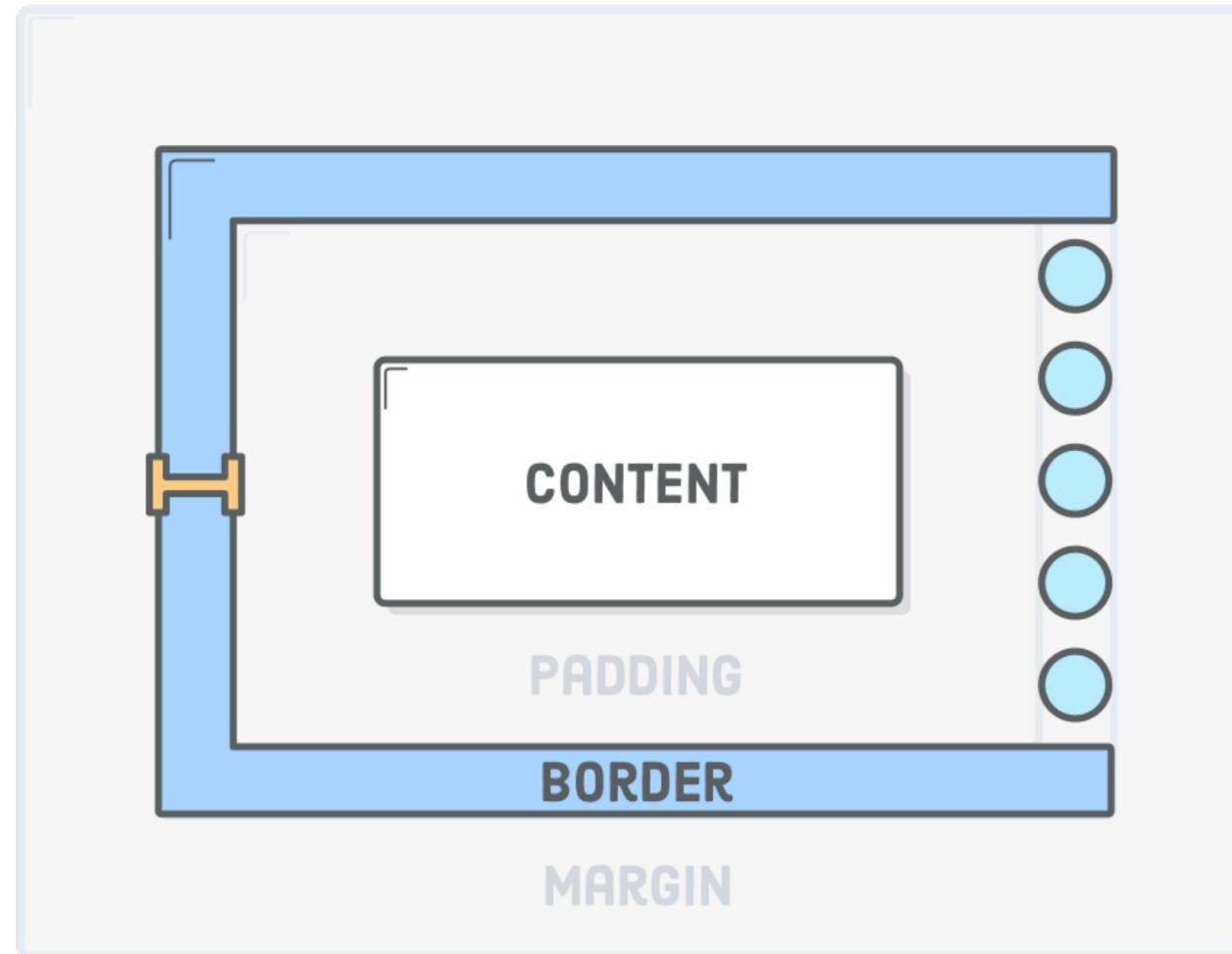
CSS Box Model

Modificando as bordas

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

CSS Box Model

Modificando as bordas



Fonte: <https://www.internetingishard.com/html-and-css/css-box-model/>

CSS Box Model

Modificando as bordas

Gerador de Bordas em CSS

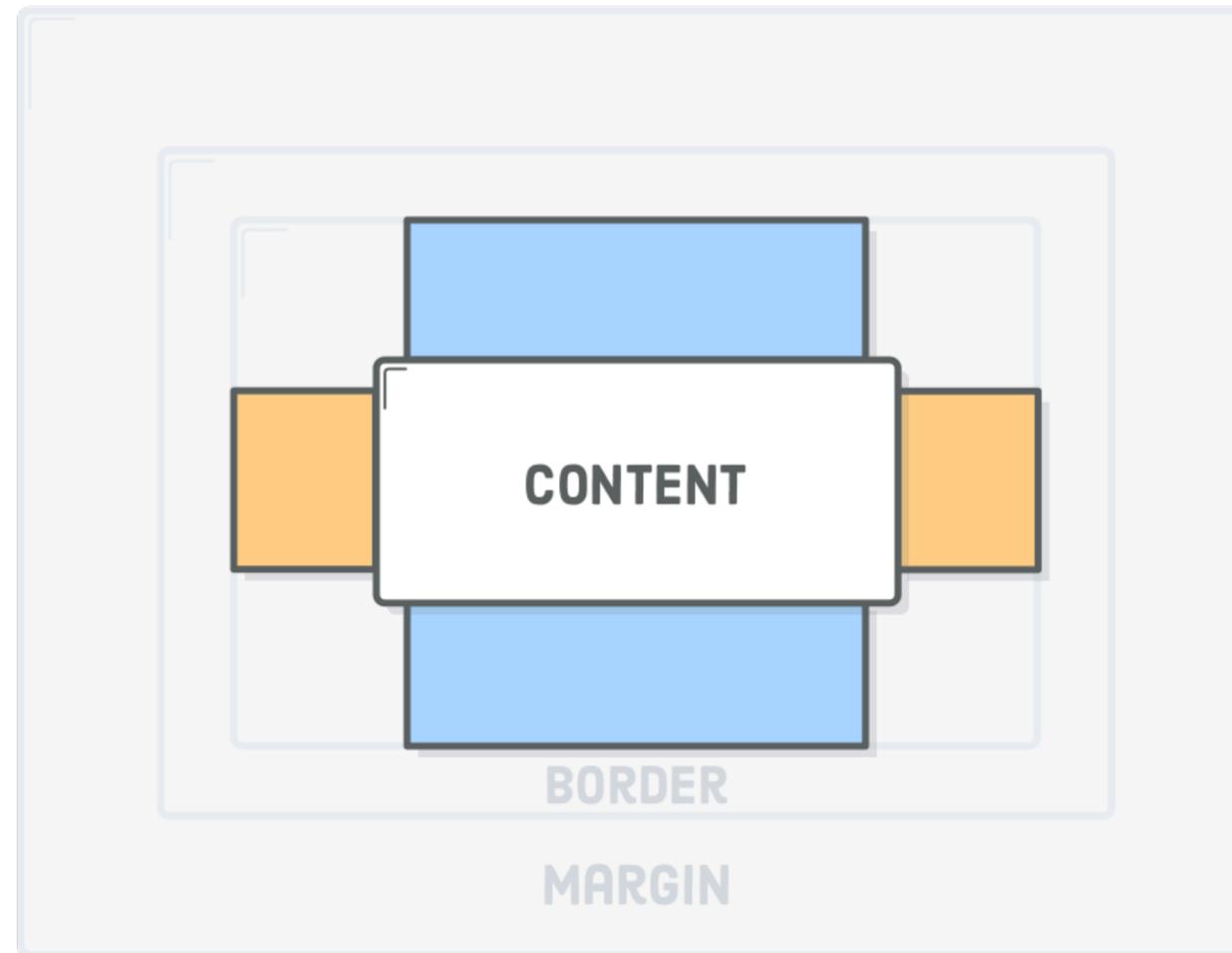
CSS Box Model

Modificando o espaçamento

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

CSS Box Model

Modificando o espaçamento



PADDING **:** **VERTICAL** **HORIZONTAL**

Fonte: <https://www.internetingishard.com/html-and-css/css-box-model/>

CSS Box Model

Modificando as margens

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

CSS Box Model

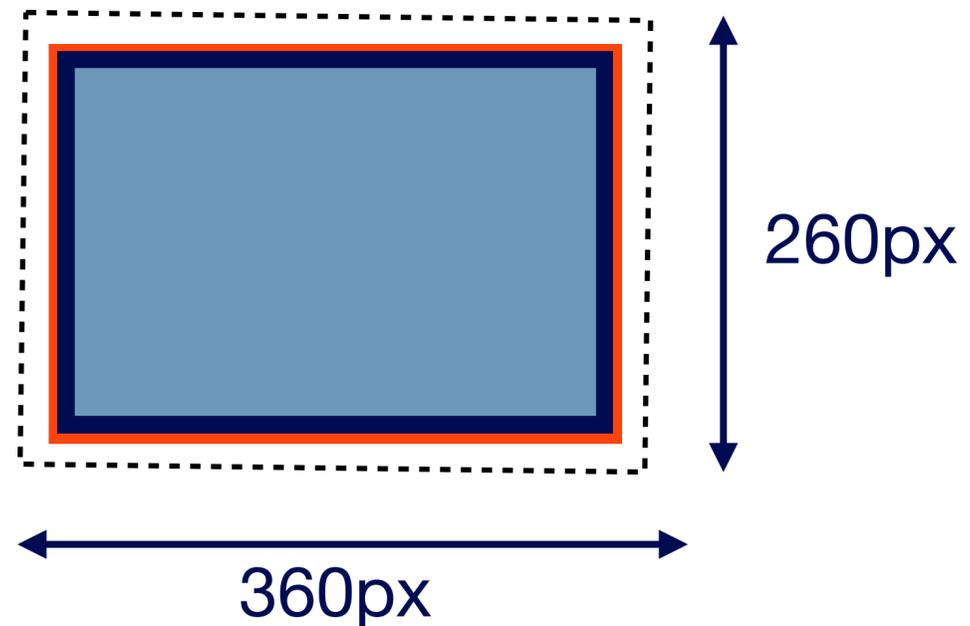
Quanto espaço ocupa um elemento?

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

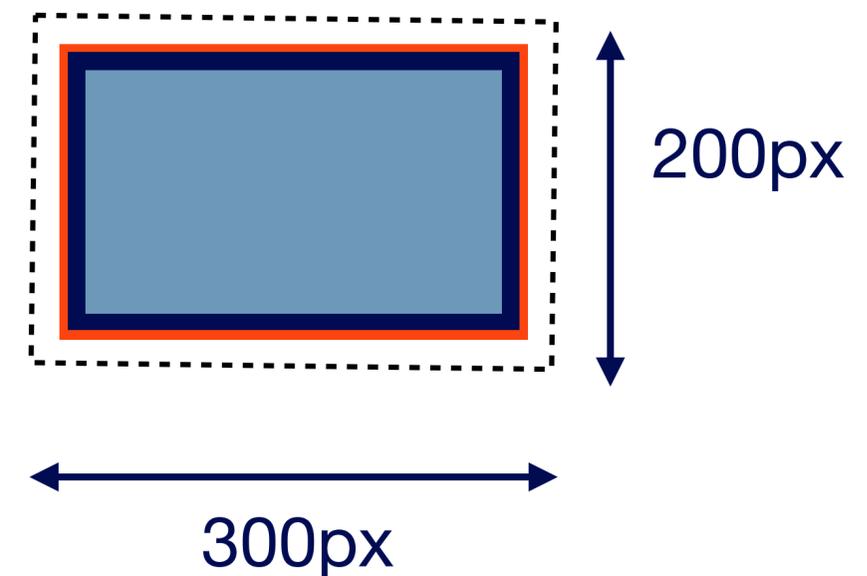
CSS Box Model

content-box vs border-box

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

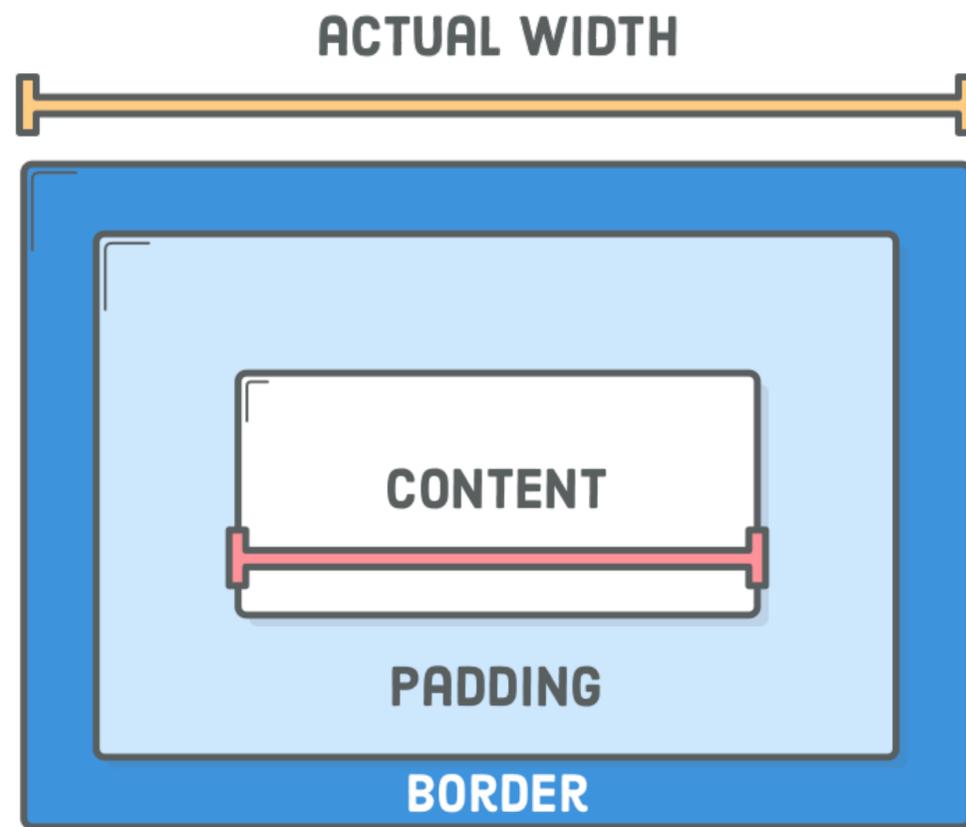


```
.box {  
  box-sizing: border-box;  
  width: 300px;  
  height: 200px;  
  padding: 10px;  
  border: 5px solid #FC440F;  
  margin: 15px;  
}
```



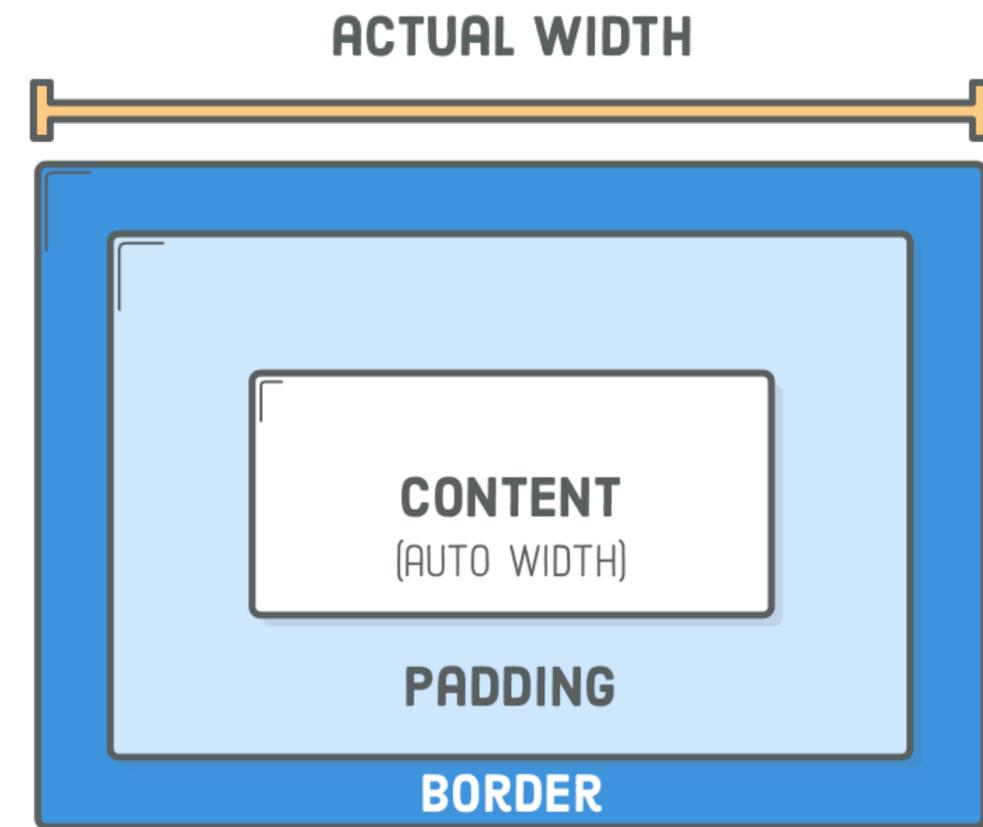
CSS Box Model

content-box vs border-box



WIDTH : 200PX

BOX-SIZING : CONTENT-BOX



WIDTH : 200PX

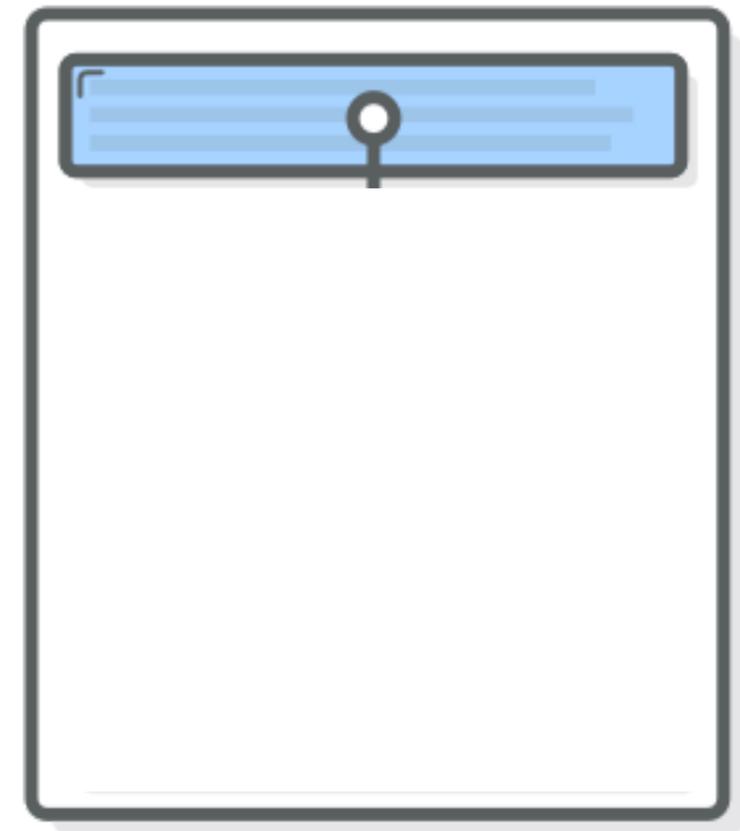
BOX-SIZING : BORDER-BOX

Elementos flutuantes



Elementos flutuantes

- Até o momento aprendemos como manipular o tamanho dos elementos
- No entanto, ainda estamos limitados ao fluxo vertical das páginas
 - Elementos em bloco sempre aparecem verticalmente um após o outro



VERTICAL FLOW

Elementos flutuantes

- Elementos flutuantes nos permitem contornar essa limitação
 - Permitindo colocar elementos lado a lado
 - São configurados pela propriedade **float**
- Essa técnica durante muito tempo foi a base da construção de layouts
 - **Flexbox** tomou esse lugar
- Ainda é uma técnica útil em outros contextos



HORIZONTAL FLOW

Elementos flutuantes

- Determinado a partir da propriedade float
 - Não segue o fluxo normal de posicionamento
 - **left** ou **right**



LEFT ALIGN

FLOAT: LEFT;



CENTER ALIGN

MARGIN: 0 AUTO;



RIGHT ALIGN

FLOAT: RIGHT;

Elementos flutuantes

Float

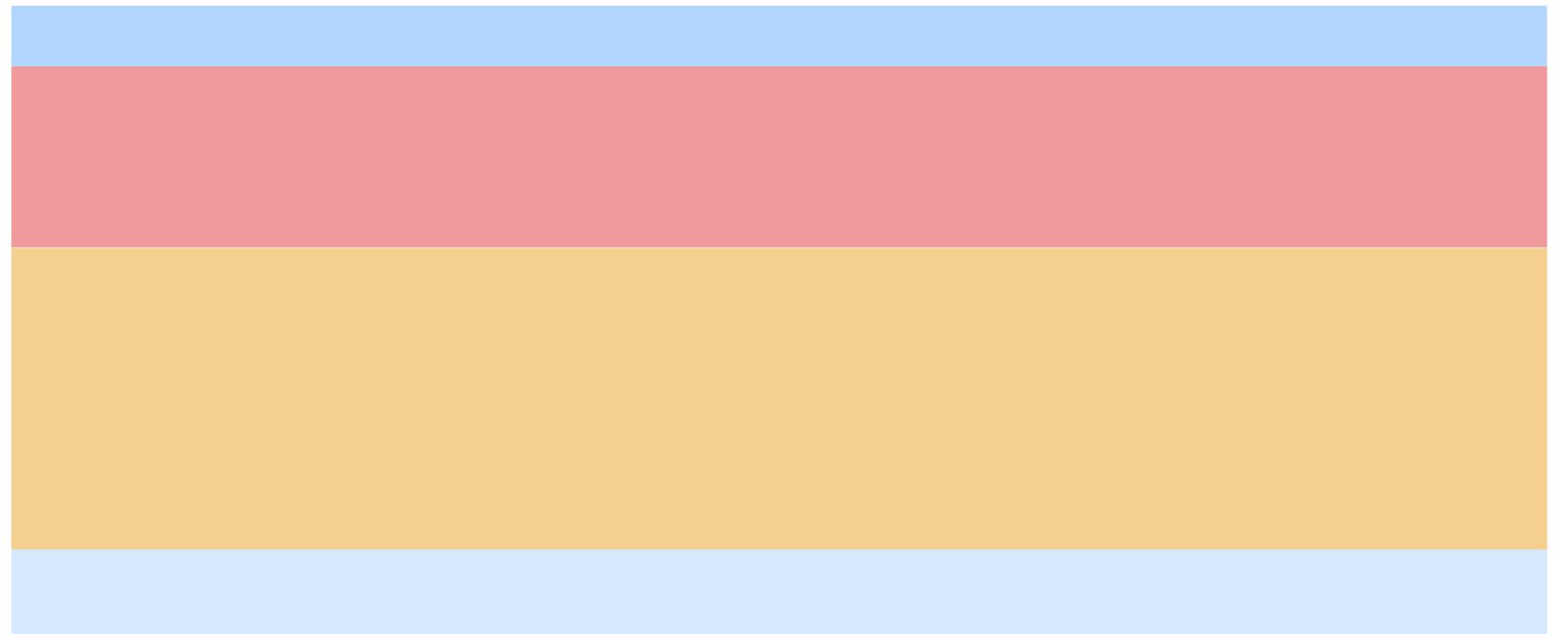
```
.menu {  
  height: 40px;  
  background-color: #B2D6FF;  
}
```

```
.sidebar {  
  height: 120px;  
  background-color: #F09A9D;  
}
```

```
.content {  
  height: 200px;  
  background-color: #F5CF8E;  
}
```

```
.footer {  
  height: 60px;  
  background-color: #D6E9FE;  
}
```

```
<div class="page">  
  <div class="menu"></div>  
  <div class="sidebar"></div>  
  <div class="content"></div>  
  <div class="footer"></div>  
</div>
```



Elementos flutuantes

Largura fixa

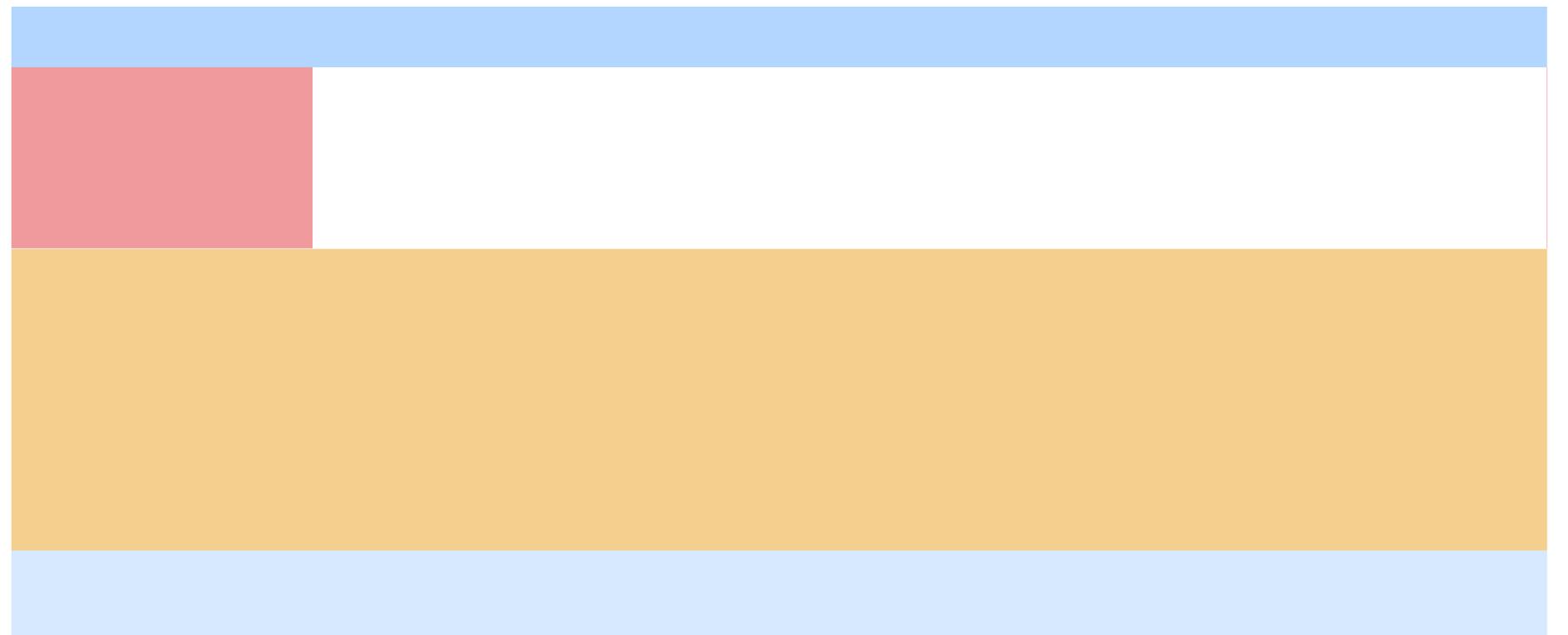
```
.menu {  
  height: 40px;  
  background-color: #B2D6FF;  
}
```

```
.sidebar {  
  height: 120px;  
  width: 200px;  
  background-color: #F09A9D;  
}
```

```
.content {  
  height: 200px;  
  background-color: #F5CF8E;  
}
```

```
.footer {  
  height: 60px;  
  background-color: #D6E9FE;  
}
```

```
<div class="page">  
  <div class="menu"></div>  
  <div class="sidebar"></div>  
  <div class="content"></div>  
  <div class="footer"></div>  
</div>
```



Elementos flutuantes

Float: left e right

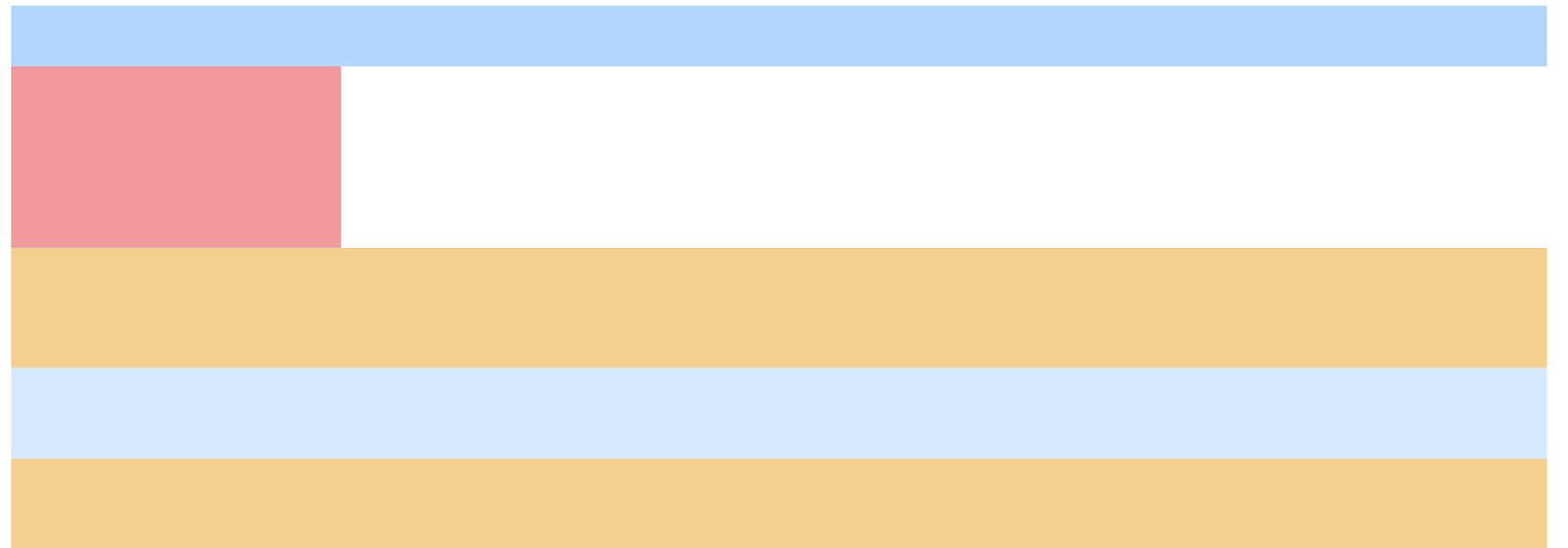
```
.menu {  
  height: 40px;  
  background-color: #B2D6FF;  
}
```

```
.sidebar {  
  float: right;  
  height: 120px;  
  width: 200px;  
  background-color: #F09A9D;  
}
```

```
.content {  
  height: 200px;  
  background-color: #F5CF8E;  
}
```

```
.footer {  
  height: 60px;  
  background-color: #D6E9FE;  
}
```

```
<div class="page">  
  <div class="menu"></div>  
  <div class="sidebar"></div>  
  <div class="content"></div>  
  <div class="footer"></div>  
</div>
```

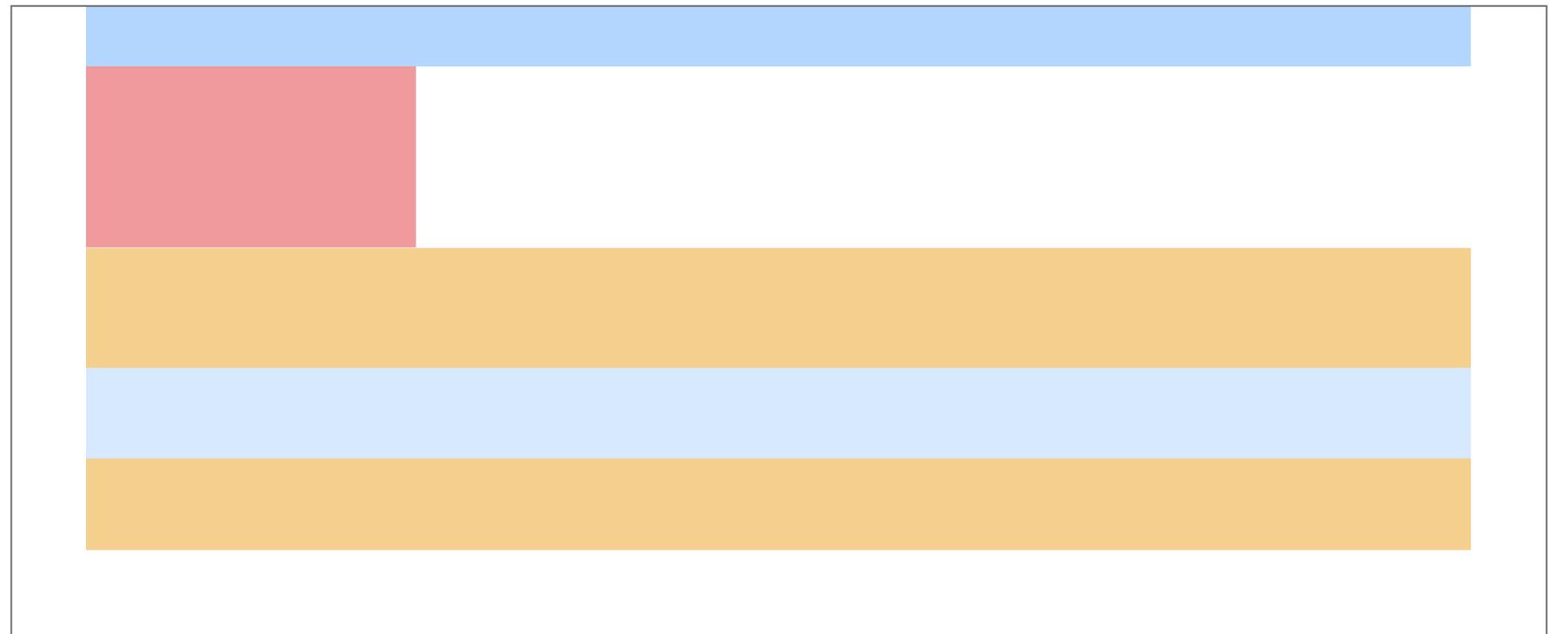


Elementos flutuantes

Float e margem

```
.page {  
  width: 900px;  
  margin: 0 auto;  
}  
  
.menu {  
  height: 40px;  
  background-color: #B2D6FF;  
}  
  
.sidebar {  
  float: left;  
  height: 120px;  
  width: 200px;  
  background-color: #F09A9D;  
}  
  
.content {  
  height: 200px;  
  background-color: #F5CF8E;  
}  
  
.footer {  
  height: 60px;  
  background-color: #D6E9FE;  
}
```

```
<div class="page">  
  <div class="menu"></div>  
  <div class="sidebar"></div>  
  <div class="content"></div>  
  <div class="footer"></div>  
</div>
```

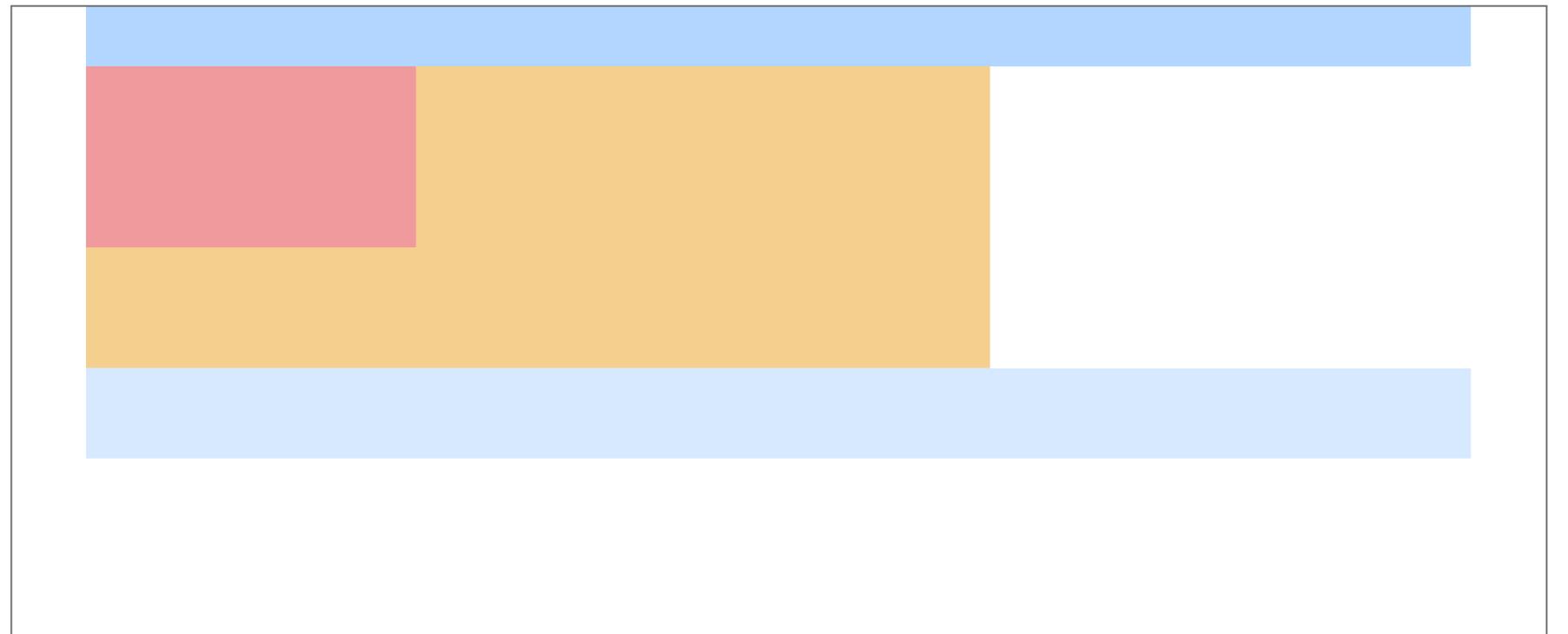


Elementos flutuantes

Float e largura fixa

```
.page {  
  width: 900px;  
  Margin: 0 auto;  
}  
.menu {  
  height: 40px;  
  background-color: #B2D6FF;  
}  
.sidebar {  
  float: left;  
  height: 120px;  
  width: 200px;  
  background-color: #F09A9D;  
}  
.content {  
  height: 200px;  
  width: 650px;  
  background-color: #F5CF8E;  
}  
.footer {  
  height: 60px;  
  background-color: #D6E9FE;  
}
```

```
<div class="page">  
  <div class="menu"></div>  
  <div class="sidebar"></div>  
  <div class="content"></div>  
  <div class="footer"></div>  
</div>
```

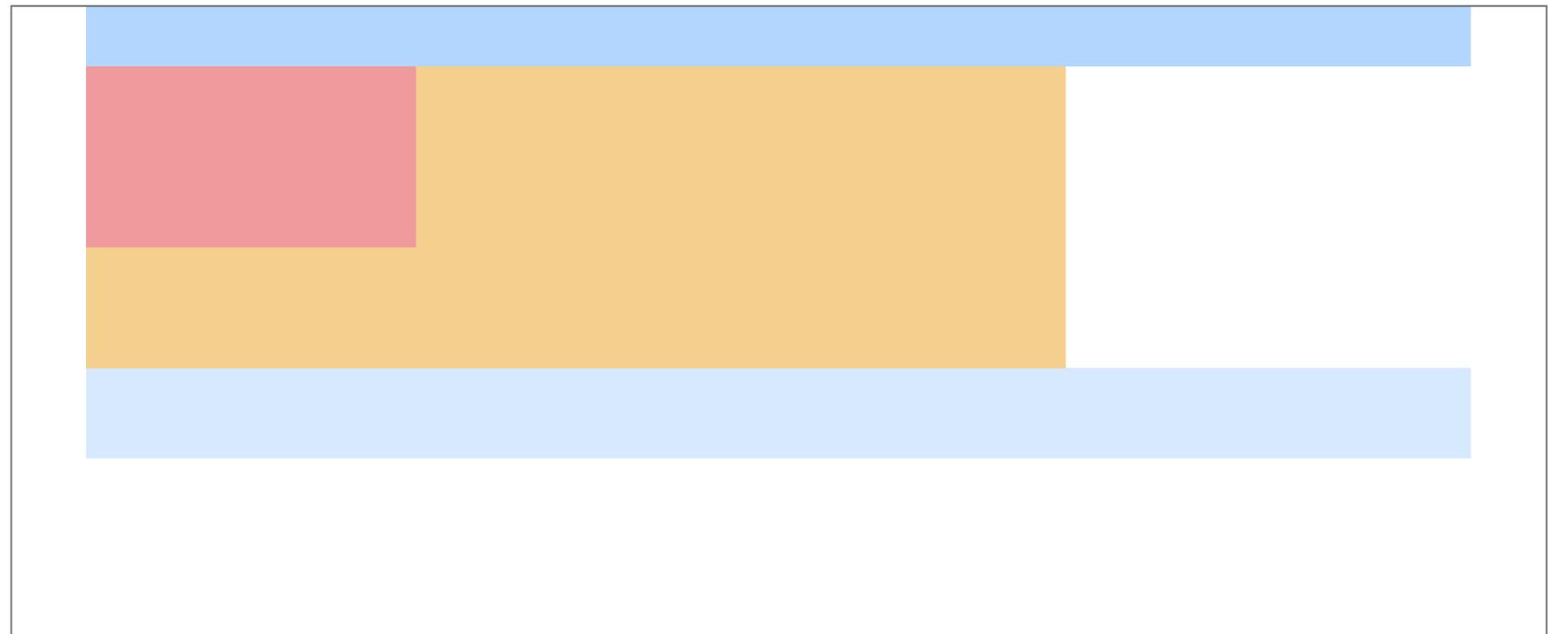


Elementos flutuantes

Múltiplos elementos

```
.page {  
  width: 900px;  
  margin: 0 auto;  
}  
.menu {  
  height: 40px;  
  background-color: #B2D6FF;  
}  
.sidebar {  
  float: left;  
  height: 120px;  
  width: 200px;  
  background-color: #F09A9D;  
}  
.content {  
  float: left;  
  height: 200px;  
  width: 650px;  
  background-color: #F5CF8E;  
}  
.footer {  
  height: 60px;  
  background-color: #D6E9FE;  
}
```

```
<div class="page">  
  <div class="menu"></div>  
  <div class="sidebar"></div>  
  <div class="content"></div>  
  <div class="footer"></div>  
</div>
```



Elementos flutuantes

Float e height

```
.page {  
  width: 900px;  
  margin: 0 auto;  
  border: 1px dashed red;  
}  
  
.menu {  
  height: 40px;  
  background-color: #B2D6FF;  
}  
  
.sidebar {  
  float: left;  
  height: 120px;  
  width: 200px;  
  background-color: #F09A9D;  
}  
  
.content {  
  float: left;  
  height: 200px;  
  width: 650px;  
  background-color: #F5CF8E;  
}  
  
.footer {  
  height: 60px;  
  background-color: #D6E9FE;  
}
```

```
<div class="page">  
  <div class="menu"></div>  
  <div class="sidebar"></div>  
  <div class="content"></div>  
  <div class="footer"></div>  
</div>
```

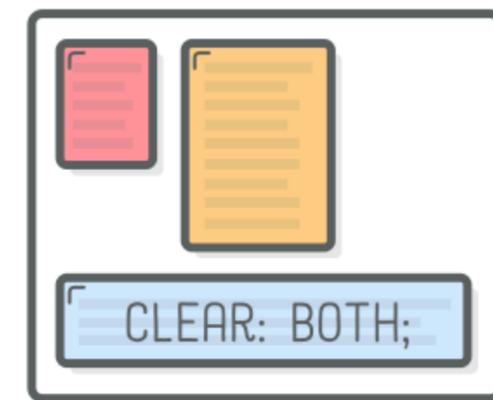


Elementos flutuantes

A propriedade clear

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

CLEARING WITH CHILD ELEMENT



Fonte: [Interneting is Hard](#)

Elementos flutuantes

A propriedade clear

```
.page {  
  width: 900px;  
  margin: 0 auto;  
  border: 1px dashed red;  
}  
.menu {  
  height: 40px;  
  background-color: #B2D6FF;  
}  
.sidebar {  
  float: left;  
  height: 120px;  
  width: 200px;  
  background-color: #F09A9D;  
}  
.content {  
  float: left;  
  height: 200px;  
  width: 650px;  
  background-color: #F5CF8E;  
}  
.footer {  
  clear: both;  
  height: 60px;  
  background-color: #D6E9FE;  
}
```

```
<div class="page">  
  <div class="menu"></div>  
  <div class="sidebar"></div>  
  <div class="content"></div>  
  <div class="footer"></div>  
</div>
```

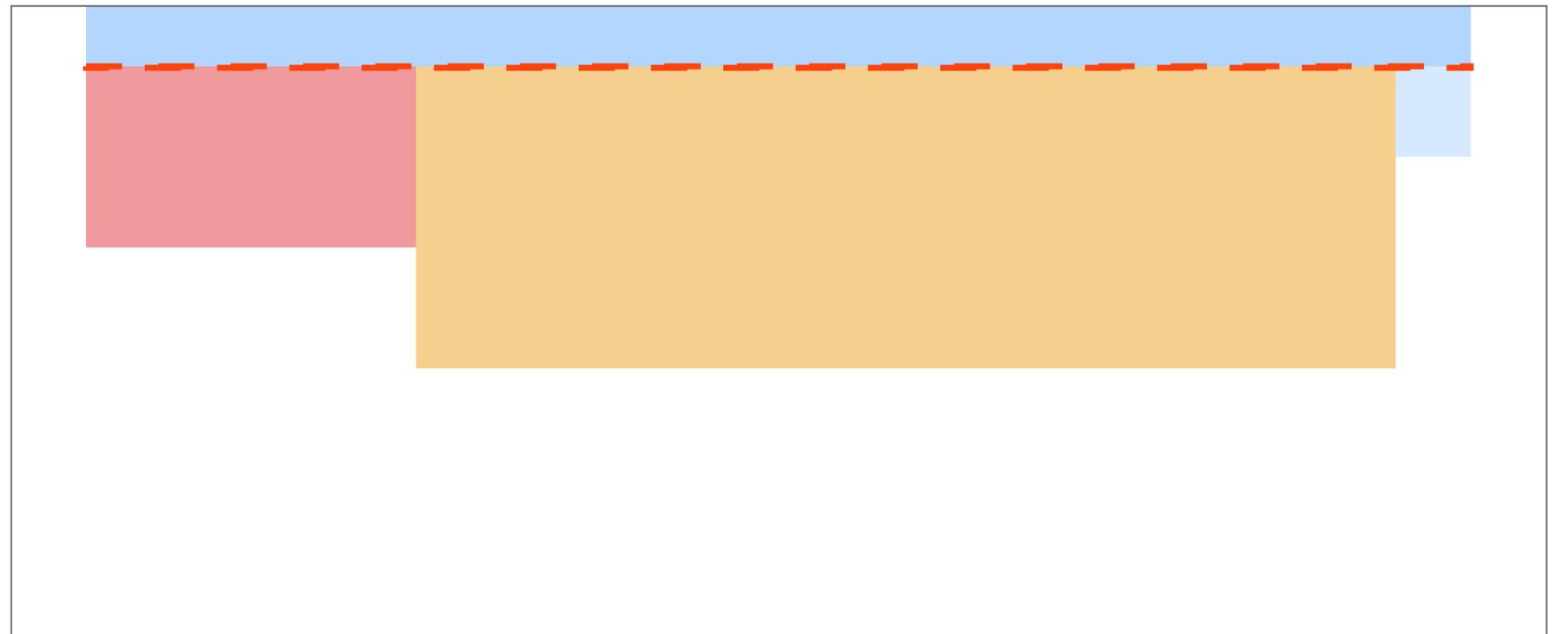


Elementos flutuantes

A propriedade clear

```
.page {
  width: 900px;
  margin: 0 auto;
  border: 1px dashed red;
}
.menu {
  height: 40px;
  background-color: #B2D6FF;
}
.sidebar {
  float: left;
  height: 120px;
  width: 200px;
  background-color: #F09A9D;
}
.content {
  float: left;
  height: 200px;
  width: 650px;
  background-color: #F5CF8E;
}
.footer {
  clear: both;
  height: 60px;
  background-color: #D6E9FE;
}
```

```
<div class='menu'>Menu</div>
<div class='page'>
  <div class='sidebar'>Sidebar</div>
  <div class='content'>Content</div>
</div>
<div class='footer'>Footer</div>
```



Elementos flutuantes

A propriedade 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**

CLEARING WITH PARENT ELEMENT



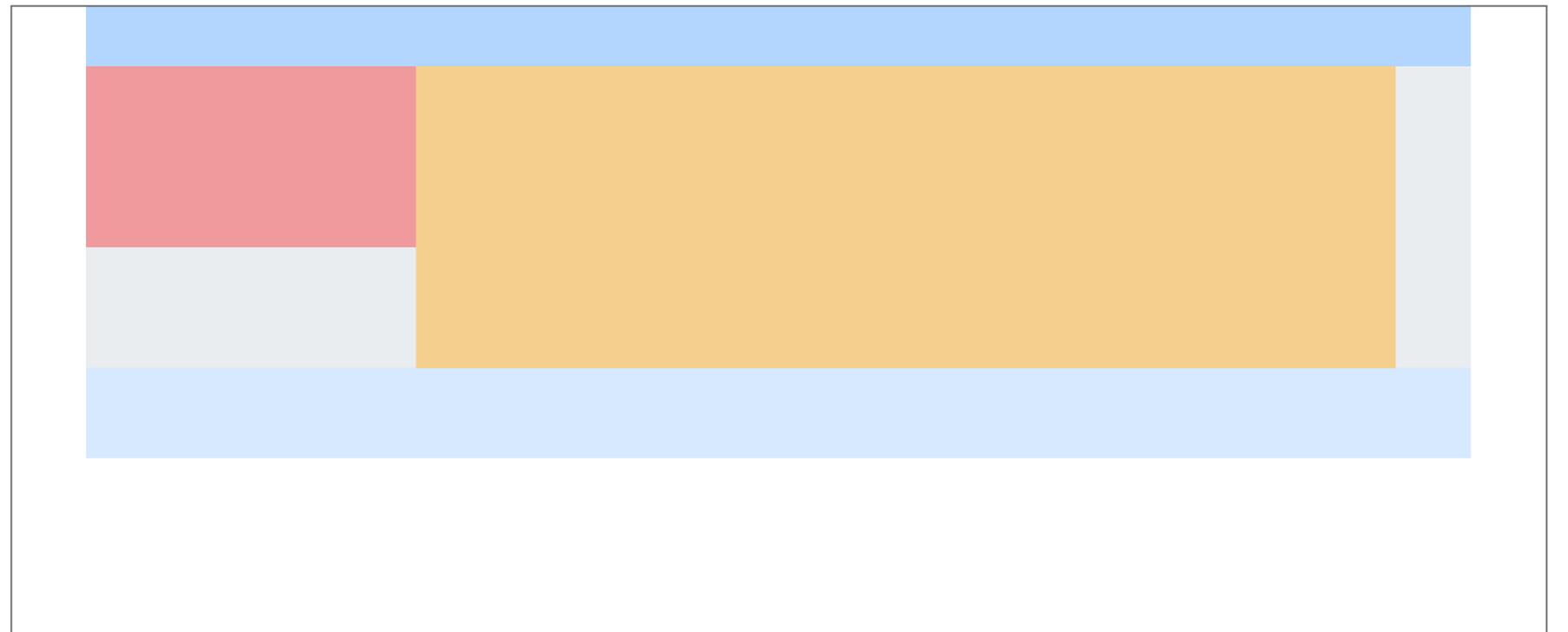
Fonte: [Interneting is Hard](#)

Elementos flutuantes

A propriedade overflow

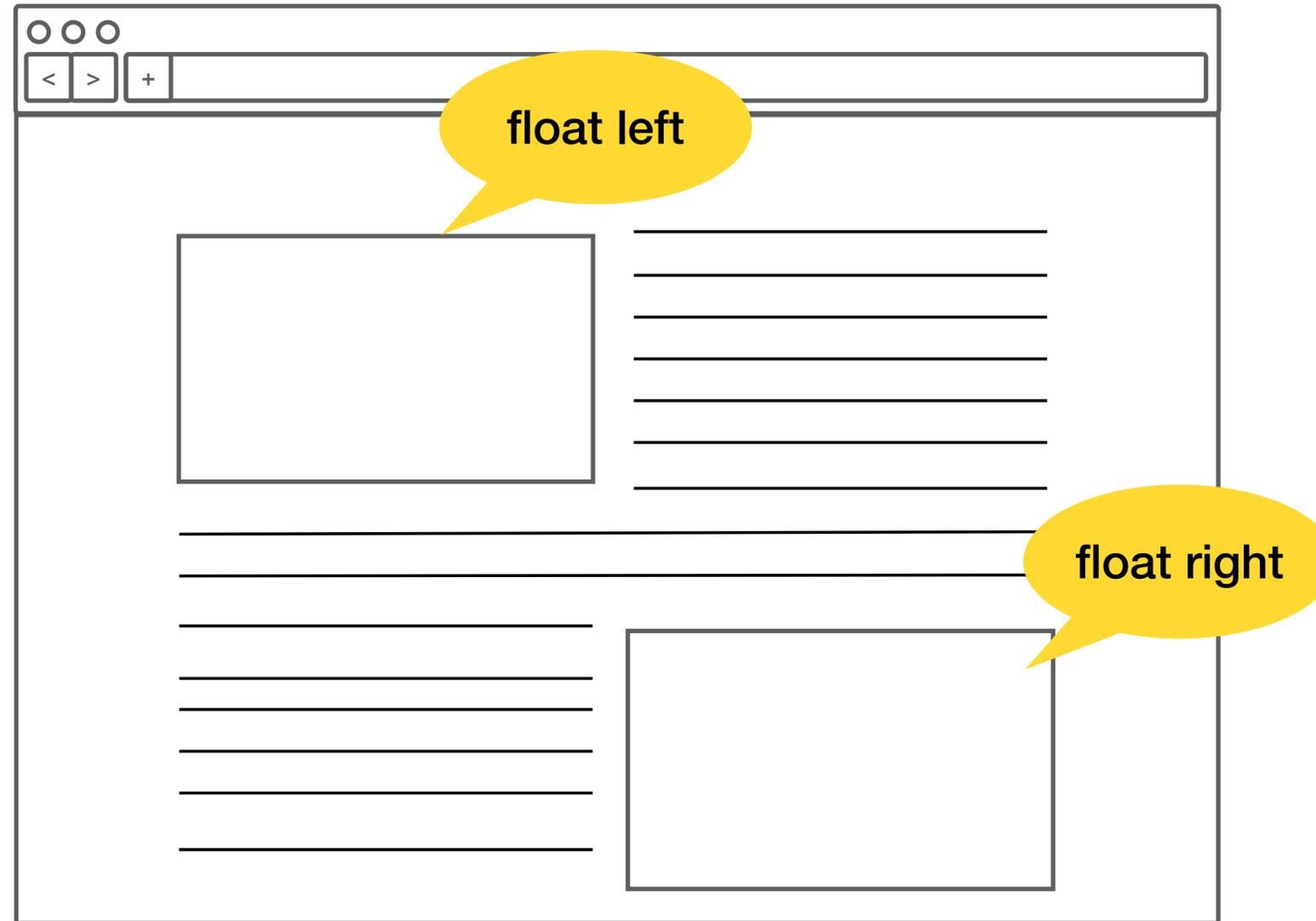
```
.page {  
  width: 900px;  
  margin: 0 auto;  
  overflow: hidden;  
  background-color: #EAEDF0;  
}  
.menu {  
  height: 40px;  
  background-color: #B2D6FF;  
}  
.sidebar {  
  float: left;  
  height: 120px;  
  width: 200px;  
  background-color: #F09A9D;  
}  
.content {  
  float: left;  
  height: 200px;  
  width: 650px;  
  background-color: #F5CF8E;  
}  
.footer {  
  clear: both;  
  height: 60px;  
  background-color: #D6E9FE;  
}
```

```
<div class='menu'>Menu</div>  
<div class='page'>  
  <div class='sidebar'>Sidebar</div>  
  <div class='content'>Content</div>  
</div>  
<div class='footer'>Footer</div>
```

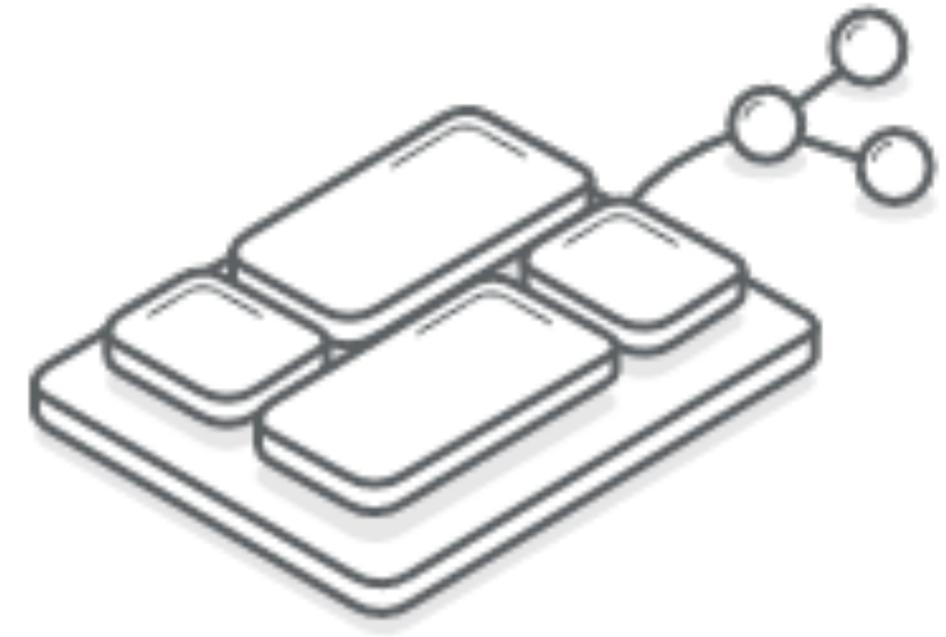


Elementos flutuantes

Uso em conteúdo

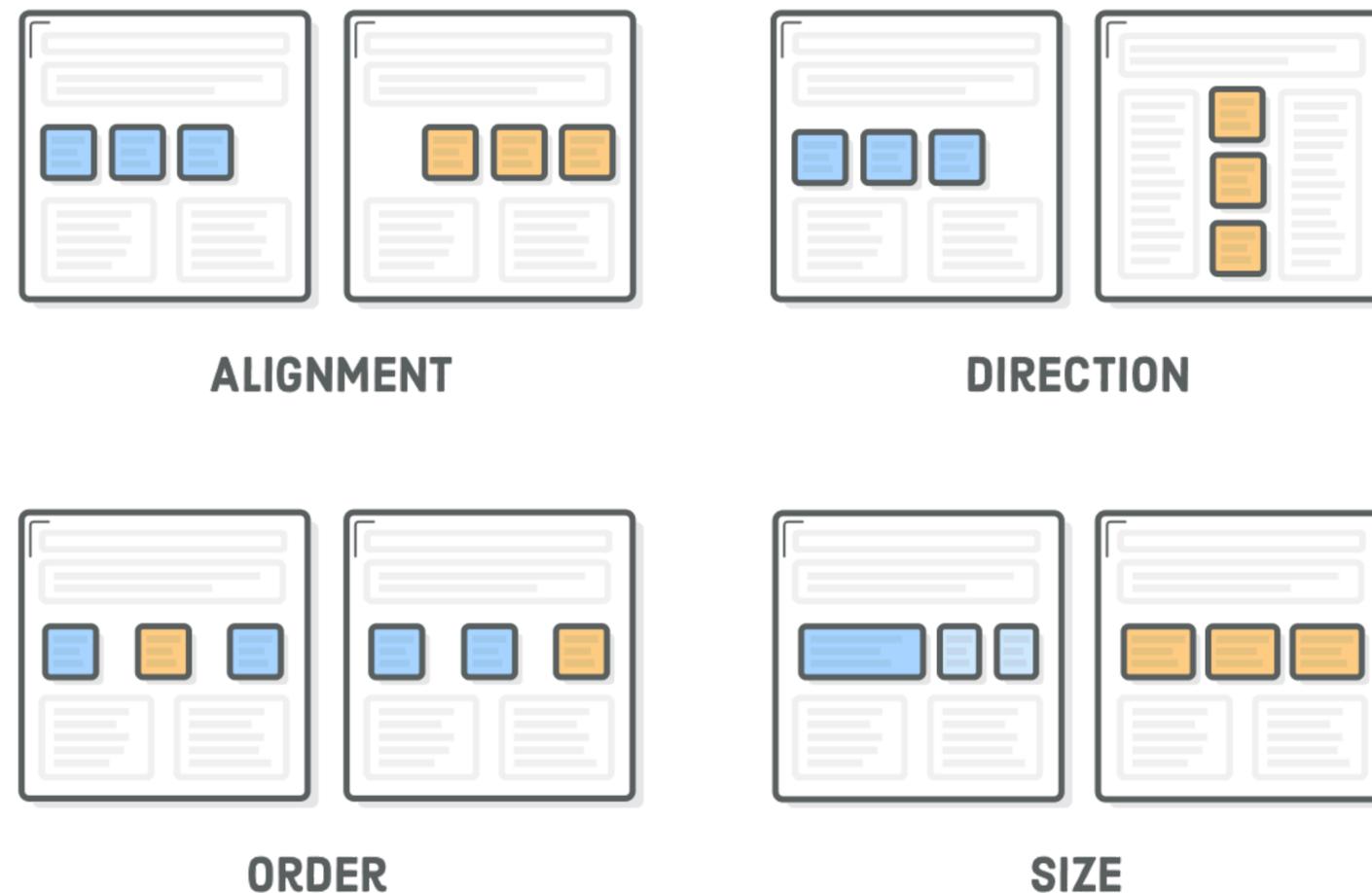


Flexbox



Flexbox

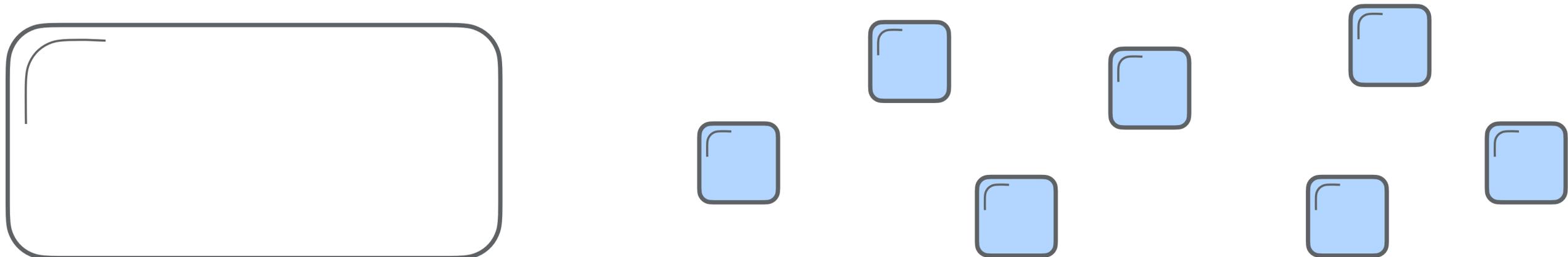
- O “*Flexible Box*” or “*Flexbox*” é uma alternativa a construção de layouts usando elementos flutuantes
 - Adicionado no CSS 3
- Permite o total controle sobre:
 - Direção
 - Alinhamento
 - Ordem
 - Tamanho



Flexbox

Contêineres e items

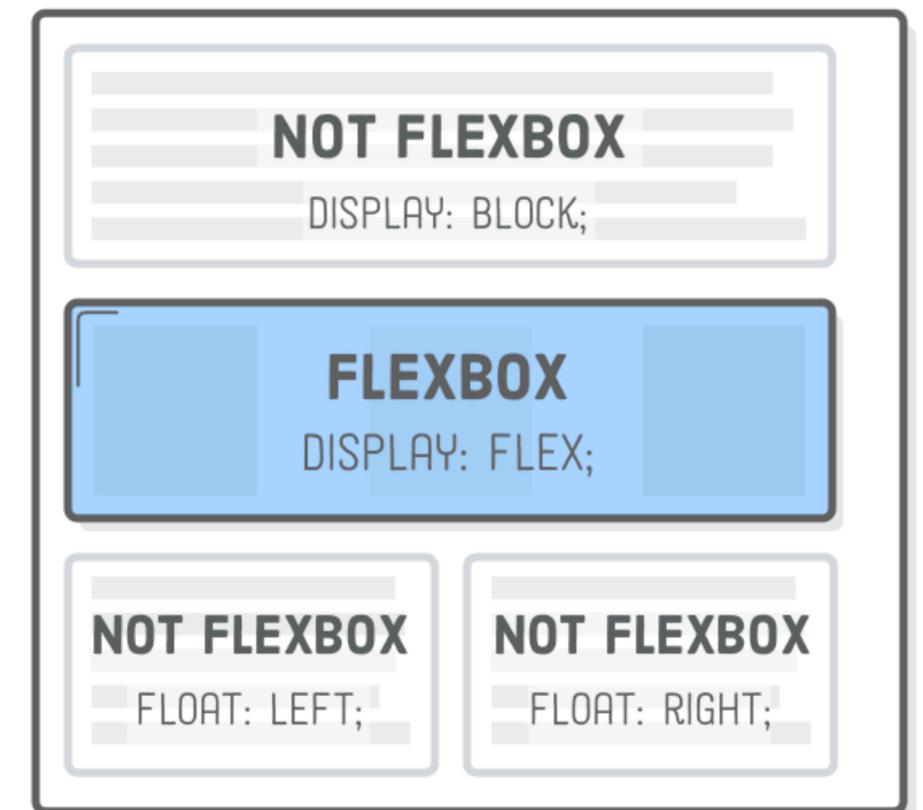
- Flexbox utiliza dois tipos “caixas”:
 - Contêineres
 - Items
- Contêineres são responsáveis por mostrar items e posicioná-los
 - Um **item** é qualquer elemento HTML que é filho direto de um **contêiner**



Flexbox

Contêiner

- O primeiro passo para utilizar o Flexbox é definir um contêiner
 - Usamos a propriedade **display: flex**
- Ainda assim é possível mesclar contêineres flex e contêineres regulares



Fonte: [Interneting is Hard](#)

Flexbox

- Vamos mostrar todo o potencial do uso do Flexbox para construir o layout a seguir:



Elementos flutuantes

justify-content

```
.menu-container {  
  color: #fff;  
  background-color: #5995DA;  
  padding: 20px 0;  
  display: flex;  
  justify-content: center;  
}
```

```
.menu {  
  border: 1px solid #fff;  
  width: 700px;  
}
```

```
<div class='menu-container'>  
  <div class='menu'>  
    <div class='date'>17 de março 2022</div>  
    <div class='signup'>Sign Up</div>  
    <div class='login'>Login</div>  
  </div>  
</div>
```

17 de março 2022

Sign Up

Login

Elementos flutuantes

justify-content

```
.menu-container {  
  color: #fff;  
  background-color: #5995DA;  
  padding: 20px 0;  
  display: flex;  
  justify-content: center;  
}
```

```
.menu {  
  border: 1px solid #fff;  
  width: 700px;  
  display: flex;  
  justify-content: space-between;  
}
```

```
<div class='menu-container'>  
  <div class='menu'>  
    <div class='date'>17 de março 2022</div>  
    <div class='signup'>Sign Up</div>  
    <div class='login'>Login</div>  
  </div>  
</div>
```

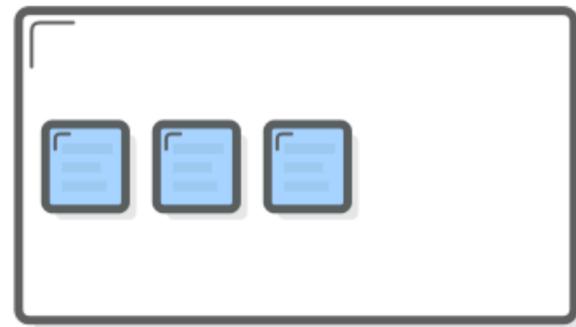
17 de março 2022

Sign Up

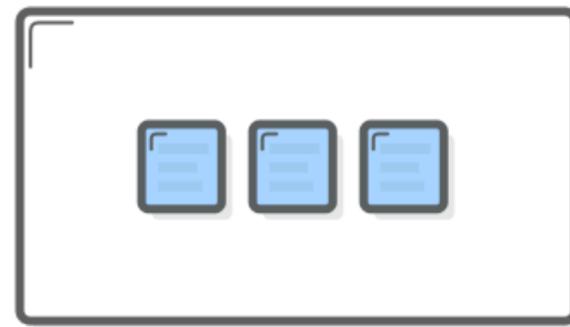
Login

Elementos flutuantes

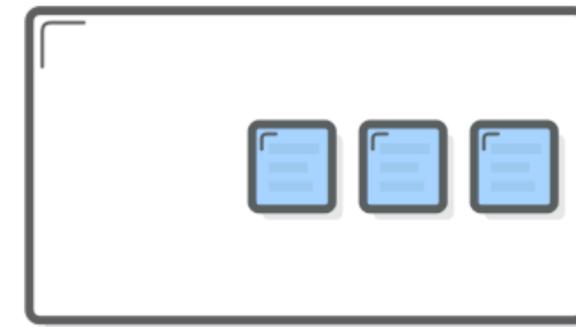
justify-content



FLEX-START



CENTER



FLEX-END



SPACE-AROUND



SPACE-BETWEEN

Fonte: [Interneting is Hard](#)

Elementos flutuantes

Agrupando elementos

```
.menu-container {  
  color: #fff;  
  background-color: #5995DA;  
  padding: 20px 0;  
  display: flex;  
  justify-content: center;  
}
```

```
.menu {  
  border: 1px solid #fff;  
  width: 700px;  
  display: flex;  
  justify-content: space-between;  
}
```

```
<div class='menu-container'>  
  <div class='menu'>  
    <div class='date'>17 de março 2022</div>  
    <div class='links'>  
      <div class='signup'>Sign Up</div>  
      <div class='login'>Login</div>  
    </div>  
  </div>  
</div>
```

17 de março 2022

Sign Up
Login

Elementos flutuantes

Agrupando elementos

```
.menu-container {  
  color: #fff;  
  background-color: #5995DA;  
  padding: 20px 0;  
  display: flex;  
  justify-content: center;  
}
```

```
.menu {  
  border: 1px solid #fff;  
  width: 700px;  
  display: flex;  
  justify-content: space-between;  
}
```

```
.links {  
  display: flex;  
  justify-content: flex-end;  
}
```

```
<div class='menu-container'>  
  <div class='menu'>  
    <div class='date'>17 de março 2022</div>  
    <div class='links'>  
      <div class='signup'>Sign Up</div>  
      <div class='login'>Login</div>  
    </div>  
  </div>  
</div>
```

17 de março 2022

Sign Up Login

Elementos flutuantes

Alinhamento vertical

```
.header-container {  
  color: #5995DA;  
  background-color: #D6E9FE;  
  display: flex;  
  justify-content: center;  
}  
  
.header {  
  width: 700px;  
  height: 200px;  
  display: flex;  
  justify-content: space-between;  
}
```

```
<div class='header-container'>  
  <div class='header'>  
    <div class='subscribe'>Assinar</div>  
    <div class='logo'><img/></div>  
    <div class='social'><img/></div>  
  </div>  
</div>
```

Assinar

Awesome Logo

Ícones

Elementos flutuantes

Agrupando elementos

```
.header-container {  
  color: #5995DA;  
  background-color: #D6E9FE;  
  display: flex;  
  justify-content: center;  
}
```

```
.header {  
  width: 700px;  
  height: 200px;  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
}
```

```
<div class='header-container'>  
  <div class='header'>  
    <div class='subscribe'>Assinar</div>  
    <div class='logo'><img/></div>  
    <div class='social'><img/></div>  
  </div>  
</div>
```

Assinar

Awesome Logo

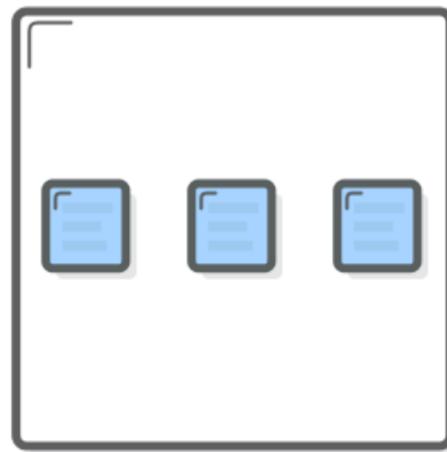
Ícones

Elementos flutuantes

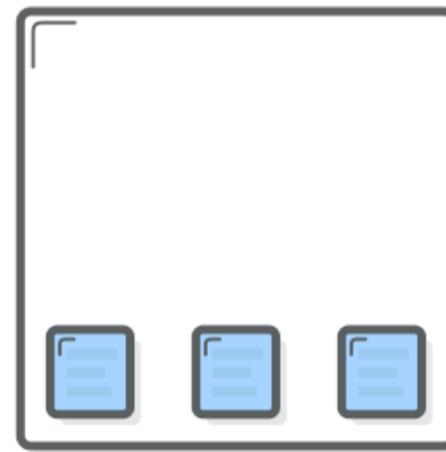
align-items



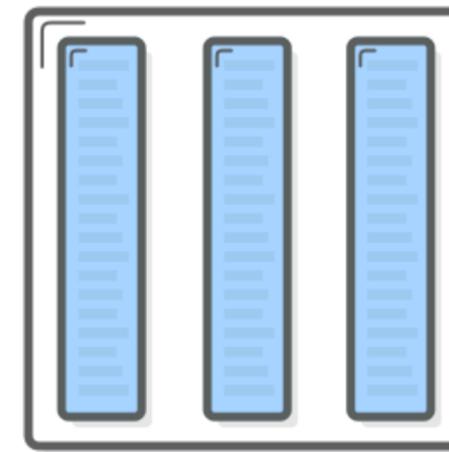
FLEX-START



CENTER



FLEX-END



STRETCH

Fonte: [Interneting is Hard](#)

Elementos flutuantes

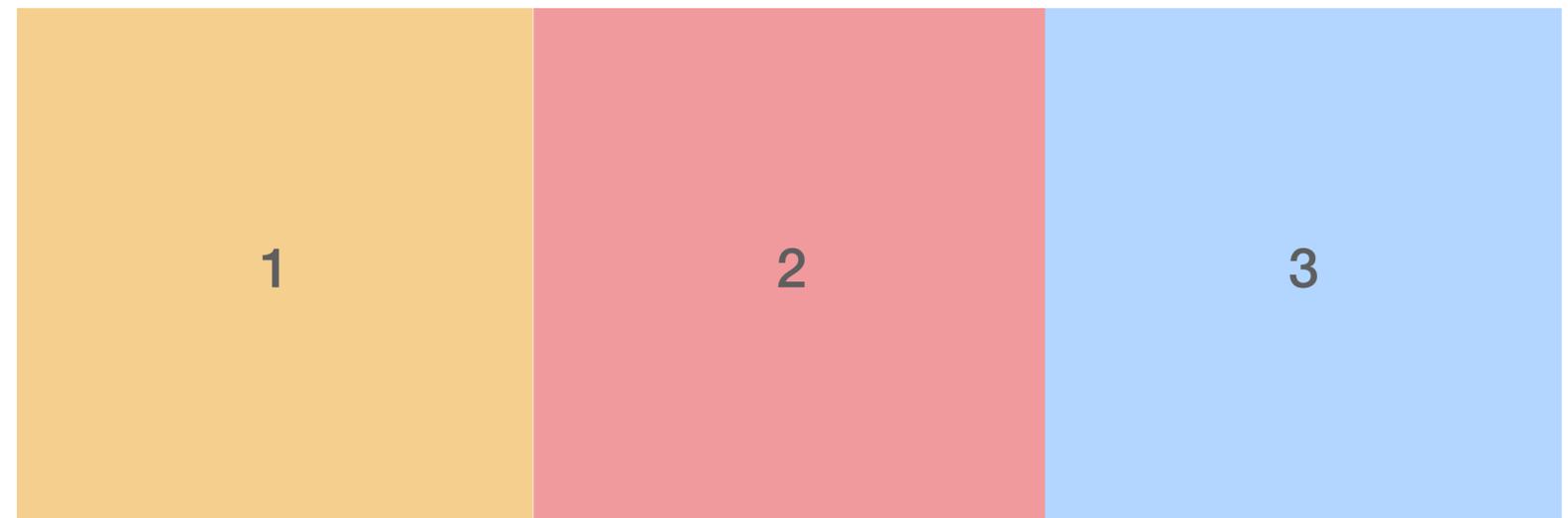
Agrupando elementos

```
.photo-grid-container {  
  display: flex;  
  justify-content: center;  
}
```

```
.photo-grid {  
  width: 200px;  
  display: flex;  
  justify-content: flex-start;  
}
```

```
.photo-grid-item {  
  border: 1px solid #fff;  
  width: 200px;  
  height: 200px;  
}
```

```
<div class='photo-grid-container'>  
  <div class='photo-grid'>  
    <div class='photo-grid-item'><img/></div>  
    <div class='photo-grid-item'><img/></div>  
    <div class='photo-grid-item'><img/></div>  
  </div>  
</div>
```

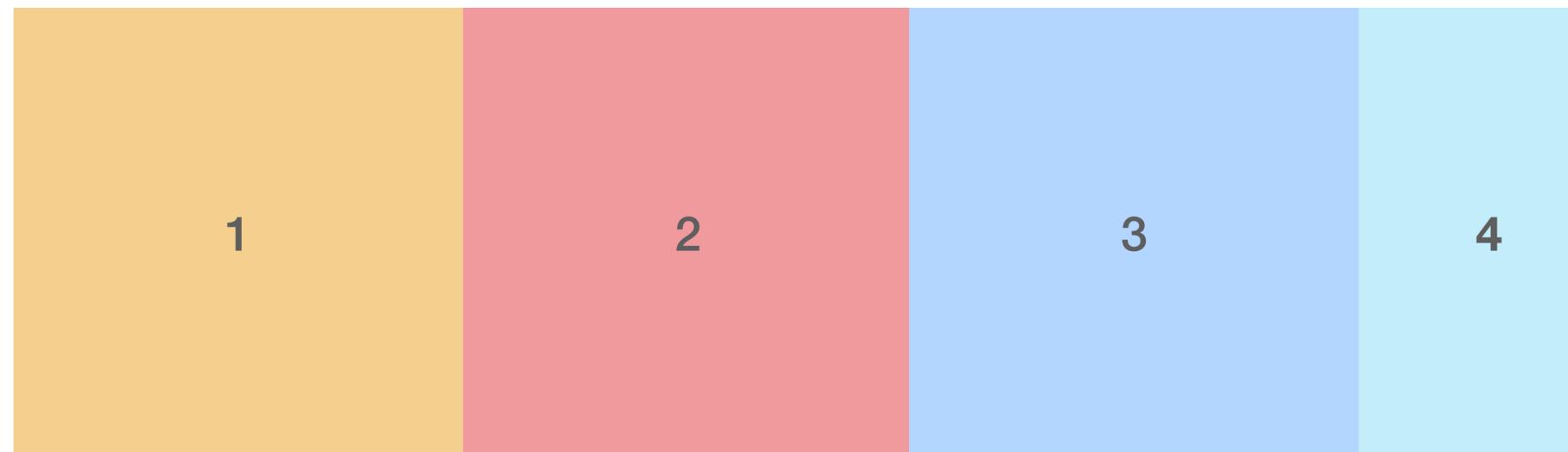


Elementos flutuantes

Agrupando elementos

```
.photo-grid-container {  
  display: flex;  
  justify-content: center;  
}  
  
.photo-grid {  
  width: 200px;  
  display: flex;  
  justify-content: flex-start;  
}  
  
.photo-grid-item {  
  border: 1px solid #fff;  
  width: 200px;  
  height: 200px;  
}
```

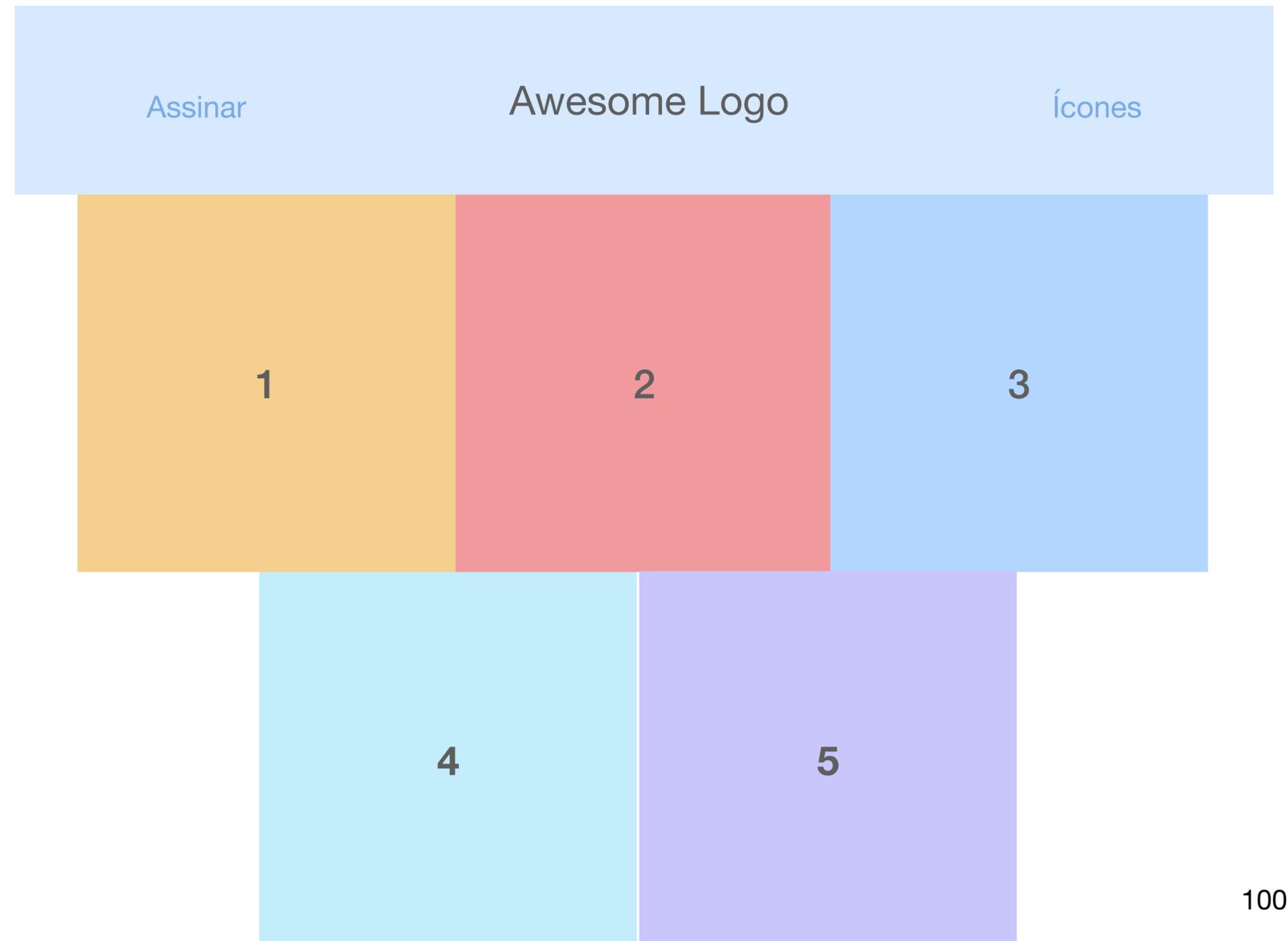
```
<div class='photo-grid-container'>  
  <div class='photo-grid'>  
    <div class='photo-grid-item'><img/></div>  
    <div class='photo-grid-item'><img/></div>  
    <div class='photo-grid-item'><img/></div>  
    <div class='photo-grid-item'><img/></div>  
  </div>  
</div>
```



Elementos flutuantes

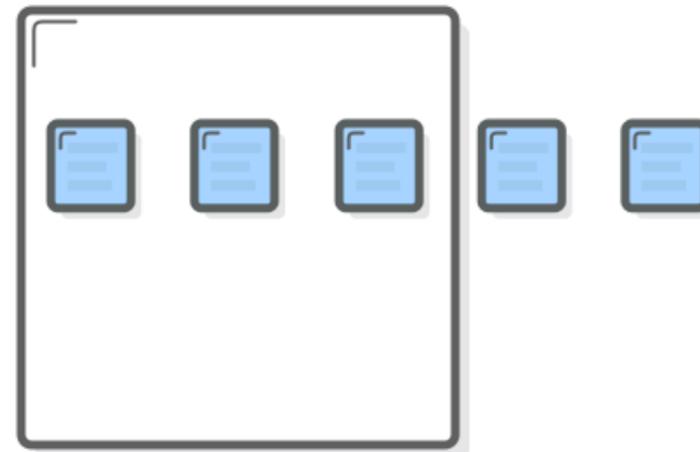
Agrupando elementos

```
.photo-grid-container {  
  display: flex;  
  justify-content: center;  
}  
.photo-grid {  
  width: 200px;  
  display: flex;  
  flex-wrap: wrap;  
  justify-content: center;  
}  
.photo-grid-item {  
  border: 1px solid #fff;  
  width: 200px;  
  height: 200px;  
}
```



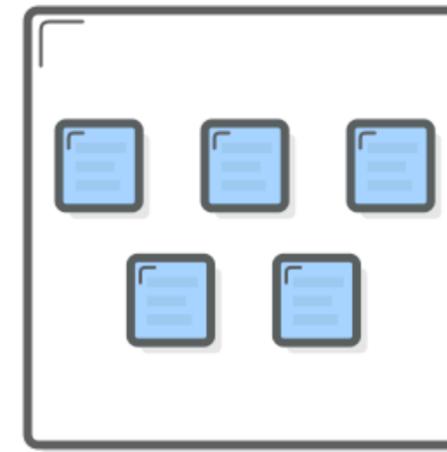
Elementos flutuantes

flex-wrap



NO WRAPPING

FLEX-WRAP: NOWRAP;



WITH WRAPPING

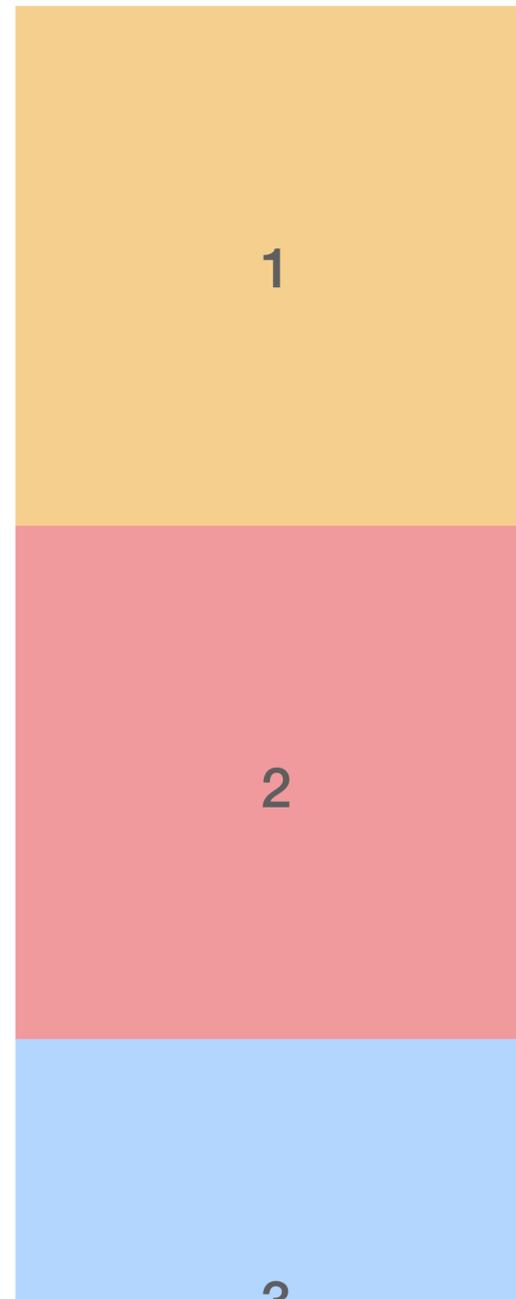
FLEX-WRAP: WRAP;

Fonte: [Interneting is Hard](#)

Elementos flutuantes

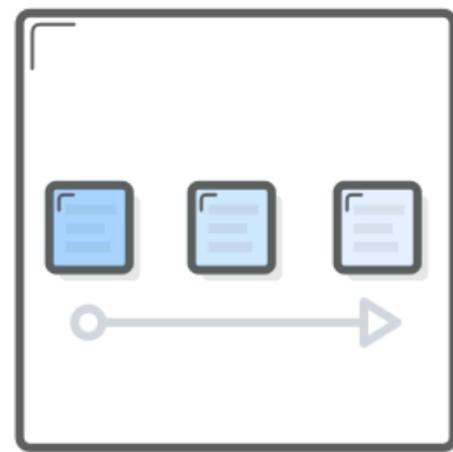
Direção do contêiner

```
.photo-grid-container {  
  display: flex;  
  justify-content: center;  
}  
  
.photo-grid {  
  width: 200px;  
  display: flex;  
  flex-wrap: wrap;  
  flex-direction: column;  
  justify-content: center;  
}  
  
.photo-grid-item {  
  border: 1px solid #fff;  
  width: 200px;  
  height: 200px;  
}
```

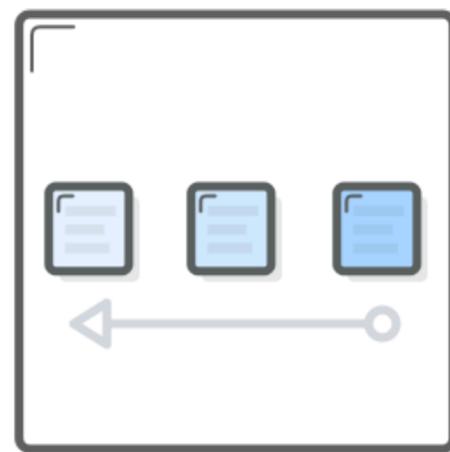


Elementos flutuantes

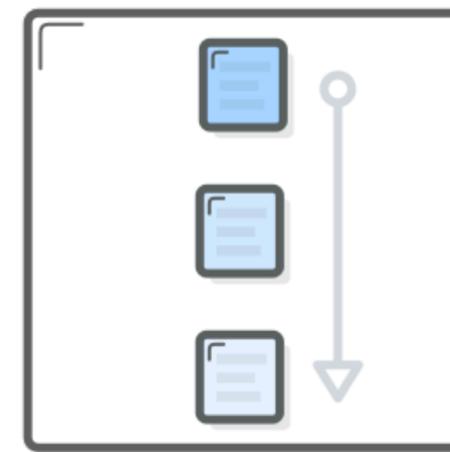
flex-direction



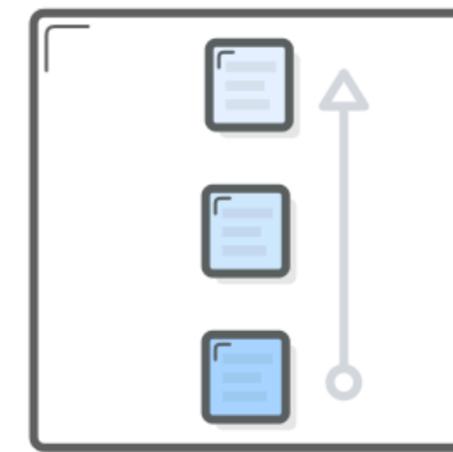
ROW



ROW-REVERSE



COLUMN



COLUMN-REVERSE

Fonte: [Interneting is Hard](#)

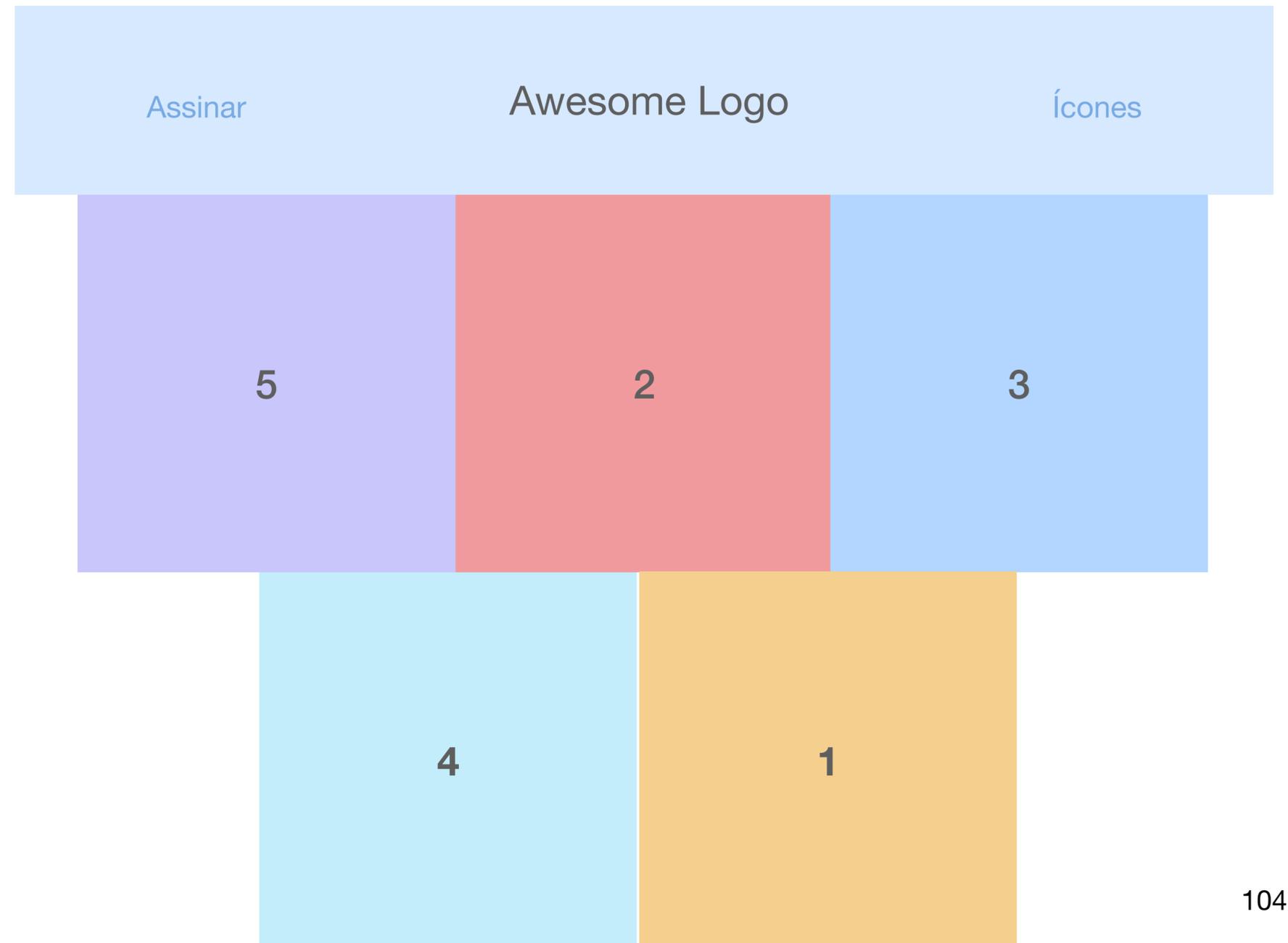
Elementos flutuantes

Ordem do contêiner

```
.photo-grid {  
  width: 200px;  
  display: flex;  
  flex-wrap: wrap;  
  flex-direction: row;  
  justify-content: center;  
}
```

```
.first-item {  
  order: 1;  
}
```

```
.last-item {  
  order: -1;  
}
```



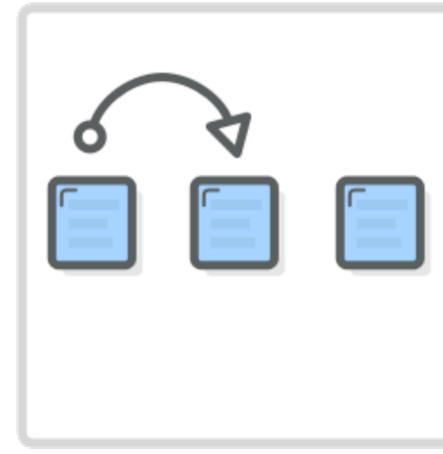
Elementos flutuantes

order



FLEX-DIRECTION

(WHOLE CONTAINER)



ORDER

(INDIVIDUAL ITEMS)

Fonte: [Interneting is Hard](#)

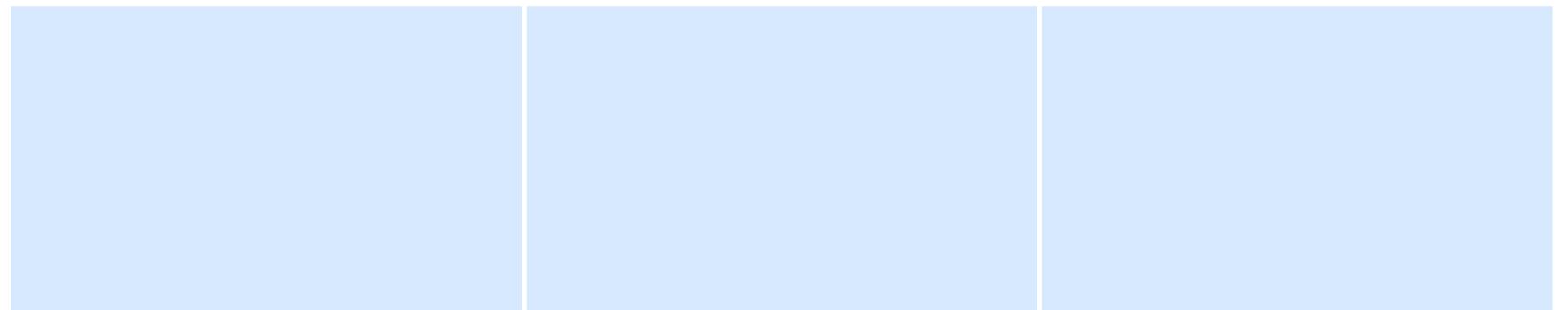
Elementos flutuantes

Tamanhos flexíveis

```
.footer {  
  display: flex;  
  justify-content: space-between;  
}
```

```
.footer-item {  
  border: 1px solid #fff;  
  background-color: #D6E9FE;  
  height: 200px;  
  flex: 1;  
}
```

```
<div class='footer'>  
  <div class='footer-item footer-one'></div>  
  <div class='footer-item footer-two'></div>  
  <div class='footer-item footer-three'></div>  
</div>
```



Elementos flutuantes

Tamanhos flexíveis

```
.footer {  
  display: flex;  
  justify-content: space-between;  
}
```

```
.footer-item {  
  border: 1px solid #fff;  
  background-color: #D6E9FE;  
  height: 200px;  
  flex: 1;  
}
```

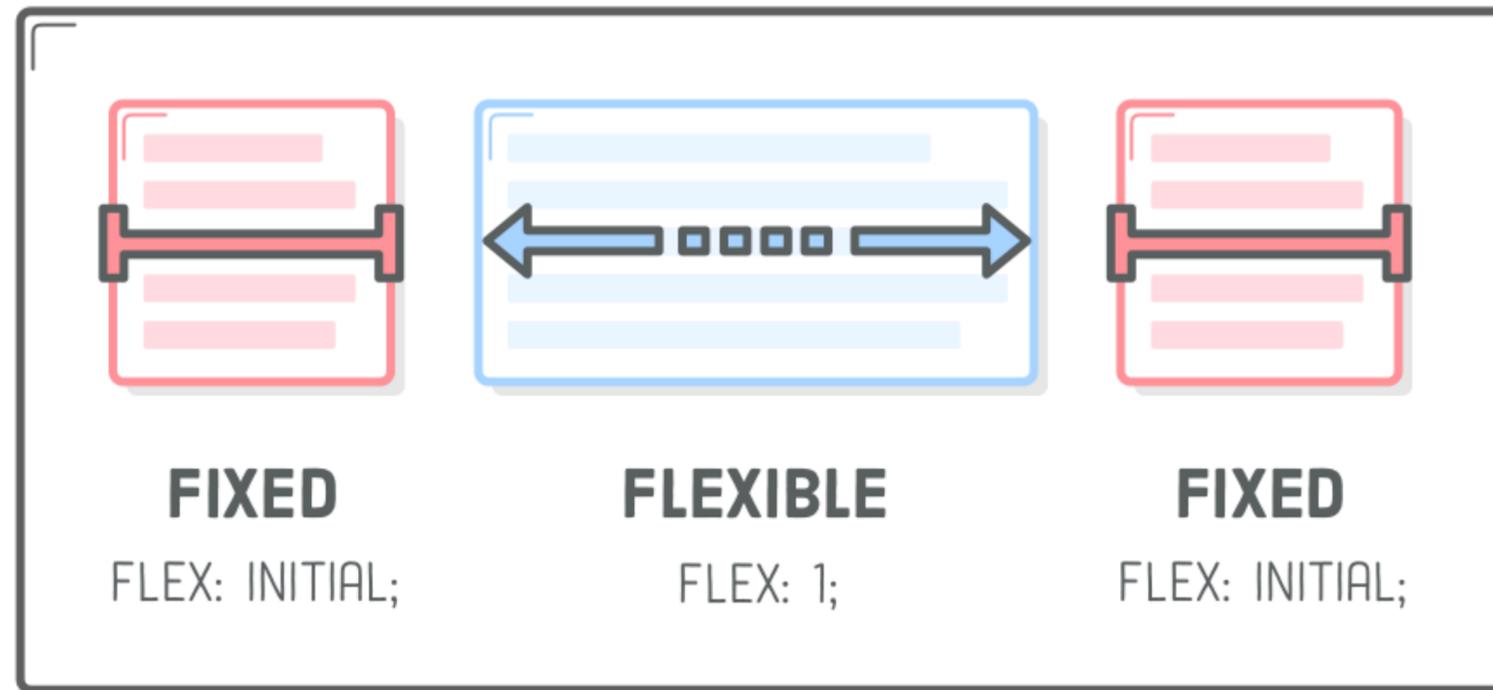
```
.footer-three {  
  flex: 2;  
  background-color: #5995DA;  
}
```

```
<div class='footer'>  
  <div class='footer-item footer-one'></div>  
  <div class='footer-item footer-two'></div>  
  <div class='footer-item footer-three'></div>  
</div>
```



Elementos flutuantes

flex-direction



Fonte: [Interneting is Hard](#)

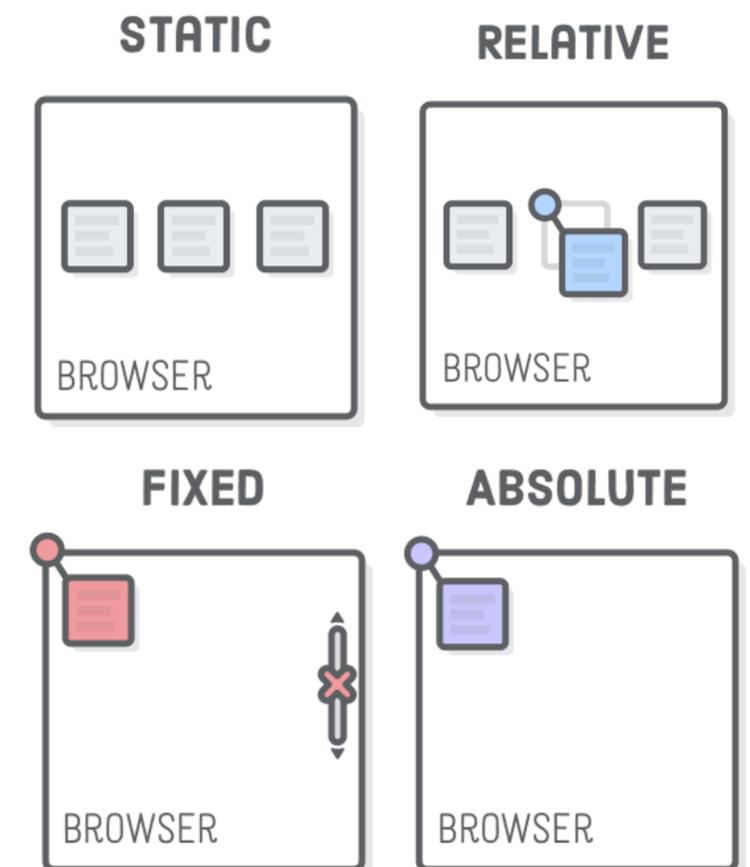
Posicionamiento



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 (static)
fixed	Define a posição em relação a janela do navegador
absolute	Calcula a posição do elemento em relação ao elemento superior do primeiro ancestral não static



Posicionamento

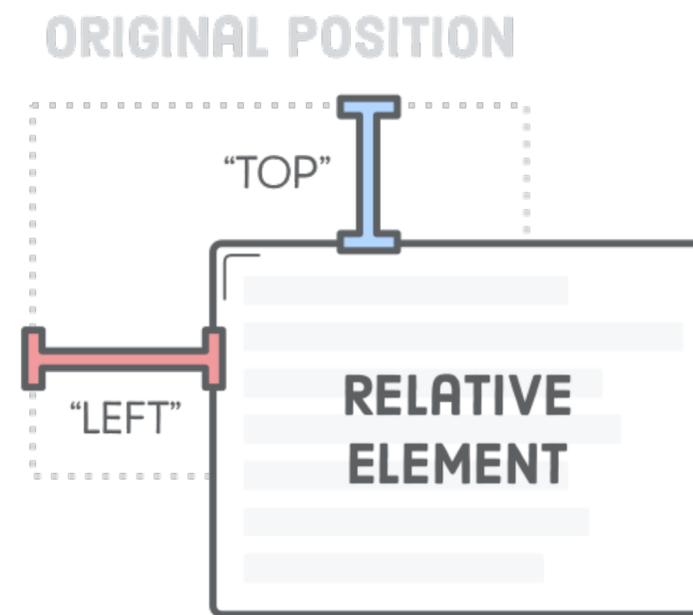
Static vs Relative

```
.parent {  
  color: blue;  
  background-color: gray;  
  width: 500px;  
  height: 500px;  
}
```

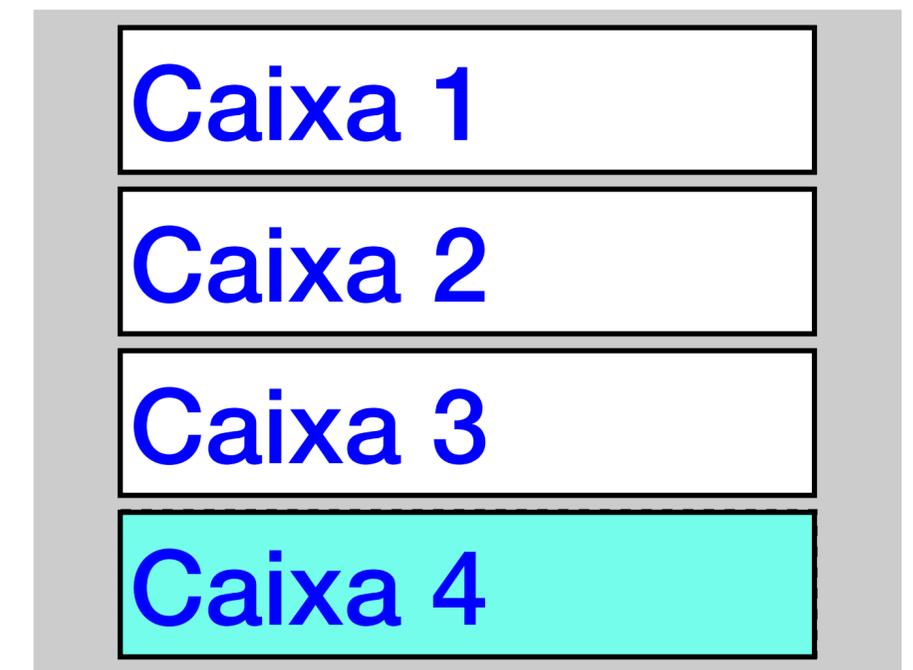
```
.special.box {  
  position: relative;  
  top: 20px;  
  left: 20px;  
  background-color: cyan;  
}
```

```
.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>
```



Fonte: [Interneting is Hard](#)



Posicionamento

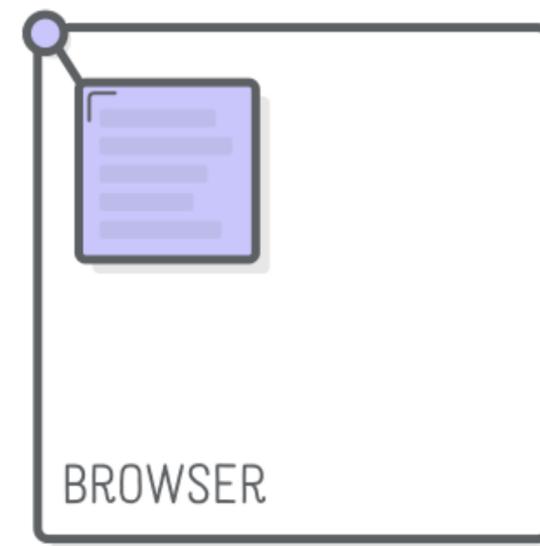
Absolute

```
.parent {  
  color: blue;  
  background-color: gray;  
  width: 500px;  
  height: 500px;  
}
```

```
.special.box {  
  position: absolute;  
  top: 20px;  
  left: 20px;  
  background-color: cyan;  
}
```

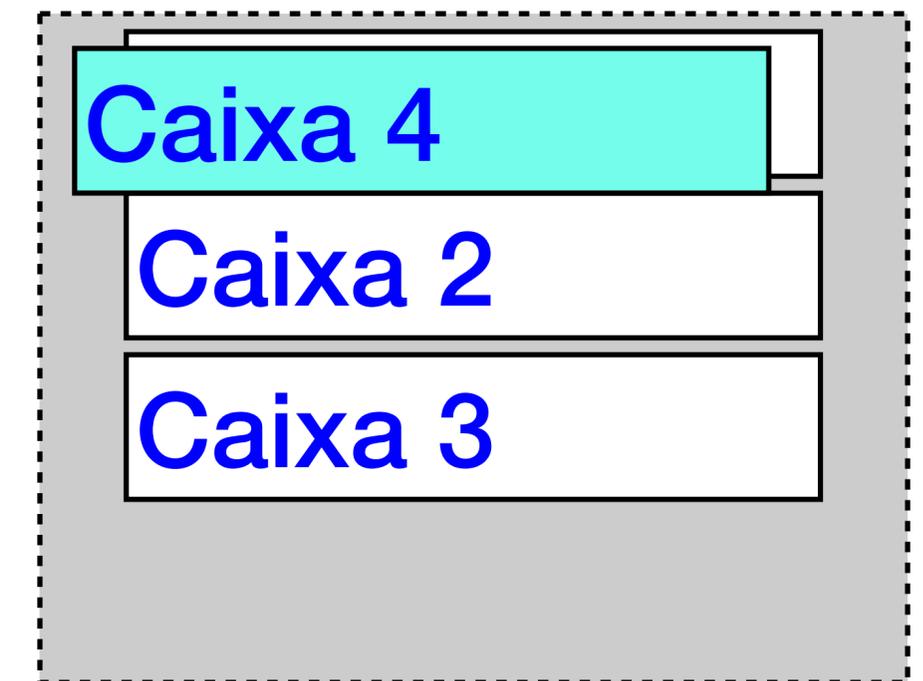
```
.box {  
  font-size: 2em;  
  background-color: white;  
  width: 400px;  
  border: 1px solid black;  
  margin: 10px auto;  
  box-sizing: border-box;  
}
```

```
<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

Fonte: [Interneting is Hard](#)



Posicionamento

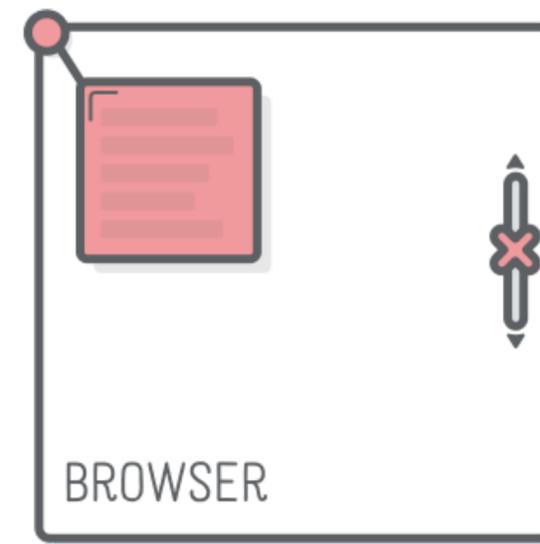
Fixed

```
.parent {  
  color: blue;  
  background-color: gray;  
  width: 500px;  
  height: 500px;  
}
```

```
.special.box {  
  position: fixed;  
  top: 20px;  
  left: 20px;  
  background-color: cyan;  
}
```

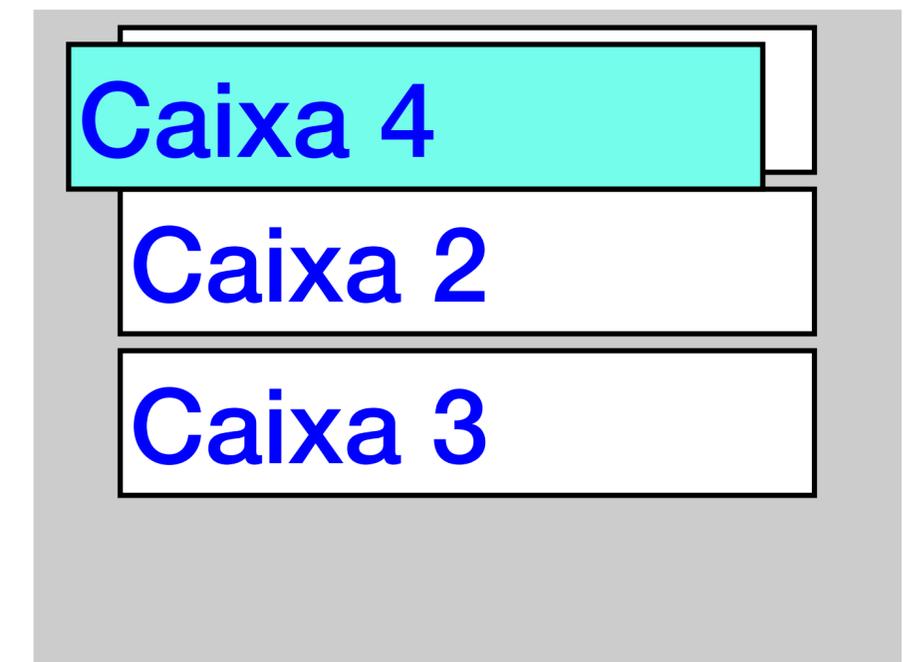
```
.box {  
  font-size: 2em;  
  background-color: white;  
  width: 400px;  
  border: 1px solid black;  
  margin: 10px auto;  
  box-sizing: border-box;  
}
```

```
<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>
```



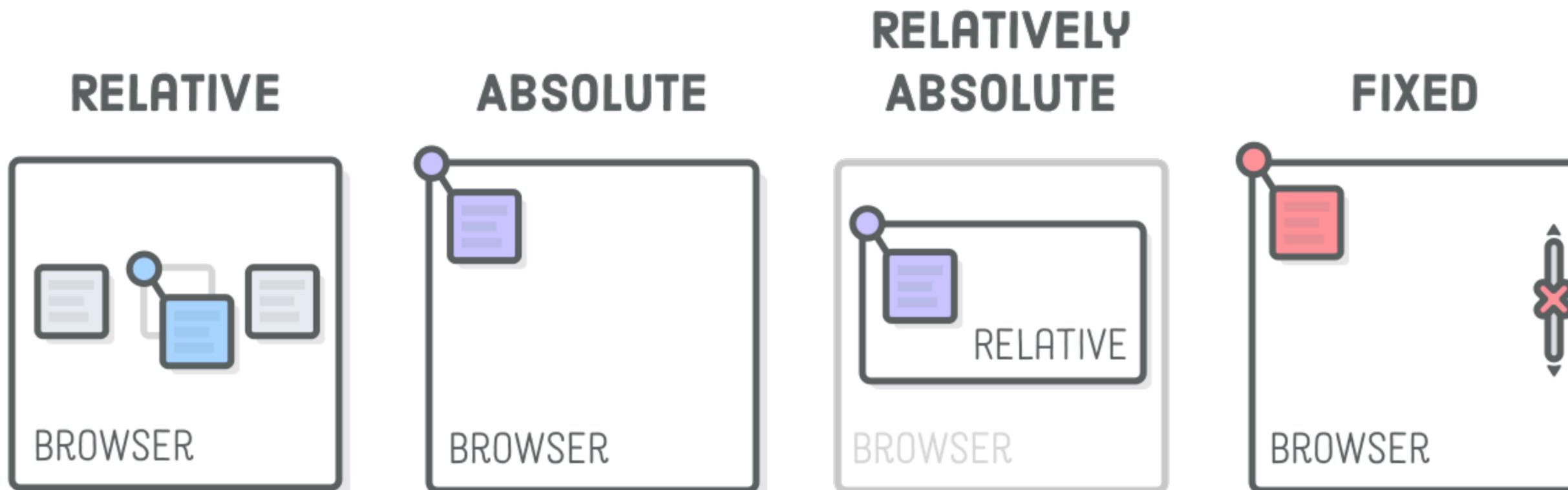
FIXED POSITIONING

Fonte: [Interneting is Hard](#)



Posicionamento

Todas as estratégias

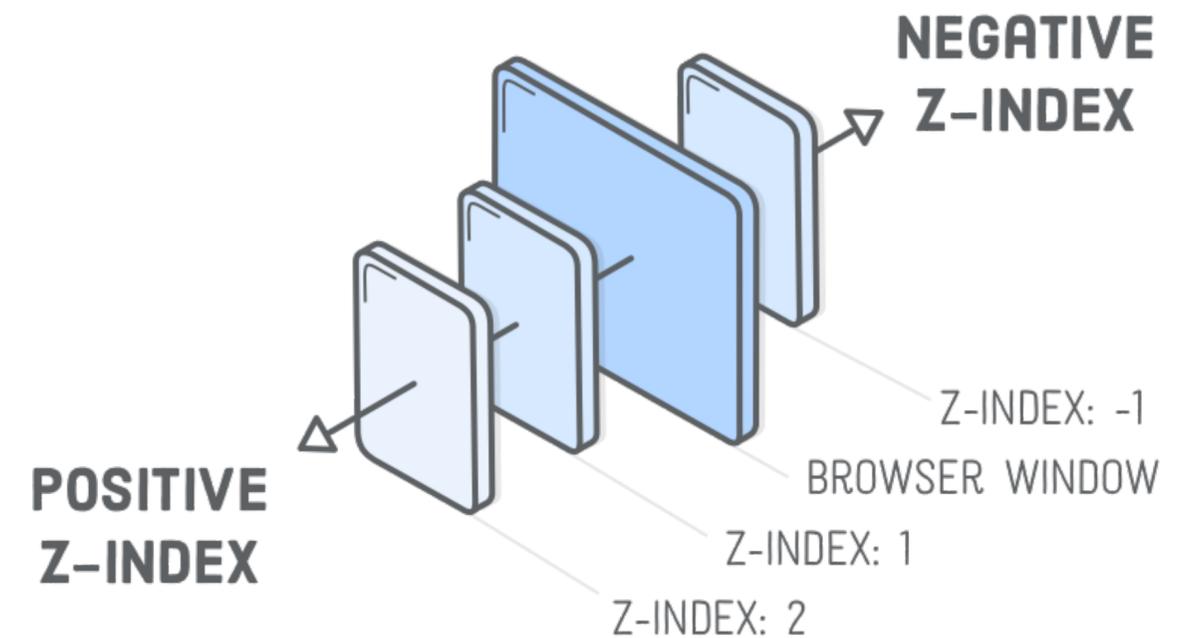


Estratégia de posicionamento: [Interneting is Hard](#)

Posicionamento

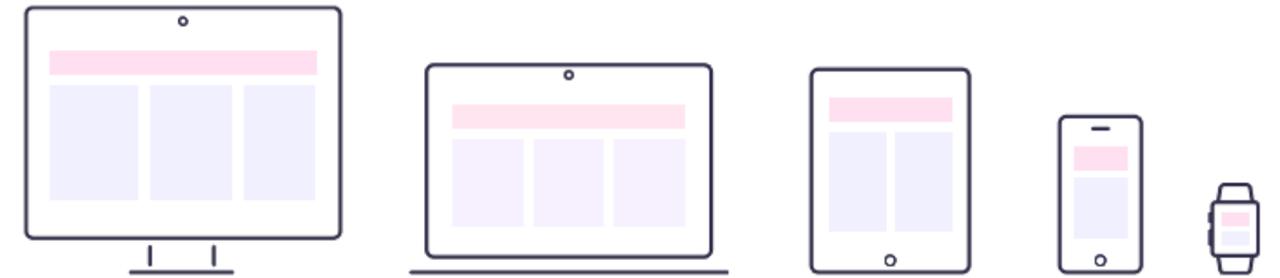
Z-index

- Elementos que usam *relative*, *absolute* e *fixed* são também chamados *positioned elements*
- Único afetados pela propriedade **z-index**
- O posicionamento *absolute* e *fixed* não obedecem o **fluxo normal** de posicionamento



Fonte: [Interneting is Hard](#)

Responsividade



Responsividade

Introdução

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

resoluções

Como ?

- Media queries
- Layout fluído
- Imagens flexíveis

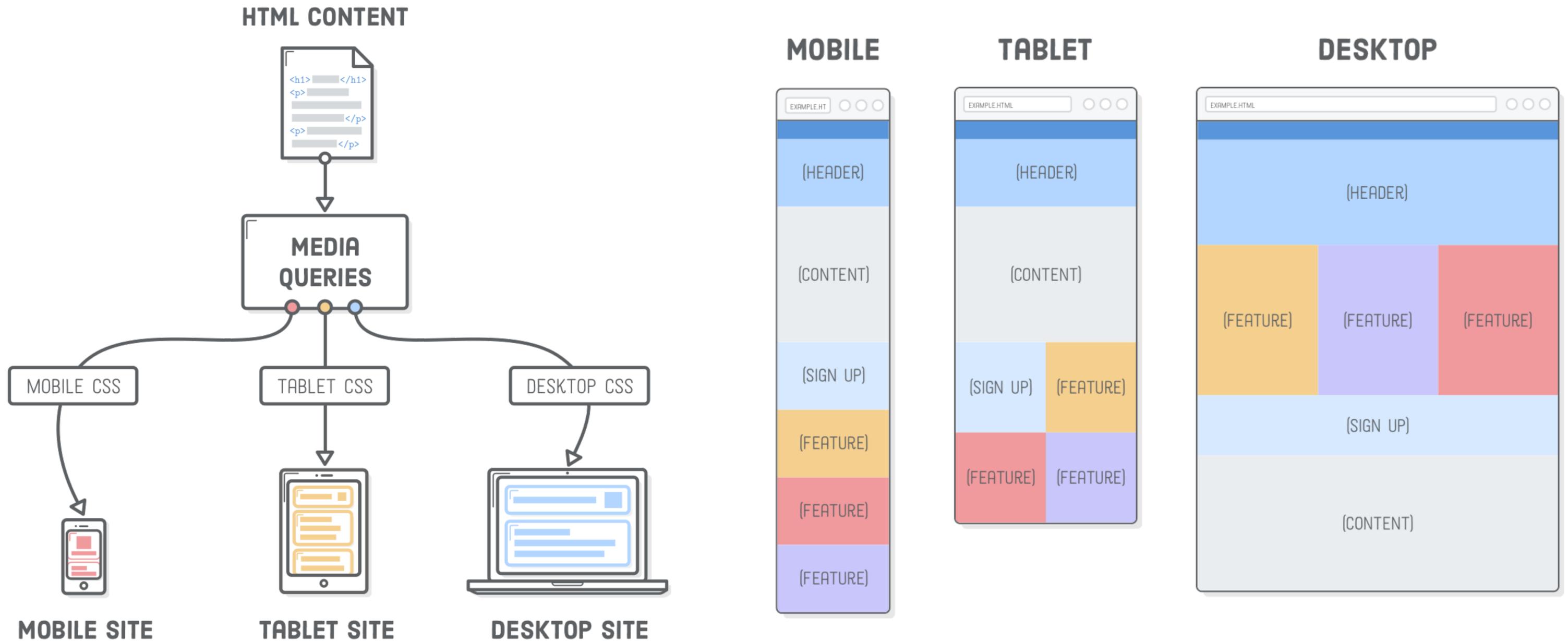
Responsividade

Media query

- Neste contexto o CSS tem papel fundamental por meio das *media queries*
 - Mecanismo que permite aplicar regras mediante uma condição
 - Permite mostrar o mesmo conteúdo HTML com aparência distinta

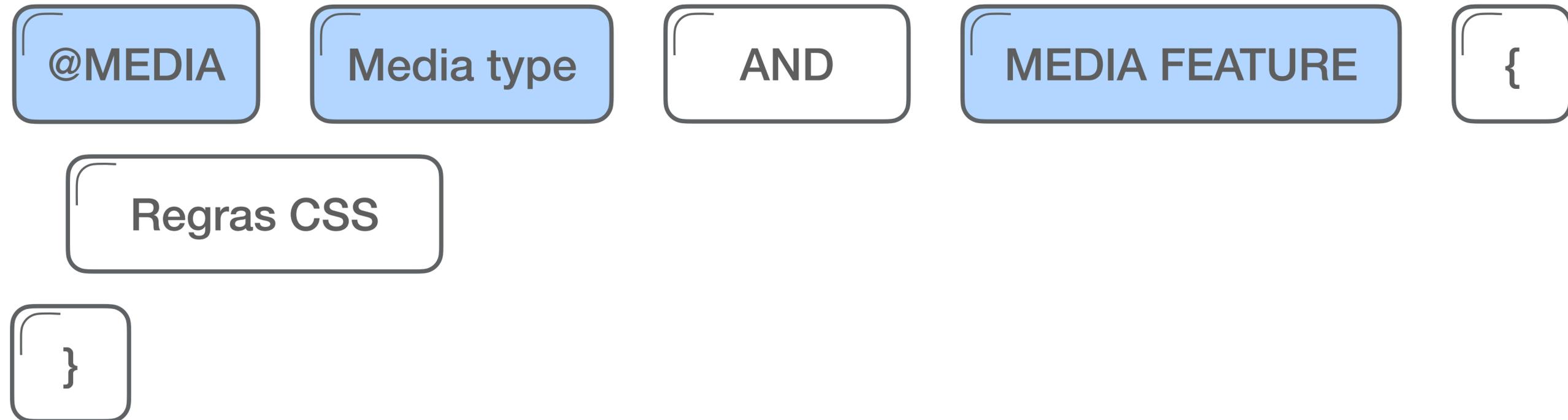
Responsividade

Media query



Responsividade

Media query



Responsividade

```
@media only screen and (max-width: 400px) {  
  body {  
    background-color: #F09A9D;  
  }  
}
```

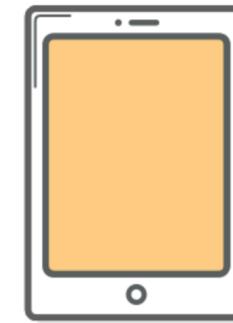
```
@media only screen and (min-width: 401px)  
and (max-width: 960px) {  
  body {  
    background-color: #F5CF8E;  
  }  
}
```

```
@media only screen and (min-width: 961px) {  
  body {  
    background-color: #B2D6FF;  
  }  
}
```

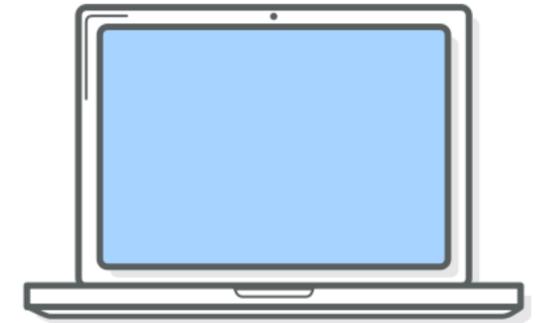
```
<body>  
</body>
```



MOBILE



TABLET

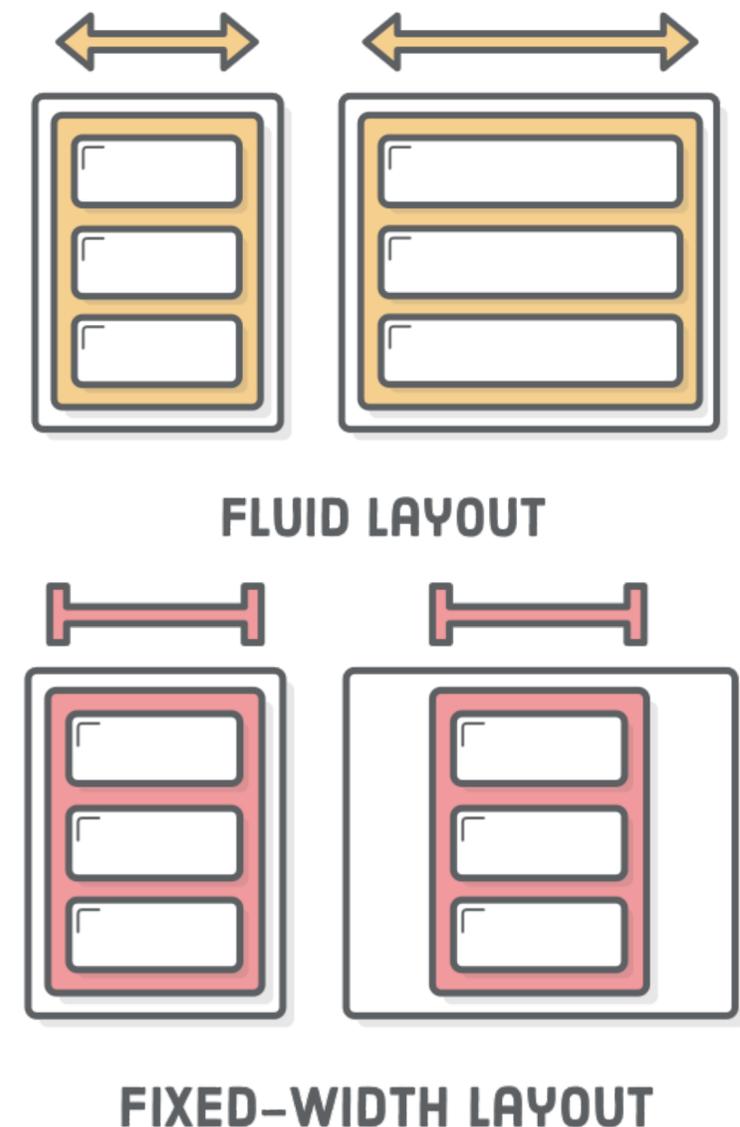


DESKTOP

Responsividade

Tipos de layout

- Existem dois tipos de layouts que devemos conhecer
 - Fluído
 - Estica e encolhe de acordo com a largura da tela
 - Largura-fixa
 - Possui a mesma largura independente das dimensões da tela
- Em geral usamos fluídos para celulares e tablets e largura fixa para telas maiores



Responsividade

Mobile-First

- Abordagem simples e evolutiva
- Maximiza o reuso do código CSS
- Começar o desenvolvimento pelo mobile e, depois, adicionar suporte a layouts desktop
 - O layout de desktop tende a ser mais complexo
- Uso de media queries min-width ao invés de max-width, mais comum em códigos desktop-first

Responsividade

```
.page {  
  display: flex;  
  flex-wrap: wrap;  
}
```

```
.section {  
  width: 100%;  
  height: 300px;  
}
```

```
.menu {  
  background-color: #5995DA;  
  height: 80px;  
}
```

```
.header {  
  background-color: #B2D6FF;  
}
```

```
.content {  
  background-color: #EAEDF0;  
  height: 600px;  
}
```

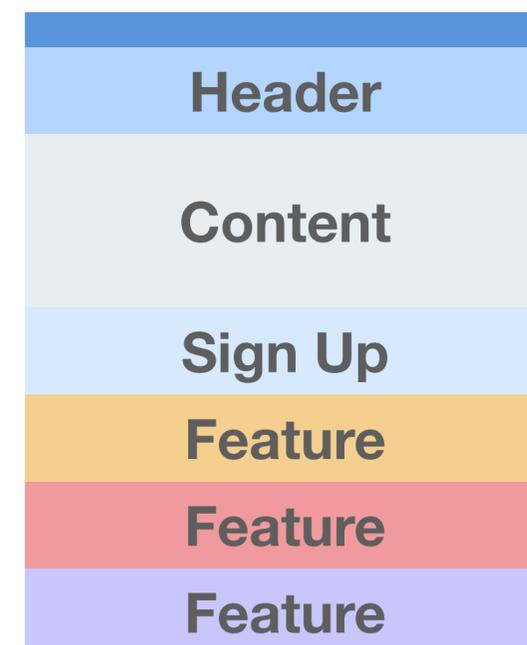
```
.sign-up {  
  background-color: #D6E9FE;  
}
```

```
.feature-1 {  
  background-color: #F5CF8E;  
}
```

```
.feature-2 {  
  background-color: #F09A9D;  
}
```

```
.feature-3 {  
  background-color: #C8C6FA;  
}
```

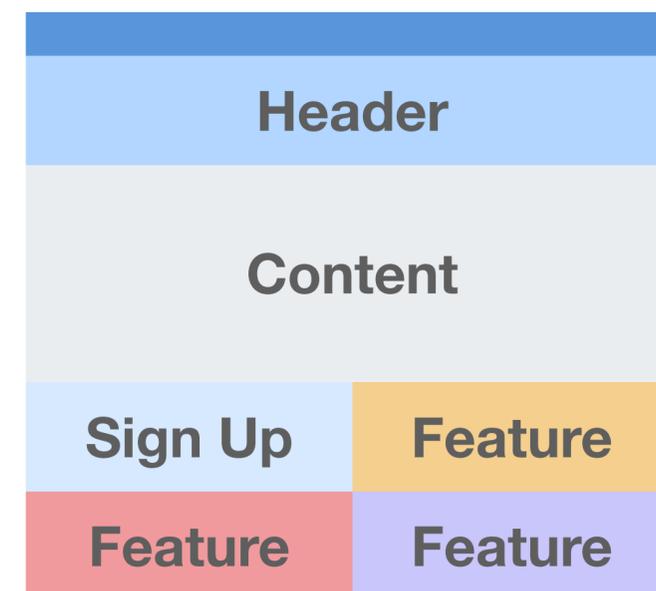
```
<div class='page'>  
  <div class='section menu'></div>  
  <div class='section header'></div>  
  <div class='section content'></div>  
  <div class='section sign-up'></div>  
  <div class='section feature-1'></div>  
  <div class='section feature-2'></div>  
  <div class='section feature-3'></div>  
</div>
```



Responsividade

```
@media only screen and (min-width: 401px)
and (max-width: 960px) {
  .sign-up,
  .feature-1,
  .feature-2,
  .feature-3 {
    width: 50%;
  }
}
```

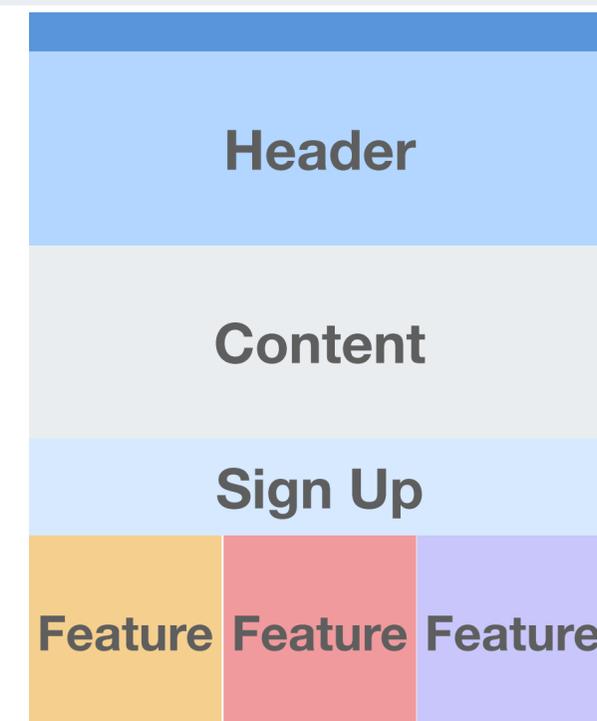
```
<div class='page'>
  <div class='section menu'></div>
  <div class='section header'></div>
  <div class='section content'></div>
  <div class='section sign-up'></div>
  <div class='section feature-1'></div>
  <div class='section feature-2'></div>
  <div class='section feature-3'></div>
</div>
```



Responsividade

```
@media only screen and (min-width: 961px) {  
  .page {  
    width: 960px;  
    margin: 0 auto;  
  }  
  .feature-1,  
  .feature-2,  
  .feature-3 {  
    width: 33.3%;  
  }  
  .header {  
    height: 400px;  
  }  
}
```

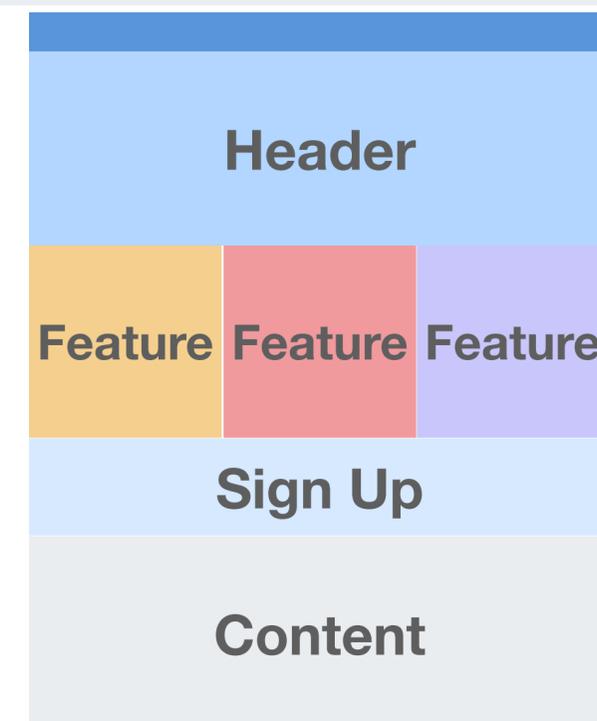
```
<div class='page'>  
  <div class='section menu'></div>  
  <div class='section header'></div>  
  <div class='section content'></div>  
  <div class='section sign-up'></div>  
  <div class='section feature-1'></div>  
  <div class='section feature-2'></div>  
  <div class='section feature-3'></div>  
</div>
```



Responsividade

```
@media only screen and (min-width: 961px) {  
  .sign-up {  
    height: 200px;  
    order: 1;  
  }  
  .content {  
    order: 2;  
  }  
}
```

```
<div class='page'>  
  <div class='section menu'></div>  
  <div class='section header'></div>  
  <div class='section content'></div>  
  <div class='section sign-up'></div>  
  <div class='section feature-1'></div>  
  <div class='section feature-2'></div>  
  <div class='section feature-3'></div>  
</div>
```



Modelos de conteúdo

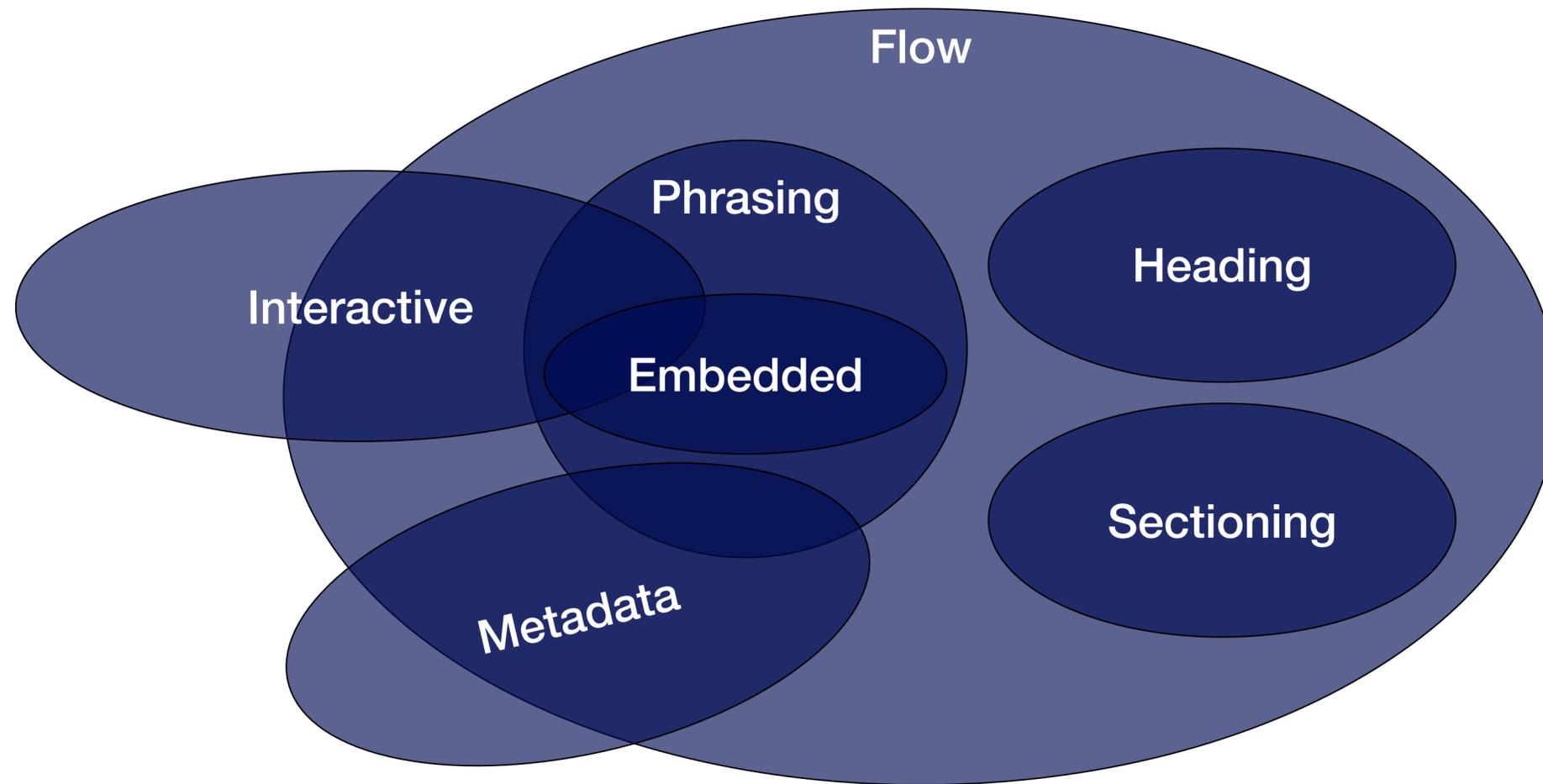
Modelos de conteúdos

Introdução

- Os elementos do HTML5 podem ser agrupados em 15 categorias que compartilham características semelhantes
 - Cada elemento pertencem a zero ou mais categorias
- Vamos tratar de elementos das seguintes categorias:
 - Flow content, Sectioning content, Heading content, Phrasing content, Embedded content, Interactive content e Metadata content
- Para mais informações sobre as 15 categorias acesse: [HTML Living standard: Element content categories](#)

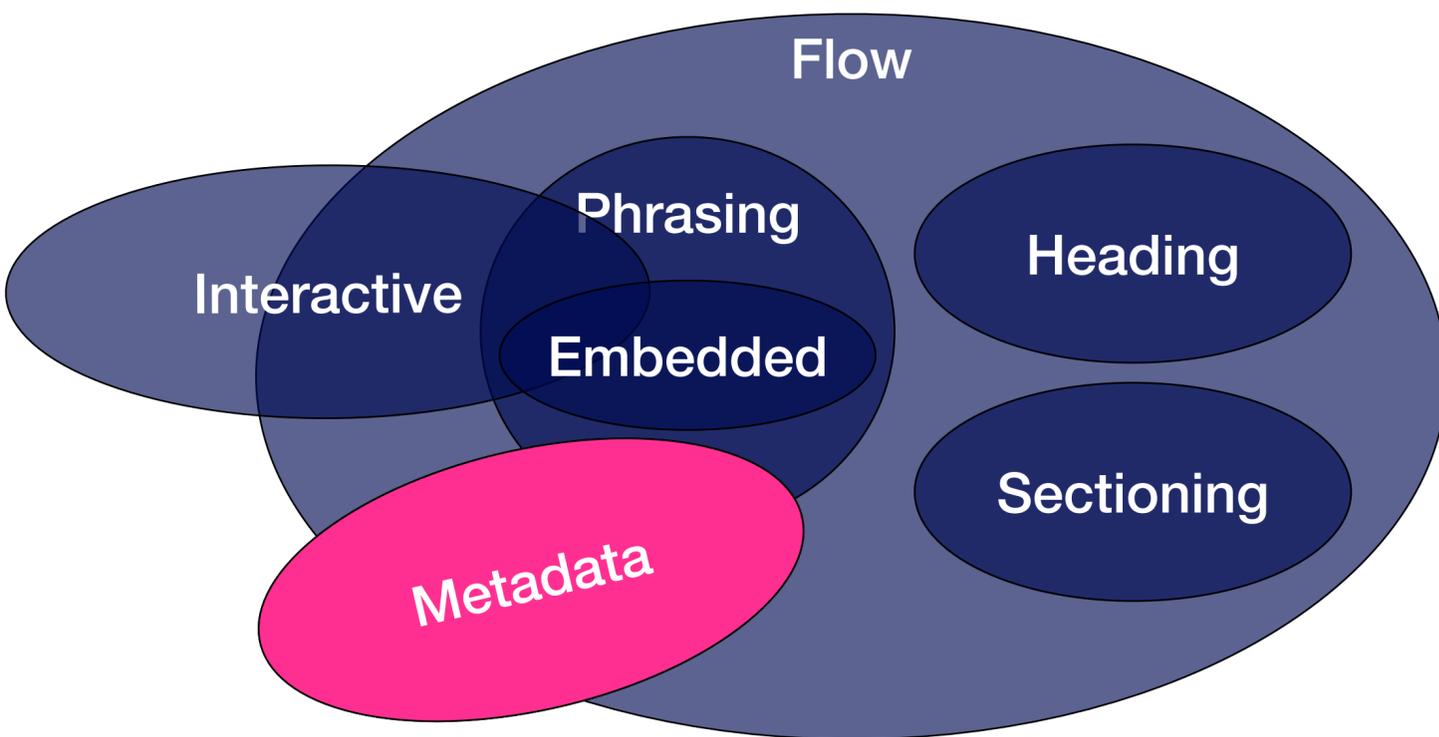
Modelos de conteúdos

Categorias



Modelos de conteúdos

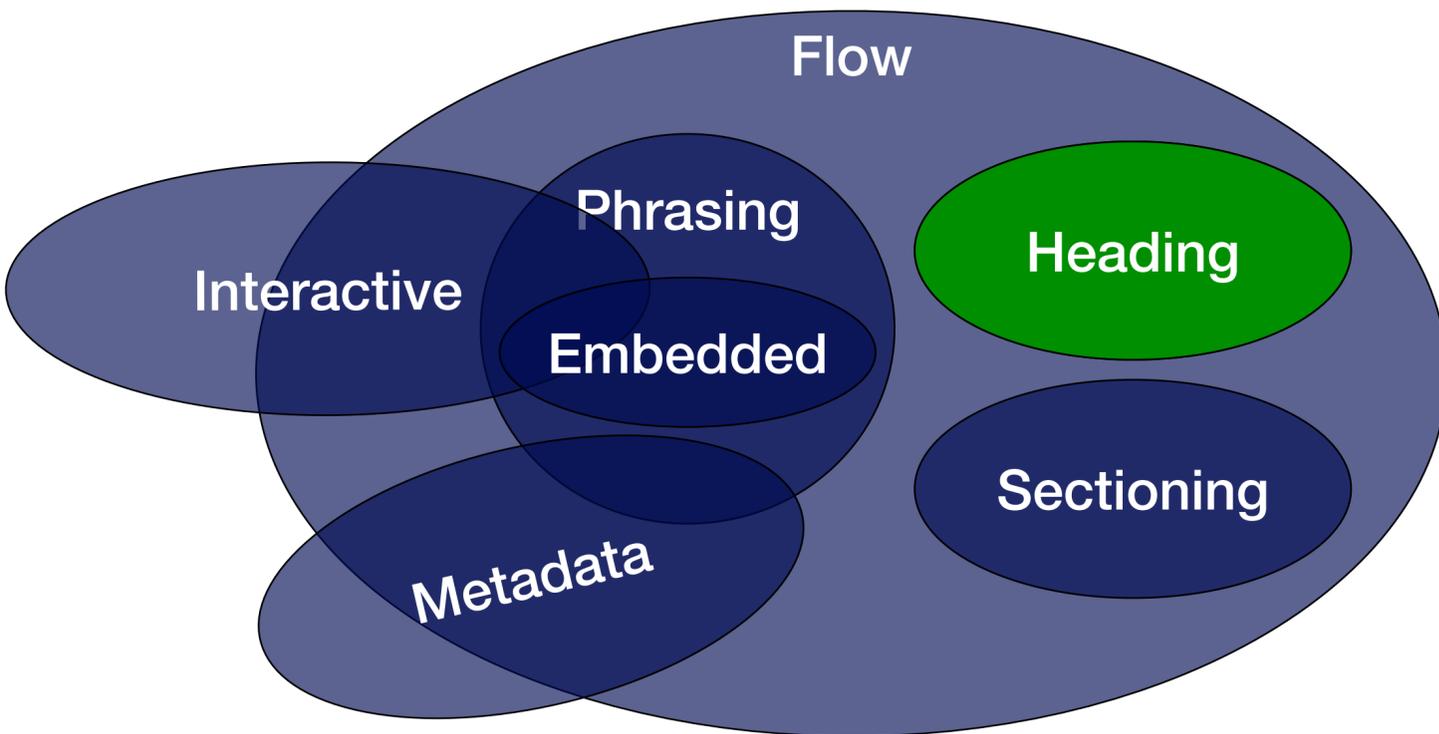
Metadata content



- Contém elementos que definem:
 - A maneira que o documento será exibido
 - O comportamento do restante do conteúdo
 - O relacionamento com outros documentos
- Elementos:
 - `<base>`, `<command>`, `<link>`, `<meta>`, `<noscript>`, `<script>`, `<style>` e `<title>`

Modelos de conteúdos

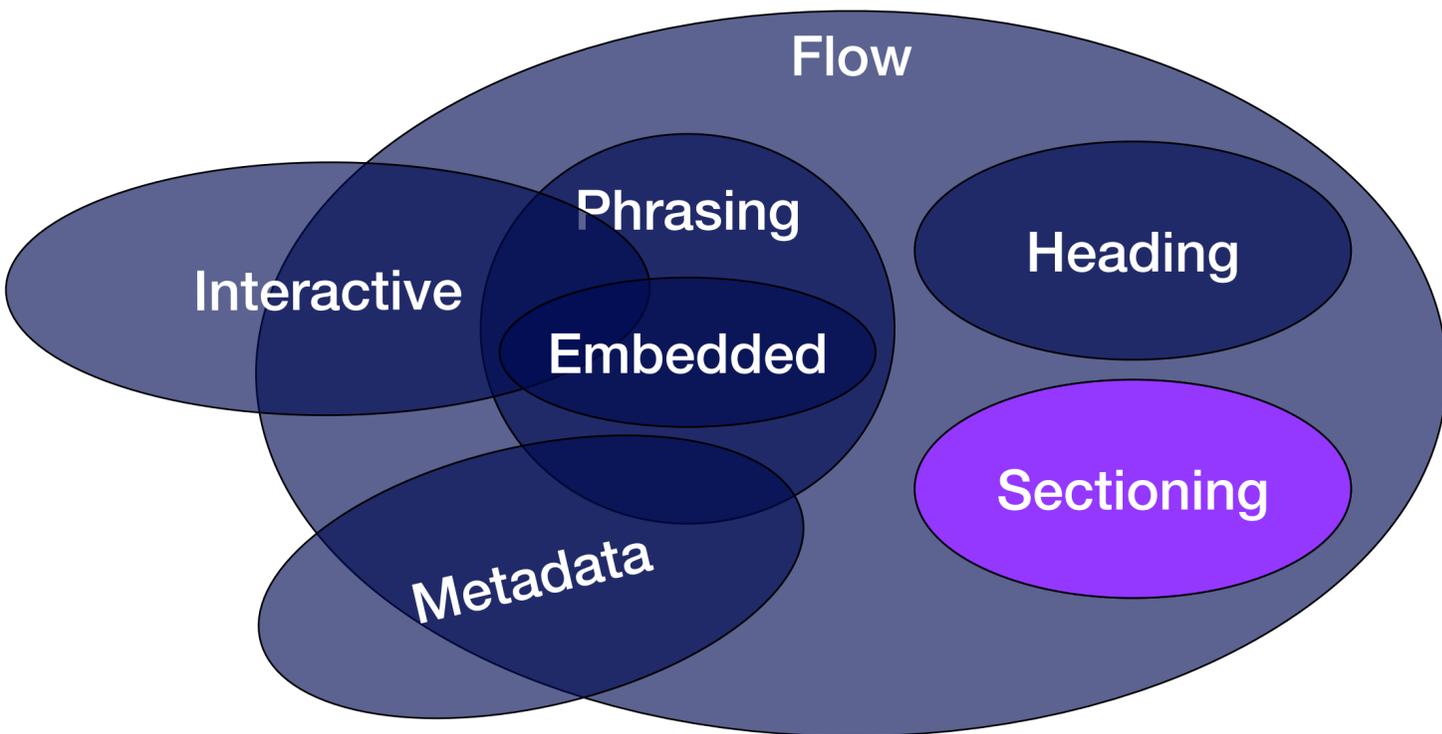
Heading content



- Definem cabeçalhos individuais ou agrupados
- Podendo definir cabeçalhos de conteúdo geral ou de seções (Sectioning content)
- Elementos:
 - `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`, `<hgroup>`

Modelos de conteúdos

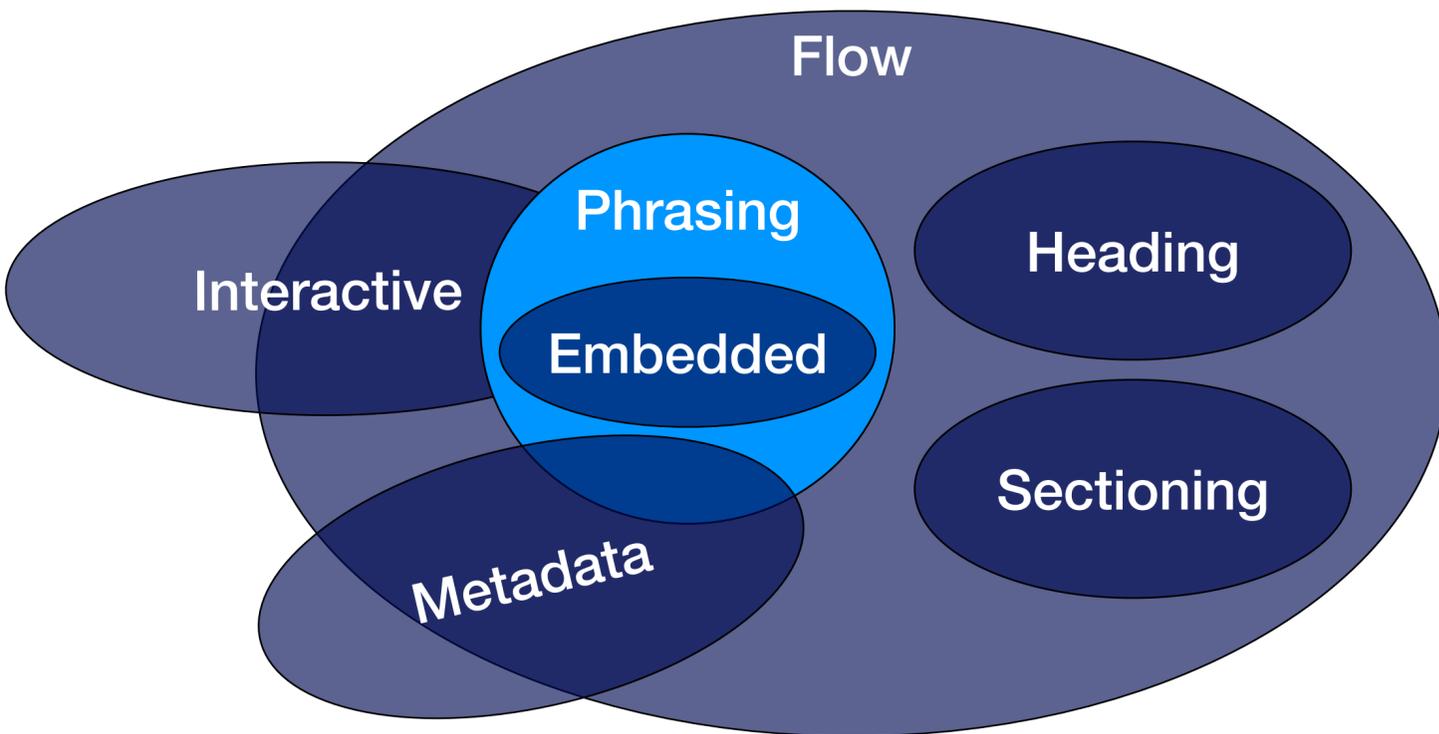
Sectioning content



- Definem seções lógicas (temáticas) no documento
 - Outros elementos, heading content, podem ser usados para estabelecer uma estrutura mais semântica
- Elementos:
 - `<article>`, `<aside>`, `<nav>`, `<section>`

Modelos de conteúdos

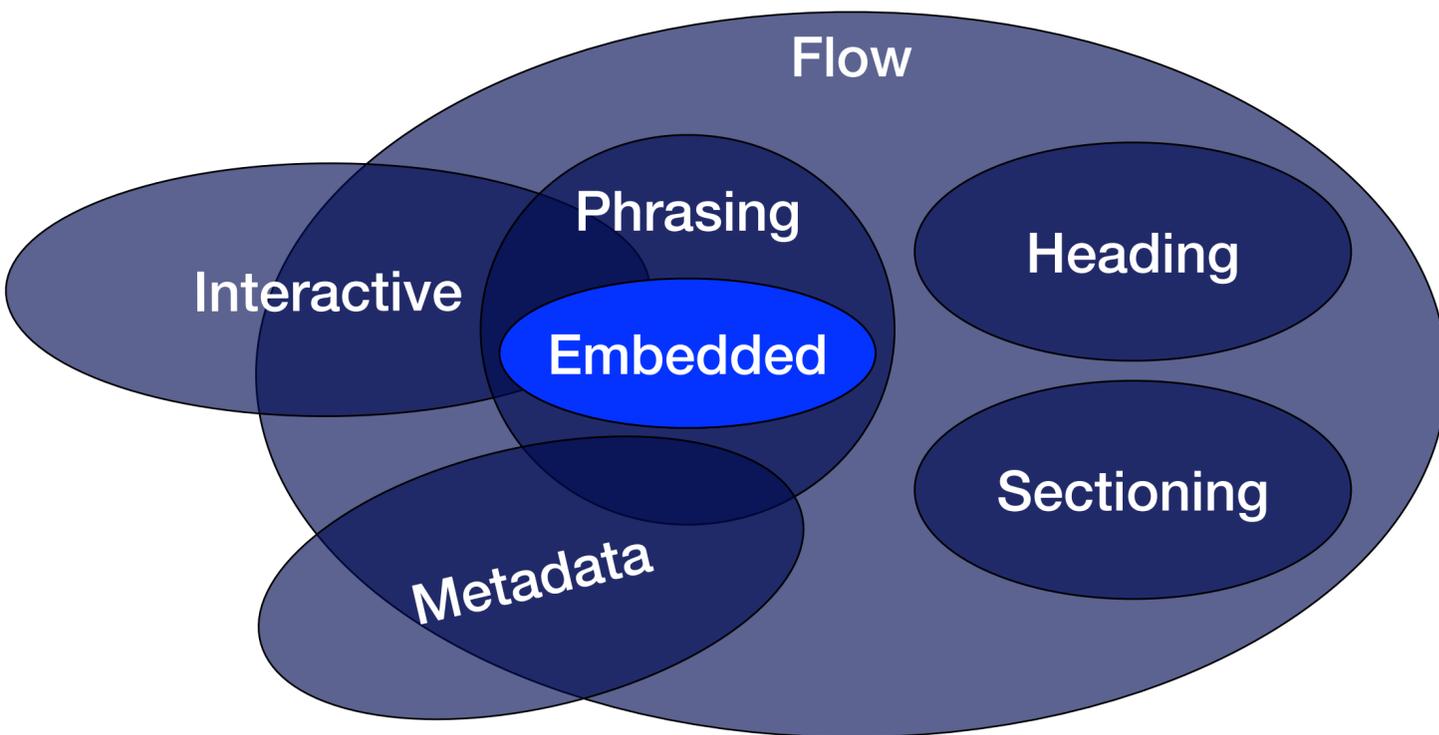
Phrasing



- Elementos que integram o texto do documentos
 - Ex: Textos inserido em parágrafos
- Podem conter somente elementos do seu próprio tipo
 - Nunca devem incluir elementos do tipo Flow content
- Elementos:
 - `<abbr>`, `<audio>`, ``, `<bdo>`, `
`, `<button>`, `<canvas>`, `<cite>`, `<code>`, `<command>`, `<datalist>`, `<dfn>`, ``, `<embed>`, `<i>`, `<iframe>`, ``, `<input>`, `<kbd>`, `<keygen>`, `<label>`, `<mark>`, `<math>`, `<meter>`, `<noscript>`, `<object>`, `<output>`, `<progress>`, `<q>`, `<ruby>`, `<samp>`, `<script>`, `<select>`, `<small>`, ``, ``, `<sub>`, `<sup>`, `<svg>`, `<textarea>`, `<time>`, `<var>`, `<video>`, `<wbr>`

Modelos de conteúdos

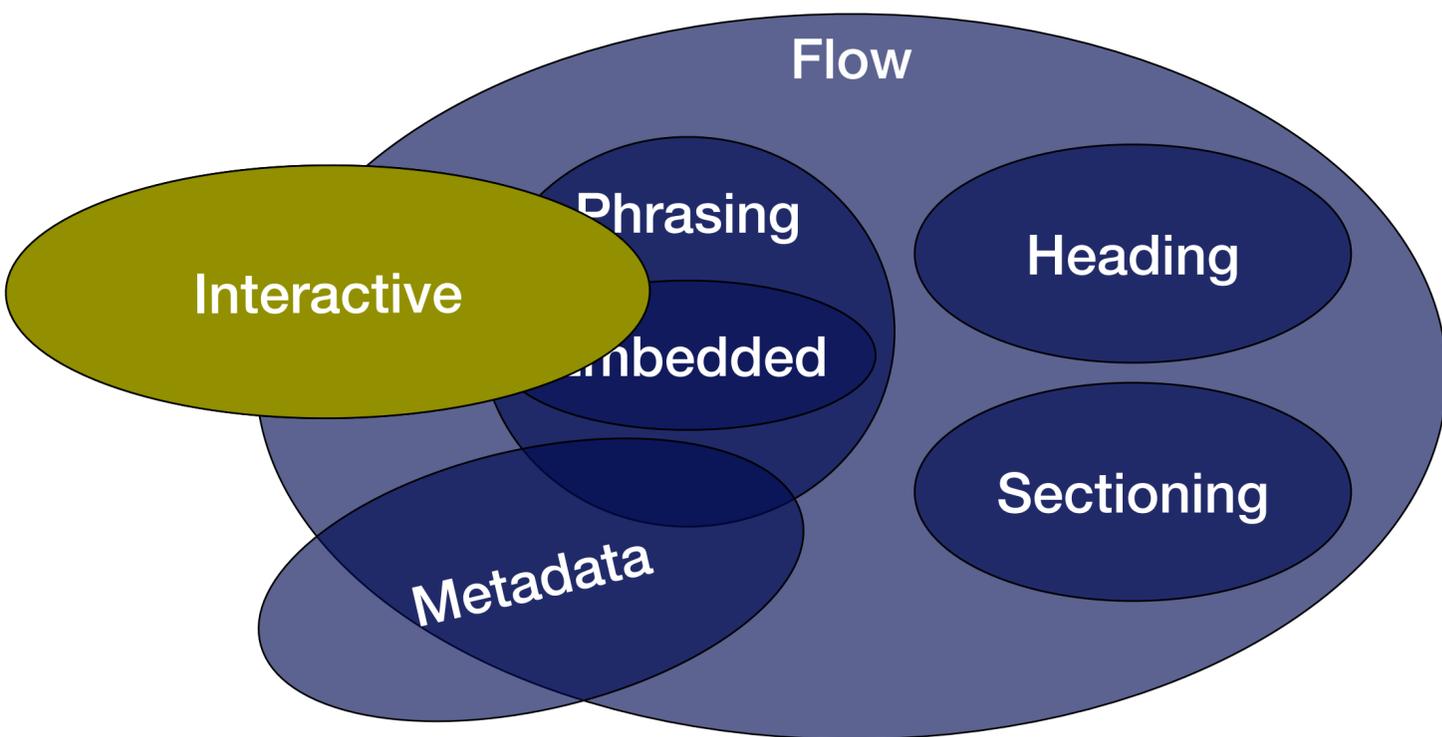
Embedded



- Elementos que importam outro recurso ou inserem conteúdo de uma outra linguagem de marcação
- Elementos:
 - `<audio>`, `<canvas>`, `<embed>`, `<iframe>`, ``, `<math>`, `<object>`, `<svg>` e `<video>`

Modelos de conteúdos

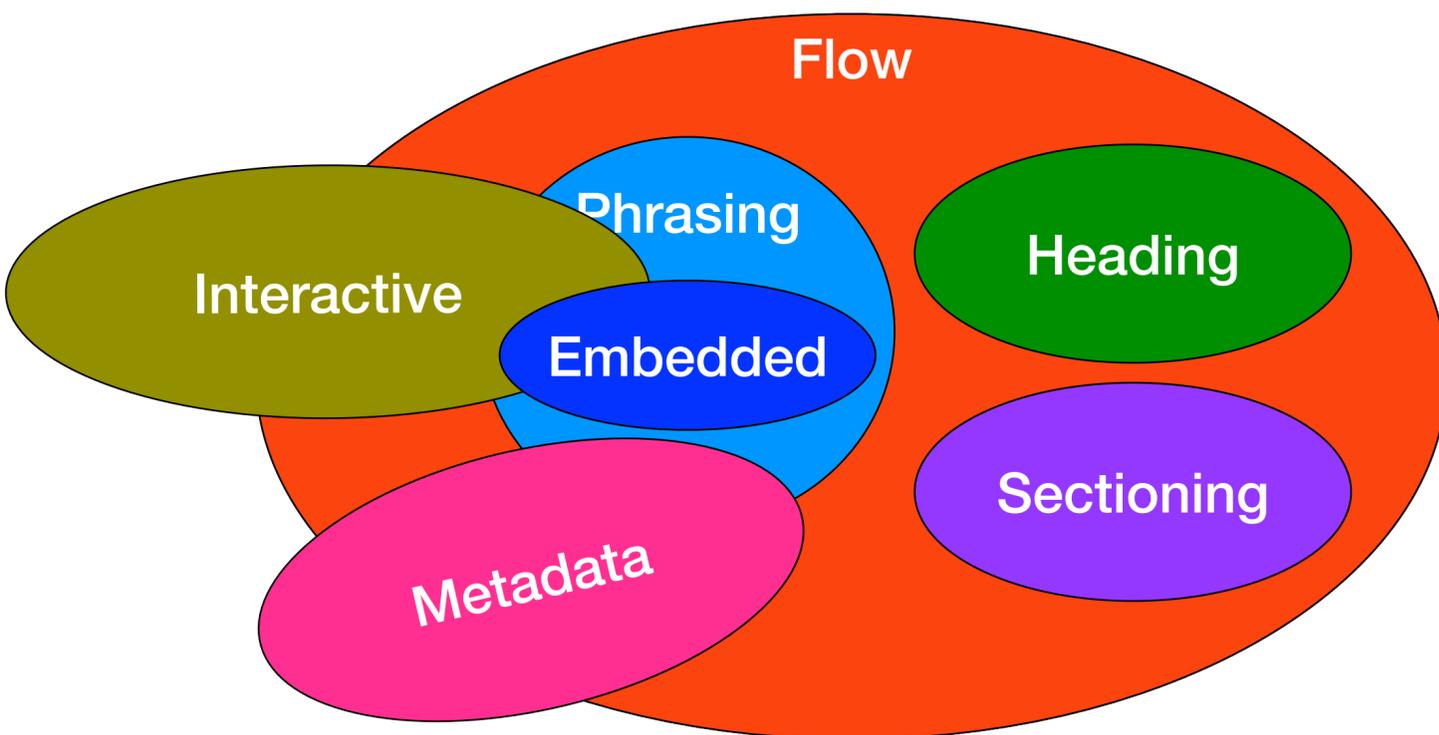
Categorias



- Elementos destinados à interação do usuário
- Elementos:
 - `<a>`, `<button>`, `<details>`, `<embed>`, `<iframe>`, `<keygen>`, `<label>`, `<select>`, e `<textarea>`

Modelos de conteúdos

Categorias



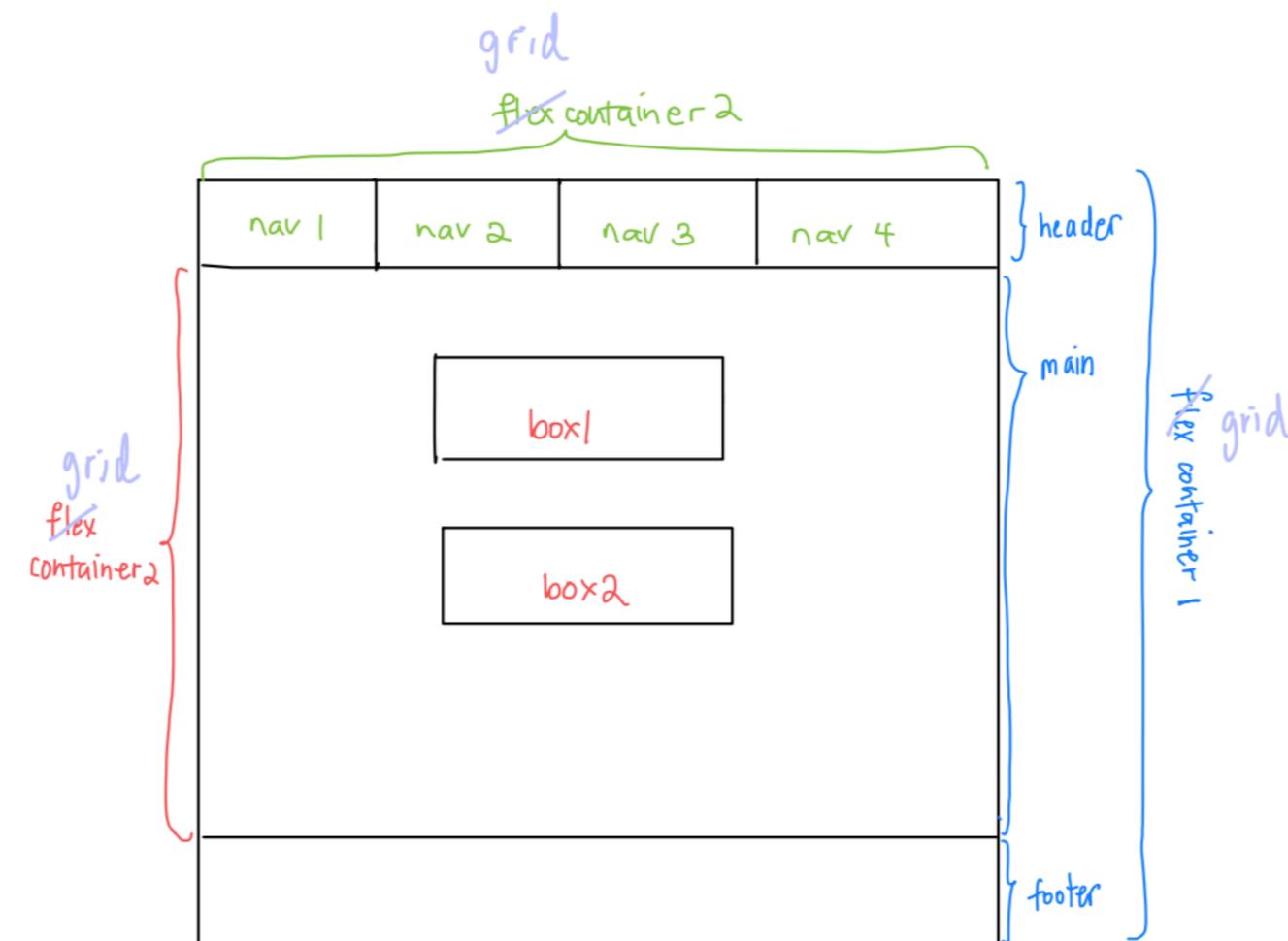
- A grande maioria dos elementos inseridos no corpo (<body>) do documento

- Alguns elementos:

- <a>, <abbr>, <address>, <article>, <aside>, <audio>, , <bdo>, <bdi>, <blockquote>,
, <button>, <canvas>, <cite>, <code>, <command>, <data>, <datalist>, , <details>, <dfn>, <div>, <dl>, , <embed>, <fieldset>, <figure>, <footer>, <form>, <h1>, <h2>, <h3>, <h4>, <h5>, <h6>, <header>, <hgroup>, <hr>, <i>, <iframe>, , <input>, <ins>, <kbd>, <keygen>, <label>, <main>, <map>, <mark>, <math>, <menu>, <meter>, <nav>, <noscript>, <object>, , <output>, <p>, <pre>, <progress>, <q>, <ruby>, <s>, <samp>, <script>, <section>, <select>, <small>, , , <sub>, <sup>, <svg>, <table>, <template>, <textarea>, <time>, , <var>, <video>, <wbr>

[*Clique aqui para ver a lista completa](#)

Layout com HTML Semântico



Layout com HTML Semântico

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**
 - Cada **div** deveria ser identificada com um nome significativo
- O HTML 5 introduziu uma série de elementos mais semânticos
 - Tags semânticas para texto
 - **Tags semânticas para estrutura**
 - *Section elements*

Layout com HTML Semântico

Construindo layouts

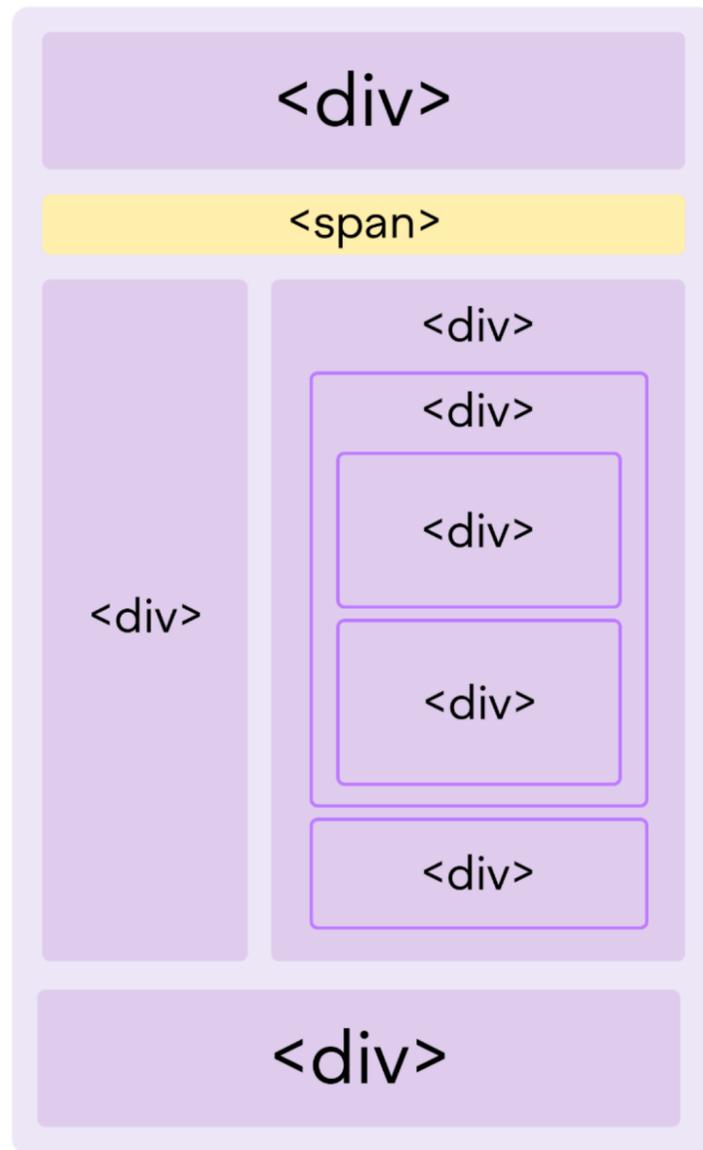
```
<body>
  <div id="header">
  </div>
  <div id="nav">
  </div>
  <div id="section">
  </div>
  <div id="footer">
  </div>
</body>
```

HTML4 → HTML5

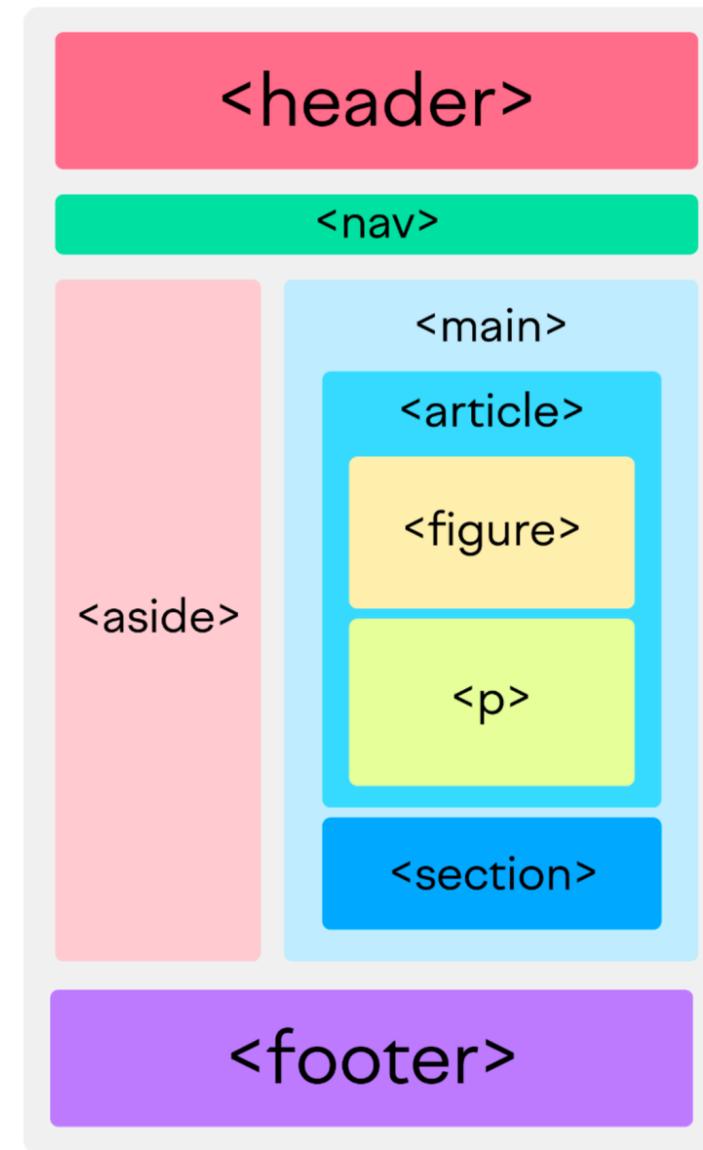
```
<body>
  <header>
  </header>
  <nav>
  </nav>
  <section>
  </section>
  <footer>
  </footer>
</body>
```

Layout com HTML Semântico

Construindo layouts



HTML4 → HTML5



Layout com HTML Semântico

HTML Semântico

- Refere-se ao uso de tags HTML que transmitem o significado (**semântica**) do conteúdo contido nelas
 - Também conhecido como marcação semântica
- Fornece informações adicionais que ajudam a definir as funções e a importância relativa das diferentes partes da sua página
 - Gerando uma página mais acessível
 - Motores de busca e leitores de tela passam a entender melhor a sua página
- **É uma boa prática da programação web**

Layout com HTML Semântico

O esboço do documento (*Outline*)

- Todo documento HTML tem um “esboço” (*outline*)
 - É como os mecanismos de pesquisa e os leitores de tela visualizam a hierarquia do conteúdo da página
 - Todos os elementos de cabeçalho `<h1...h6>` contribuem para o esboço do documento de uma página.

Layout com HTML Semântico

O esboço do documento (*Outline*)

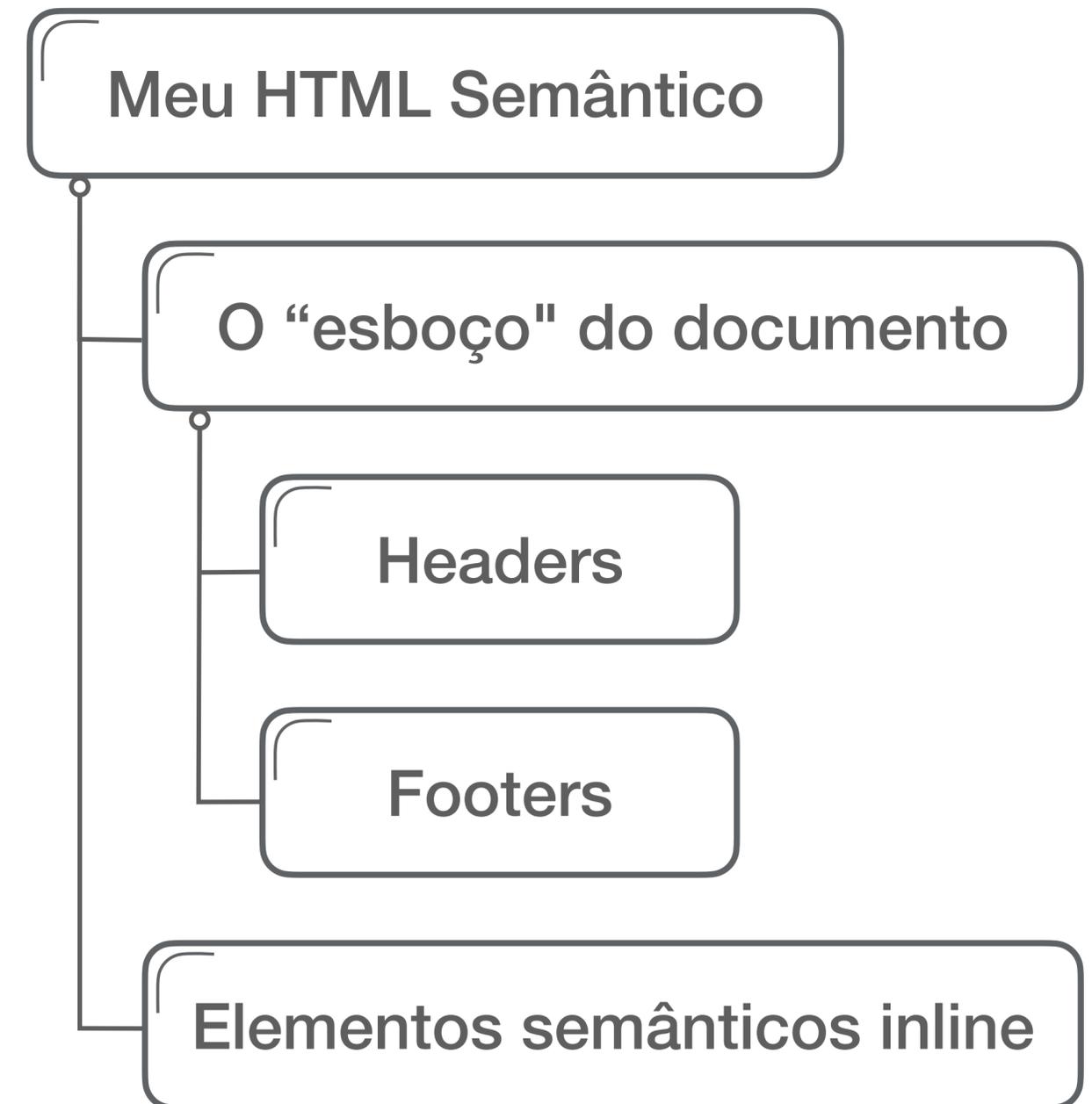
```
<h1>Meu HTML Semântico</h1>
<p>Por Bruno Góis Mateus. Publicado em 2023</p>
<p>Um belo exemplo de HTML semântico</p>

<h2>O "esboço" do documento</h2>
<p>O HTML5 inclui diversos elementos (sectioning content) que afetam o outline do documento.</p>

<h3>Headers</h3>
<p>O <code>&lt;header&gt;</code> é um desses elementos.</p>

<h3>Footers</h3>
<p>O <code>&lt;footer&gt;</code> também.</p>

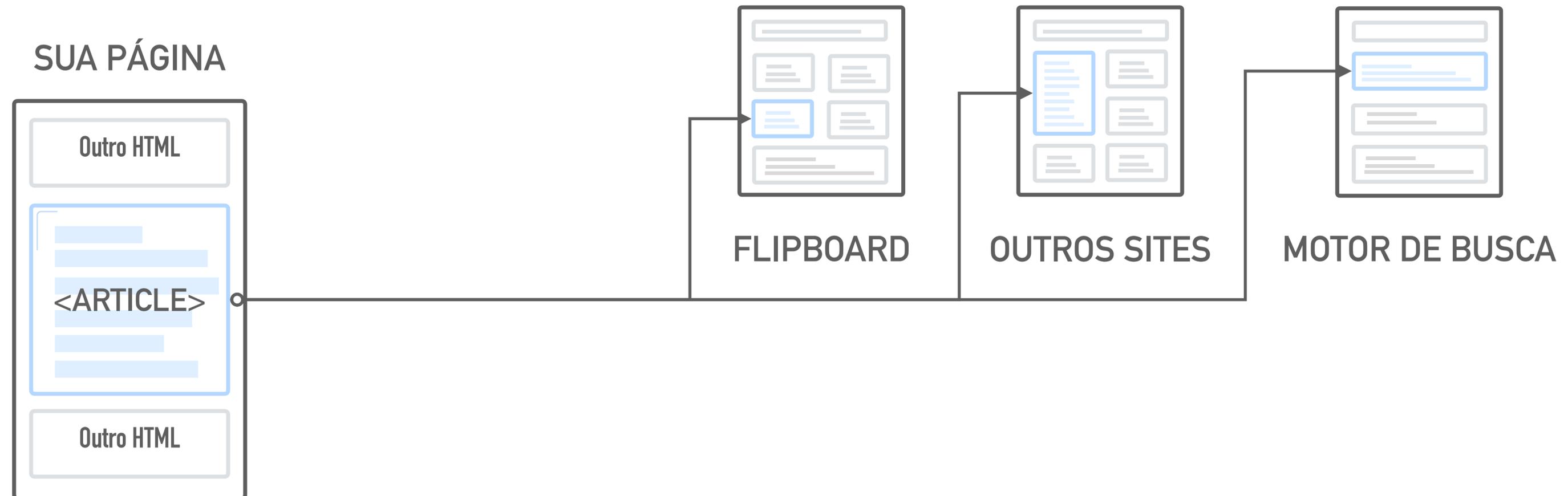
<h2>Elementos semânticos inline</h2>
<p>O <code>&lt;time&gt;</code> também é semântico, mas não é um sectioning content.</p>
<p>&copy; 2023 Bruno Góis Mateus</p>
```



Layout com HTML Semântico

<article>

- Representa um artigo independente da página web
- Deve conter um conteúdo que continua relevante mesmo quando separado do contexto da página web de origem



Layout com HTML Semântico

<article>

```
<article>
  <h1>Minha primeira postagem</h1>
  <p>Por Bruno Góis Mateus. Fevereiro 2023</p>
  <p>Mussum Ipsum, cacilds vidis litro
abertis.</p>
</article>

<p>&copy; 2023 Bruno Góis Mateus</p>
```

Seção sem título

Minha primeira postagem

Layout com HTML Semântico

<article>

```
<article>
  <h1>Minha primeira postagem</h1>
  <p>Por Bruno Góis Mateus. Fevereiro 2023</p>
  <p>Mussum Ipsum, cacilds vidis litro abertis. </p>
</article>
```

```
<article>
  <h1>Minha segunda postagem</h1>
  <p>Por Bruno Góis Mateus. Março 2023</p>
  <p>Praesent vel viverra nisi.</p>
</article>
```

```
<article>
  <h1>Minha terceira postagem</h1>
  <p>Por Bruno Góis Mateus. Abril 2023</p>
  <p>Mauris aliquet nunc non turpis scelerisque, eget.</p>
</article>
```

```
<p>&copy; 2023 Bruno Góis Mateus</p>
```

Seção sem título

Minha primeira postagem

Minha segunda postagem

Minha terceira postagem

Layout com HTML Semântico

<section>

- Similar a um <article>
- Não precisa fazer sentido fora do contexto do documento
- Maneira explícita de definir uma seção no “esboço” (*outline*) do documento

SUA PÁGINA



Layout com HTML Semântico

<section>

```
<h1>Meu HTML Semântico</h1>
<p>Por Bruno Góis Mateus. Publicado em 2023</p>
<p>Um belo exemplo de HTML semântico</p>

<section>
  <h2>0 "esboço" do documento</h2>
  <p>0 HTML5 inclui diversos elementos (sectioning content)
  que afetam o outline do documento.</p>

  <h3>Headers</h3>
  <p>0 <code>&lt;header&gt;</code> é um desses elementos.</p>

  <h3>Footers</h3>
  <p>0 <code>&lt;footer&gt;</code> também.</p>
</section>

<section>
  <h6>&u>Elementos semânticos inline</h6>
  <p>0 <code>&lt;time&gt;</code> também é semântico, mas não é
  um sectioning content.</p>
  <p>&copy; 2023 Bruno Góis Mateus</p>
</section>
```

Meu HTML Semântico

O "esboço" do documento

Headers

Footers

Elementos semânticos inline

Layout com HTML Semântico

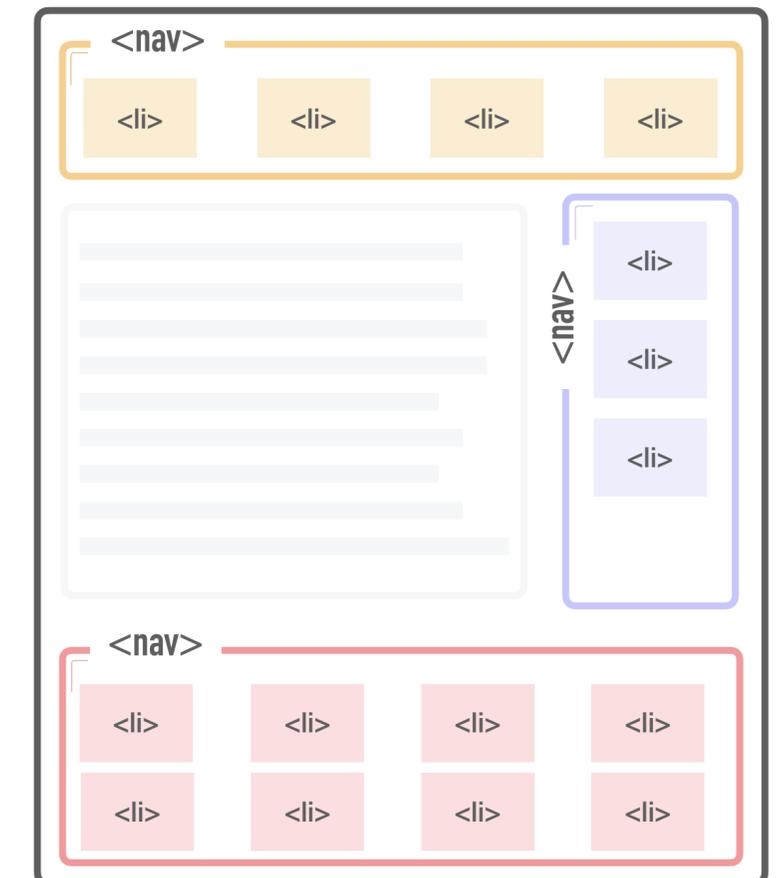
<section> - Boas práticas

- Evite a manipulação do *outline* usando `<section>`
 - Leitores de tela tem dificuldades essa tag por ser uma tag muito genérica
 - Utilize os diferentes headings `<h1...h6>`
 - Uma seção deve conter pelo menos um heading `<h1...h6>`
- Use `<section>` ao invés de `<div>`

Layout com HTML Semântico

`<nav>`

- Permite a marcação de área de navegação
 - Navegação principal, links para páginas relacionadas e etc ...
- É uma ótima fonte de informação para motores de busca
 - Facilita a descoberta de outras páginas



Layout com HTML Semântico

<header>

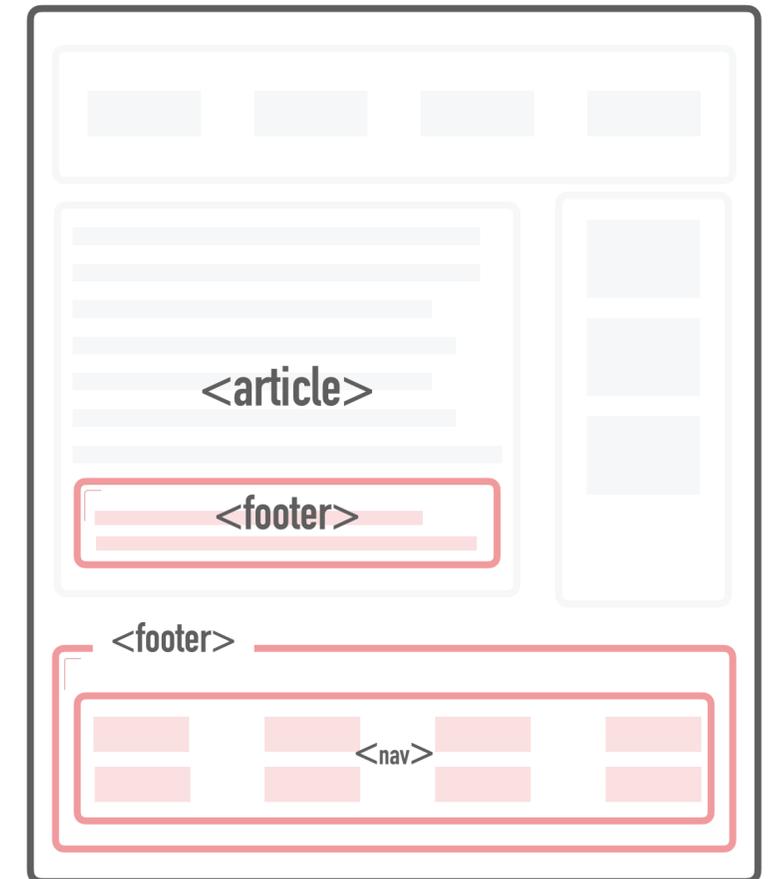
- Marca a introdução do conteúdo de uma seção, artigo ou da página
 - Pode conter qualquer coisa
 - Logo, links, informação sobre o autor ...
 - São associados aos elementos mais próximos
 - <body>, <section> ou <article>
- Não confundir com heading <h1...h6>



Layout com HTML Semântico

<footer>

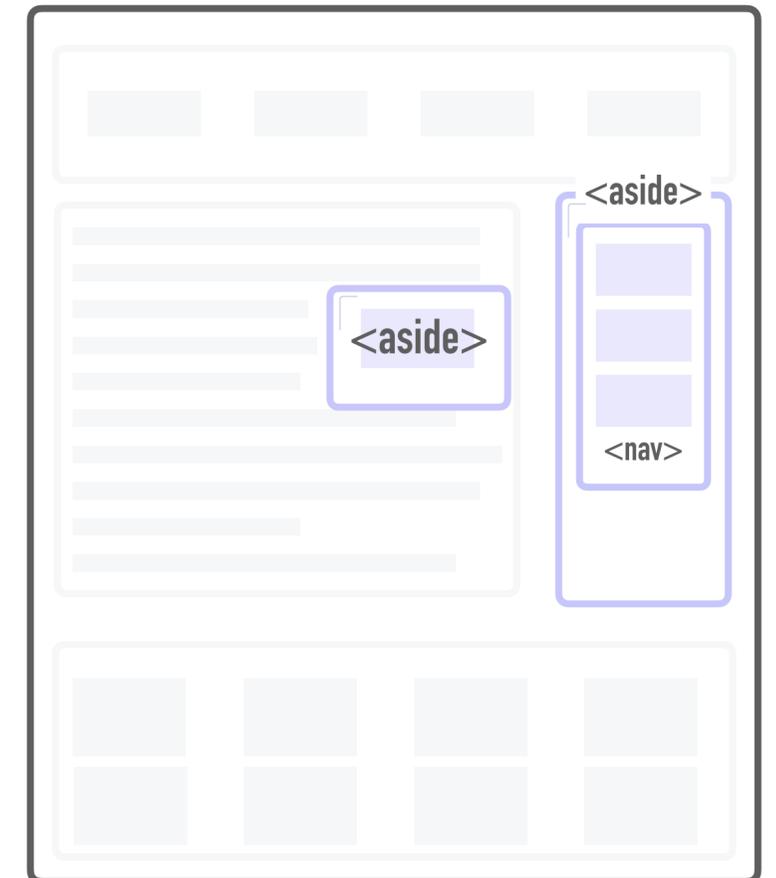
- Conceitualmente, são similares aos *headers*, com a diferença que eles aparecem no final dos elementos
- Caso comuns:
 - Copyright, navegação, informações sobre o autor
- Também são associados ao elemento HTML mais próximo



Layout com HTML Semântico

<aside>

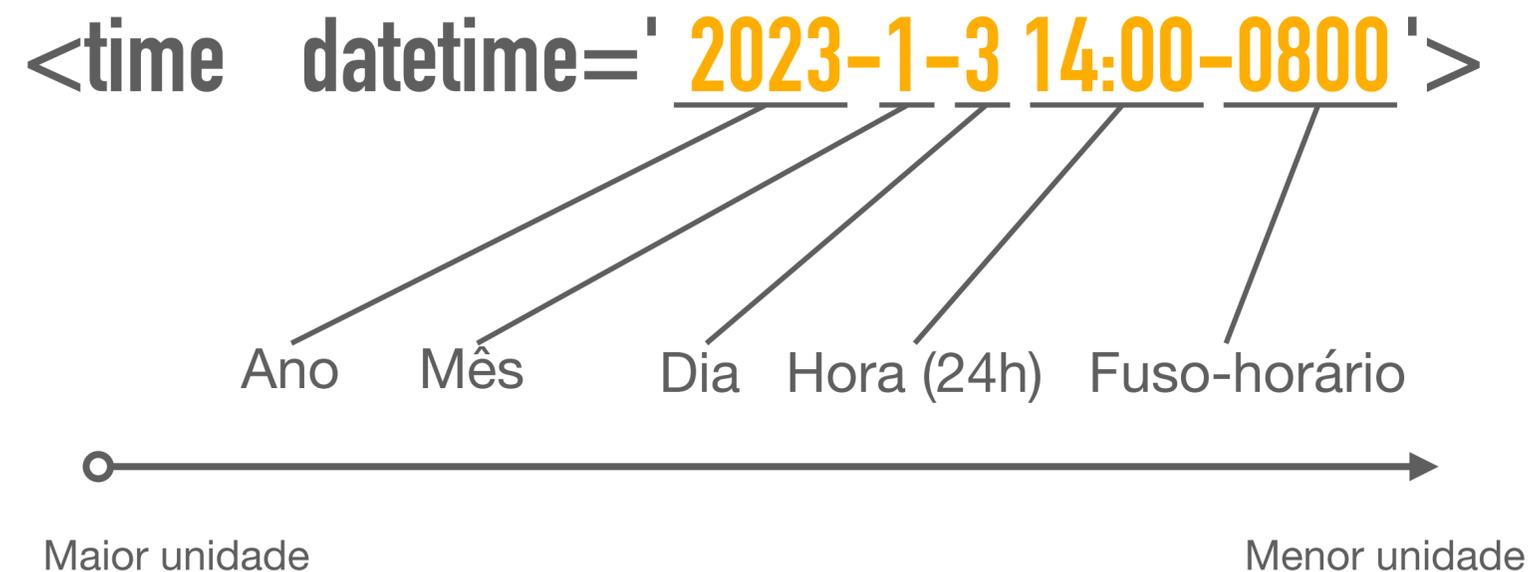
- Cabeçalhos e rodapés permitem a adição de informação extra nos artigos
- No entanto, algumas vezes precisamos *remover* conteúdo
 - Propaganda, anúncios
- Também são utilizados para destacar definições, estatísticas e citações
 - Se destoam do artigo existe a grande chance de ser um aside



Layout com HTML Semântico

<time>

- Representa um horário do dia ou um dia do calendário
 - Remove a ambiguidade da linguagem natural
 - Formato que máquina consegue ler



Layout com HTML Semântico

<address>

- Utilizado fornecer o contato do autor de um artigo ou da página
 - Em geral, tem grau de parentesco com <article> ou <body>
- Não deve ser utilizado para mostrar um endereço físico

```
<footer>  
  <p>Suco de cevadiss deixa as pessoas mais interessantes.Nec orci ornare consequat.</p>  
  <address>  
    Em caso de dúvidas, entrar em contato com <a href='mailto:brunomateus@exemplo.com'>Bruno</a>.  
  </address>  
</footer>
```

Layout com HTML Semântico

Exemplo completo

```
<article>
  <header>
    <h1>Meu HTML Semântico</h1>
    <p>
      Por Bruno Góis Mateus. Publicado em <time datetime="2023-02-27">2023</time>
    </p>
  </header>

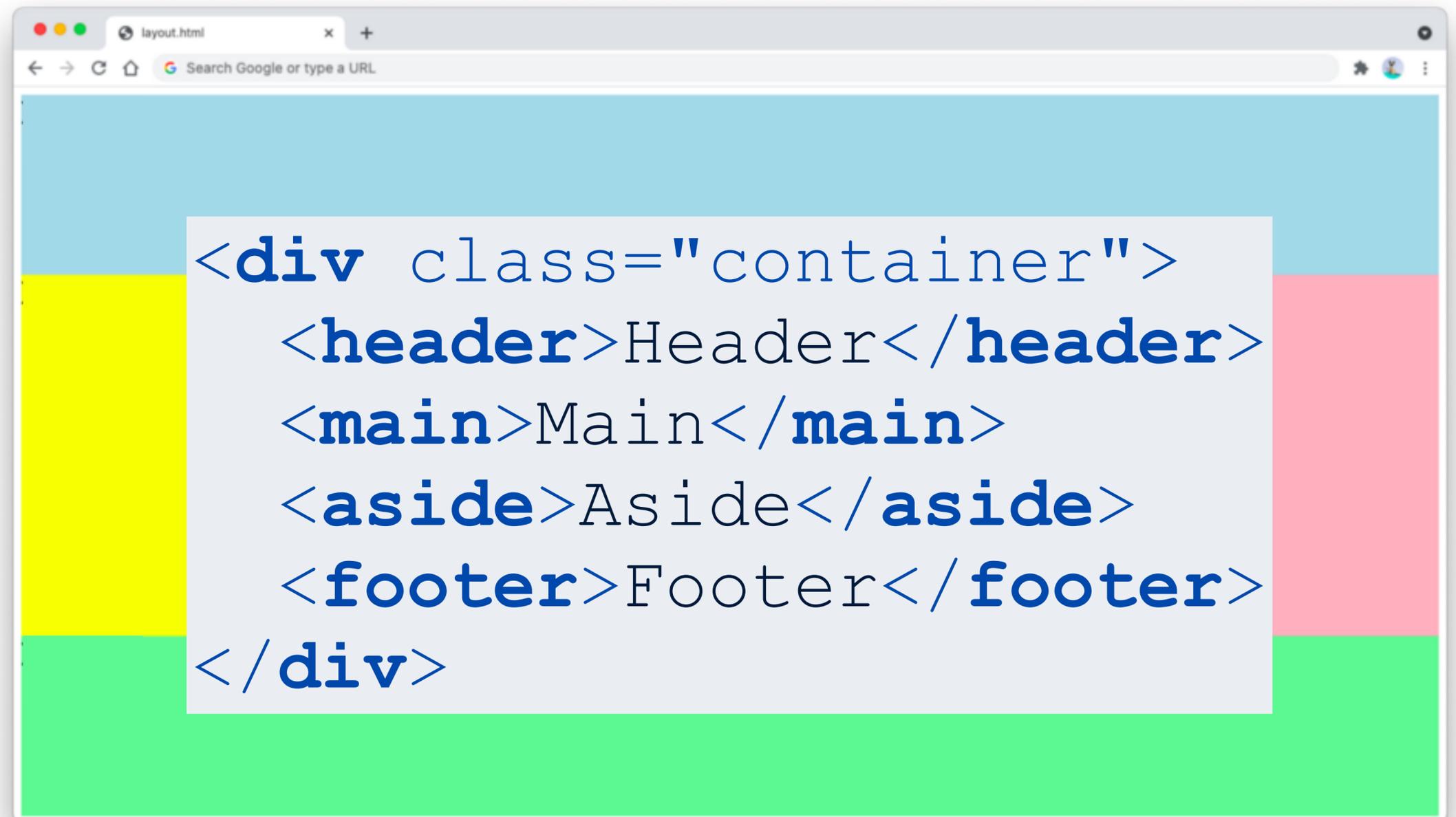
  <p>Um belo exemplo de HTML semântico</p>
  ...
  <footer>
    <address>
      Em caso de dúvidas, entrar em contato com <a href='mailto:brunomateus@exemplo.com'> Bruno</a>.
    </address>
  </footer>
</article>

<footer>
  <p>&copy; 2023 Bruno Góis Mateus</p>
</footer>
```

Layouts

Construindo layouts

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



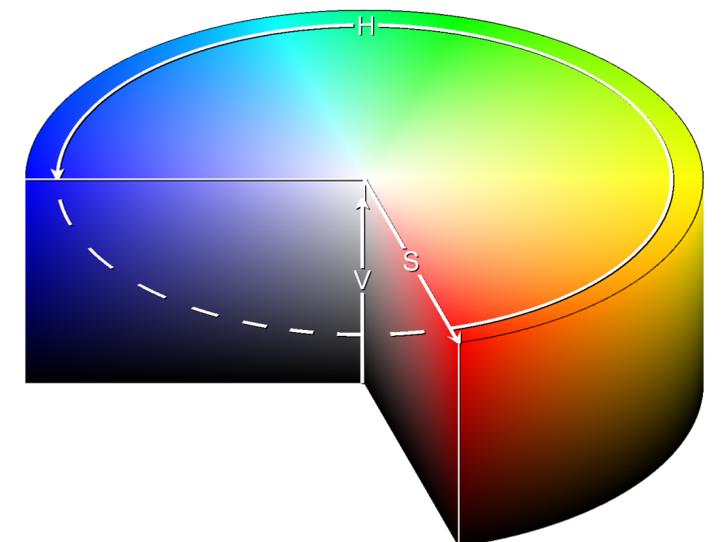
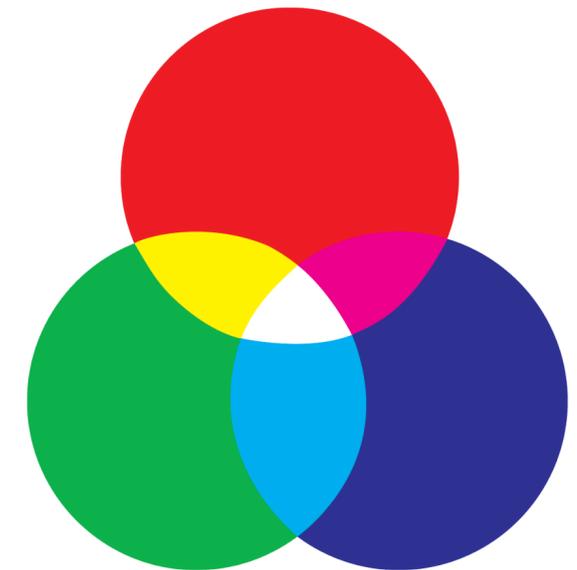
Cores em CSS



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 hlsa



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>
```

Olhem essa combinação de cores

Cores em CSS

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

Fontes em CSS



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

Maior prioridade

Mais de um palavra -> ""

Genérica

```
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  
}
```

- 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

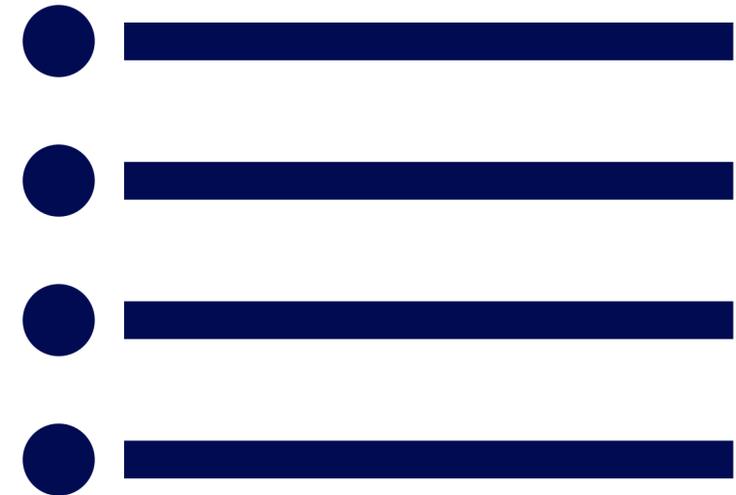
Propriedades de Texto

Propriedade	Descrição
text-align	Alinhamento do texto dentro do elemento
text-decoration	Decorações, ex: sublinhado
line-height word-spacing letter-spacing	Espaçamento entra 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

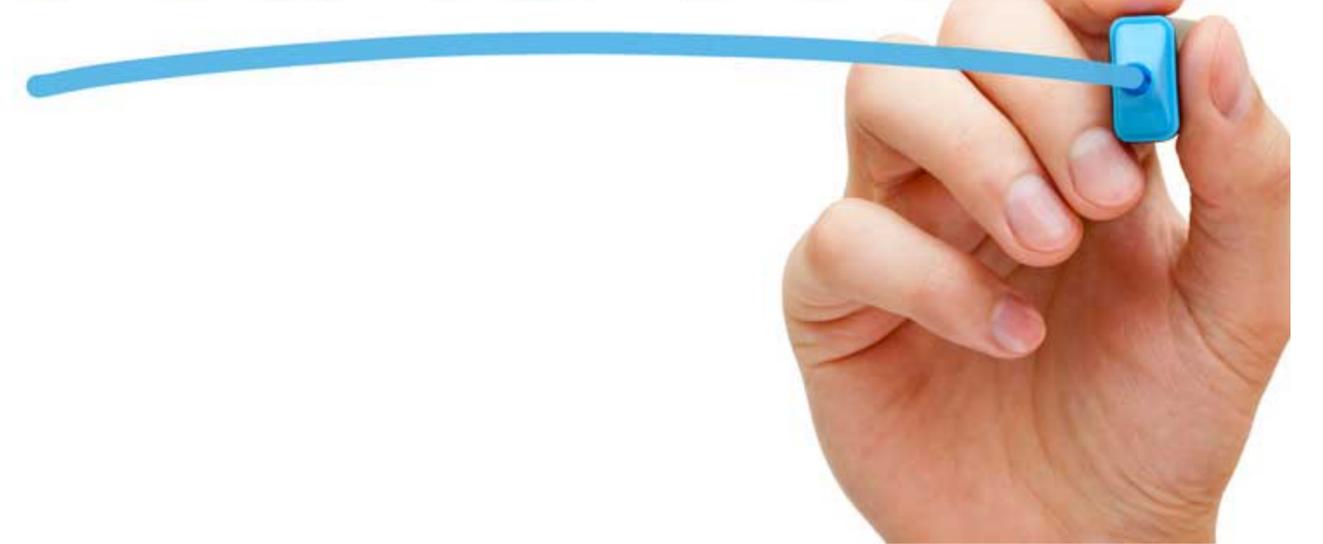
list-style-image

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

- [MDN Web Docs: CSS](#)
- [Hello, CSS | Interneting is Hard](#)
- [The Road to HTML 5: Link Relations](#)
- [HTML <link> media attribute](#)
- [MDN Web Docs: Pseudo-classes](#)
- [Meet the Pseudo Class Selectors](#)
- [CSS Pseudo-elements](#)

Referências

- [MDN Web Docs: Specificity](#)
- [The Box Model | Interneting is Hard](#)
- [Floats | Interneting is Hard](#)
- [Flexbox | Interneting is Hard](#)
- [Advanced Positioning | Interneting is Hard](#)
- [Responsive Desing | Interneting is Hard](#)
- [CSS Flexbox vs Grid layout](#)

Referências

- [Semantic HTML5 Elements Explained](#)
- [What Is Semantic HTML?](#)

Referências

Links úteis

- Borda, cores e esquema de cores
 - <https://html-css-js.com/css/generator/border-outline/>
 - <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>
 - <https://specificity.keegan.st/>
- 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