

Cascading Style Sheets (CSS)

O que é CSS?

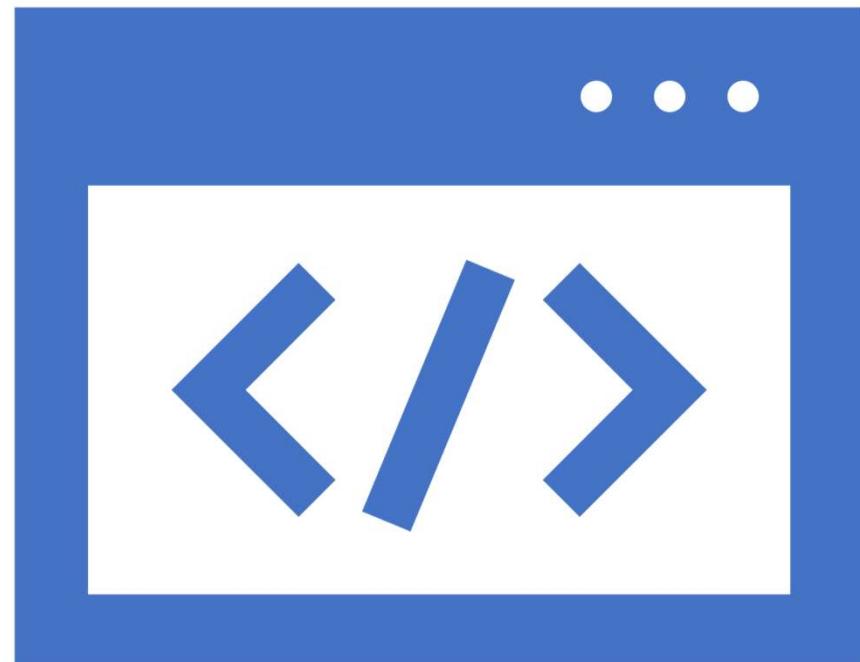
- Uma linguagem que define a aparência de um documento HTML;
- Ela descreve como os elementos HTML devem ser exibidos;
- Controla o layout de múltiplas páginas da web com apenas um arquivo;
- Os estilos externos são armazenados em arquivos CSS.



```
body {  
    font: x-small;  
    background: #  
    color: black;  
    margin: 0;  
    padding: 0;  
}
```

Motivo de usar o CSS

- Para poder separar a apresentação do conteúdo de uma página da Web;
- Inicialmente o HTML não podia estilizar o conteúdo;
 - Email do Marc Andreessen para o CERN
 - <http://ksi.cpsc.ucalgary.ca/archives/WWW-TALK/www-talk-1994q1.messages/643.html>
- W3C propôs o HTML 3.2 – introduziu novas tags para poder mudar o estilo do conteúdo HTML, porém hoje essas tags estão descontinuadas;
- W3C respondeu com o padrão CSS level 1
 - <https://www.w3.org/TR/REC-CSS1/>
- Permite adicionar estilos aos elementos HTML



Arte CSS



Homer

[View CSS](#)



Bart

[View CSS](#)



Marge

[View CSS](#)

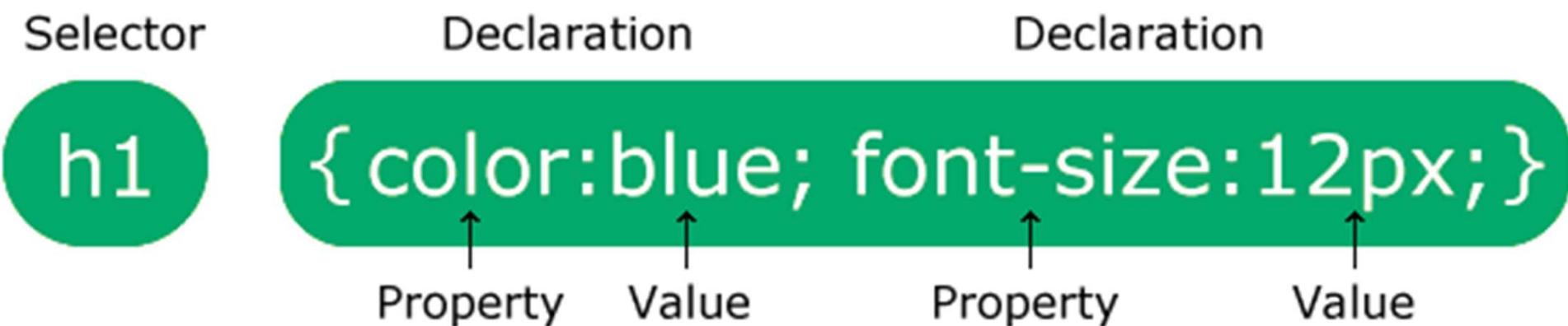


Lisa

[View CSS](#)

Como funciona o CSS?

- A regra CSS consiste em um seletor e um bloco de declaração;



Seletores CSS

- Seletor de elemento
- Seletor de classe

```
h3 {  
    color: red;  
}
```

```
.header-verde {  
    color: green  
}
```

||| Seletores CSS

- Seletor de id
- Seletor de atributo

```
#proposto {  
    color: blue  
}
```

```
p[lang] {  
    color: blueviolet;  
}
```



```
* {  
    background-color: antiquewhite;  
}
```

Seletores CSS

Seletor universal

Exercício 1

CSS Selectors

Páginas de portais de notícias

3. [G1 Notícias](#)
4. [Universo online - Uol](#)
5. [Blog do Pimenta](#)

CSS Selectors

Páginas de portais de notícias

3. [G1 Notícias](#)
4. [Universo online - Uol](#)
5. [Blog do Pimenta](#)

available space

[align-items](#)

Specifies the alignment for items inside a flexible container

[align-self](#)

Specifies the alignment for selected items inside a flexible container

[all](#)

Resets all properties (except unicode-bidi and direction)

[animation](#)

A shorthand property for all the *animation-** properties

[animation-delay](#)

Specifies a delay for the start of an animation

[animation-direction](#)

Specifies whether an animation should be played forwards, backwards or in alternate cycles

[animation-duration](#)

Specifies how long an animation should take to complete one cycle

[animation-fill-mode](#)

Specifies a style for the element when the animation is not playing (before it starts, after it ends, or both)

Propriedades

[Referência](#)

JULIA EVANS
@bark

CSS units

CSS has 2 kinds of units:
absolute & relative

absolute: px, pt, pc,
in, cm, mm

relative: em, rem,
vw, vh, %

0 is the same
in all units

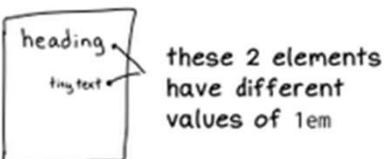
```
.btn {  
  margin: 0;  
}
```

also, 0 is different from none.
border: 0 sets the border width
and border: none sets the style

rem
the root element's
font size

1rem is the same
everywhere in the
document. rem is a
good unit for setting
font sizes!

em
the current element's
font size



these 2 elements
have different
values of 1em

1 inch = 96 px

on a screen, 1 CSS "inch"
isn't really an inch, and
1 CSS "pixel" isn't really
a screen pixel.
look up "device pixel
ratio" for more.

rem & em help with
accessibility

```
.modal {  
  width: 20rem;  
}
```

this scales nicely if the user
increases their browser's
default font size

Unidades CSS

Font-weight

font-weight: normal	bolder	lighter
font-weight: bold	bolder	lighter
font-weight: 100	bolder	lighter
font-weight: 200	bolder	lighter
font-weight: 300	bolder	lighter
font-weight: 400	bolder	lighter
font-weight: 500	bolder	lighter
font-weight: 600	bolder	lighter
font-weight: 700	bolder	lighter
font-weight: 800	bolder	lighter
font-weight: 900	bolder	lighter

Font-family

The screenshot shows the Google Fonts homepage with a dark theme. At the top, there's a search bar with the placeholder "Search fonts" and a "Filters" button. Below the search bar, it says "1576 of 1576 families" and "About these results". The first result listed is "Roboto" by Christian Robertson, which has 12 styles. A preview of the Roboto font is shown with the text "Everyone has the right to freedom of thou". The second result is "Noto Sans Kawi" by Google, described as a Variable (1 axis) font. A preview of the Noto Sans Kawi font is shown with the text "ઉચ્ચારણ અંગ્રેજી લાંબા ઉચ્ચારણ ક્રિપ્તિકાર્ય કરું". The third result is "Linefont" by Dmitry Ivanov, described as a Variable (2 axes) font. A preview of the Linefont font is shown with abstract zigzag patterns. The fourth result is "Agbalumo" by Raphael Alegbeleye, Sorkin Type, Eben Sorkin, with 1 style. A preview of the Agbalumo font is shown with the text "Everyone has the right to freedom of thou".

A large, stylized purple feather graphic is positioned on the left side of the slide. It has a long, sweeping shape with several smaller, branching offshoots. The color is a solid purple with some white highlights along the edges.

Color

- A propriedade **color** do CSS define o valor da cor de um elemento com seu conteúdo em texto e seus decorações (estilos);
- O valor hexadecimal em color é especificado como: #RRGGBB, onde RR (red), GG (green) e BB (blue), na faixa de valores: 00-FF; ou decimal 0-255
rgb(255,0,0)
- https://www.w3schools.com/css/css_color_hex.asp
- <https://colorhunt.co/>

Exercício 2

Aprenda cores em Italiano

Giallo



Rosso



Blue



Marrone

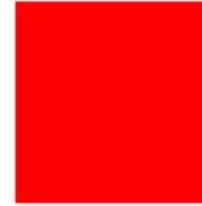


Aprenda cores em Italiano

Giallo



Rosso



Blue



Marrone



Exercício 3

Propriedades importantes do CSS

Cor

Tamanho da Fonte

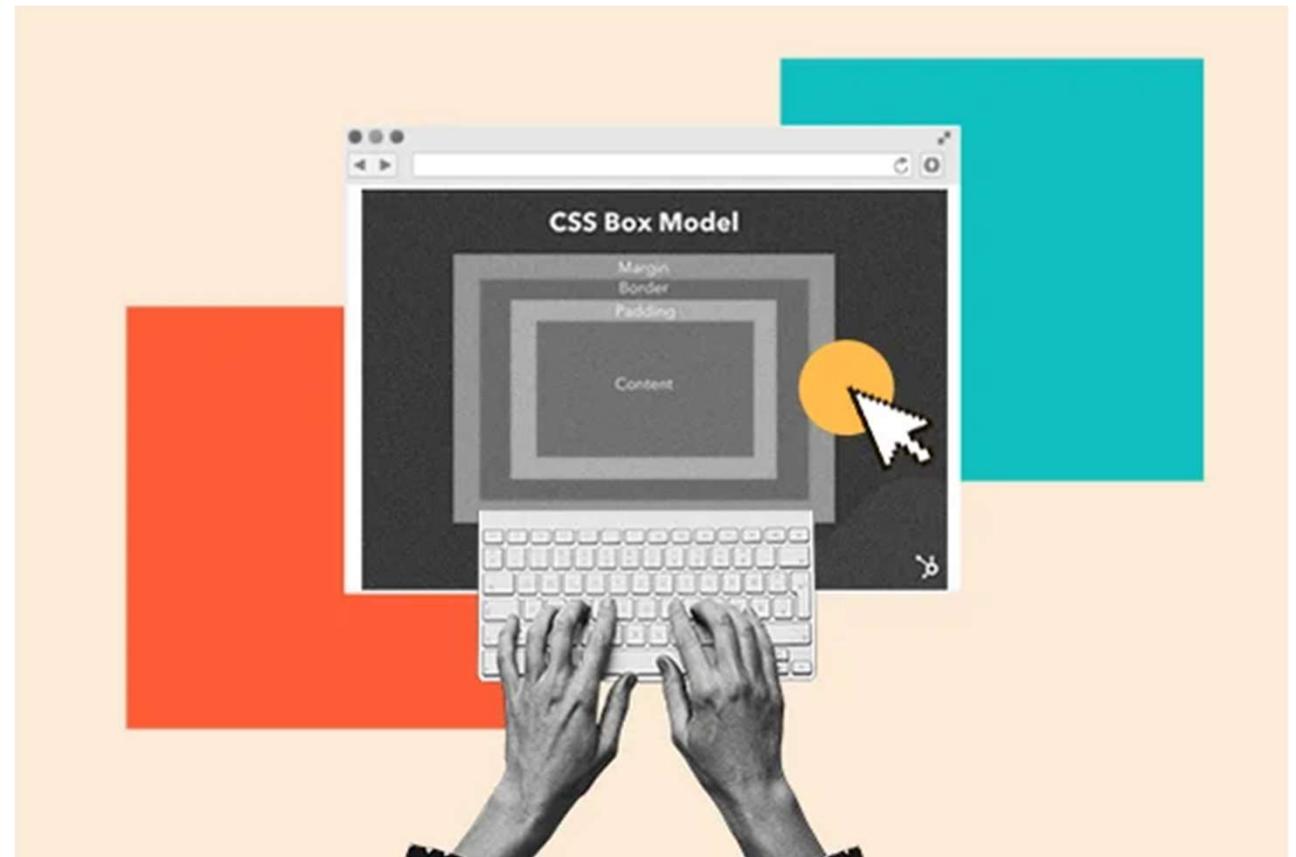
Fonte weight

Fonte Family

Text Align

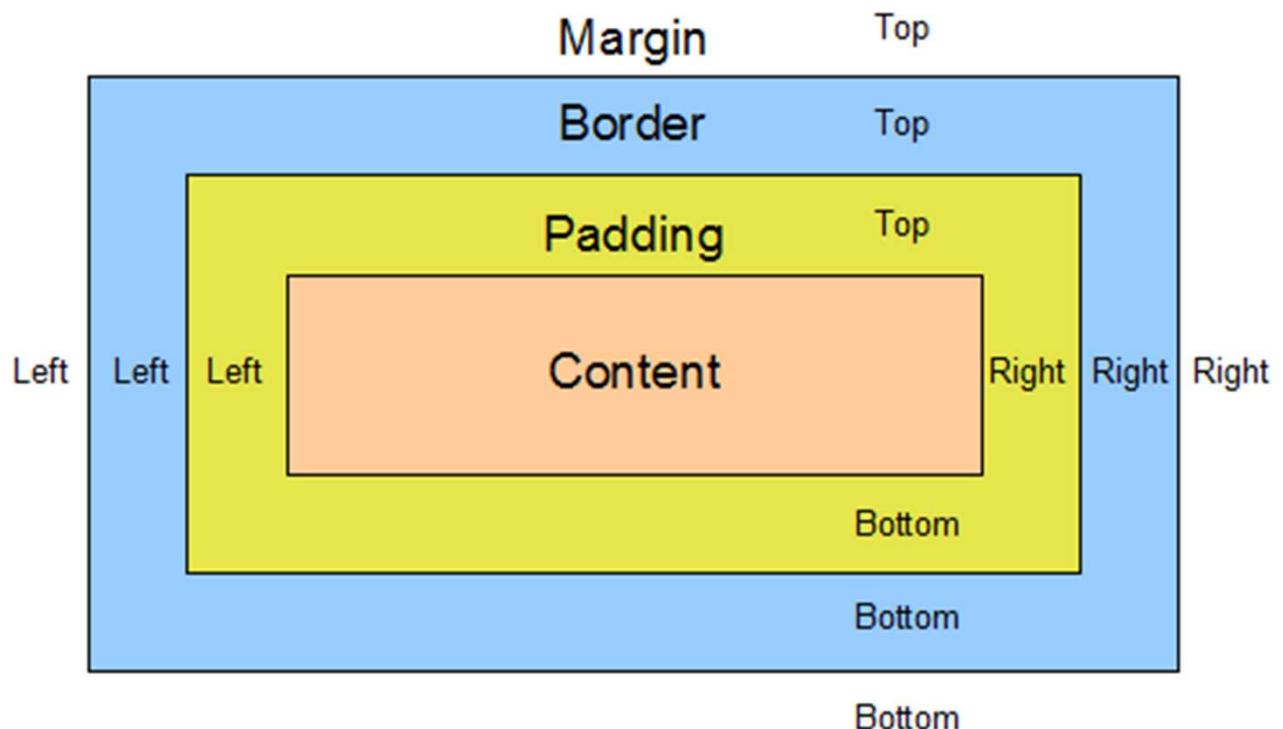
Box Model

- Cada elemento é representado como um *box* retangular;
- Determinar o tamanho, propriedades - como sua cor, fundo, estilo das bordas - e a posição desses *boxes* é o objetivo do mecanismo de renderização;
- No CSS, cada um desses *boxes* retangulares é descrita usando o *box model* padrão;
- Cada box possui 4 edges: **margin edge**, **border edge**, **padding edge** e **content edge**.



Box Model

- (content area) é a área ocupada pelo conteúdo real do elemento;
- Ele frequentemente possui um fundo, uma cor de fonte ou uma imagem;
- Suas dimensões são a largura do conteúdo, ou largura do box de conteúdo, e altura do conteúdo, ou altura do box de conteúdo.





Cascata CSS

- Como o CSS é aplicado ao HTML e como os conflitos entre as declarações de estilo são resolvidos;
- CSS significa **Folhas de Estilo em Cascata**, e a primeira palavra *cascading* é incrivelmente importante de se entender;
- A maneira como a cascata se comporta é a chave para entender o CSS;
- **Problema** – uma regra CSS que deveria estar aplicado a um elemento não funciona;
 - Criação de duas regras que aplicam valores diferentes da mesma propriedade ao mesmo elemento.

Cascata CSS

- O conteúdo `<h1>` acaba sendo colorido de azul. Isso ocorre porque ambas as regras são da mesma fonte, têm um seletor de elemento idêntico e, portanto, carregam a mesma especificidade, mas a última na ordem da fonte vence.

```
h1 {  
    color: red;  
}  
h1 {  
    color: blue;  
}
```

```
<h1>This is my heading.</h1>
```

This is my heading.



Cascata CSS

- Cascata e o conceito relacionado de especificidade são mecanismos que controlam qual regra se aplica quando há tal conflito;
- A regra que está estilizando o elemento pode não ser a que é esperada, então é preciso entender como esses mecanismos funcionam.

A silhouette of a person's head and shoulders facing right, superimposed on a night sky over mountains.

Nível de importância da regra CSS

- Position +++
- Specificity ++++++
- Type +++++++++++++++
- Importance
++++++

Position

```
h1 {  
    color: blue;  
    color: brown;  
    color: blueviolet;  
}
```

```
<h1>Position</h1>
```

Position

Especificidade do CSS

- é o algoritmo que o navegador usa para decidir qual valor de propriedade é aplicado a um elemento;
- Se vários blocos de estilo tiverem seletores diferentes que configuram a mesma propriedade com valores diferentes e visam o mesmo elemento, a especificidade decide o valor da propriedade que será aplicado ao elemento. A especificidade é basicamente uma medida de quão específica será a seleção de um seletor.

Especificidade do CSS

- O conteúdo `<h1>` abaixo acaba sendo colorido de vermelho porque o seletor de classe `main-heading` dá à sua regra uma especificidade maior

```
.main-heading {  
    color: red;  
}  
  
h1 {  
    color: blue;  
}  
  
<h1 class="main-heading">This is my heading.</h1>
```

This is my heading.

Especificidade do CSS

- Id (#) +++++++
- Attribute (tag[attribute]) +++
- Class (.) ++
- Element (tag) +

```
li {  
    color: aqua;  
}  
  
#prima-id{  
    color: black;  
}  
  
.prima-class{  
    color: blueviolet;  
}  
  
li[draggable]{  
    color: chartreuse  
}
```

```
<ul>  
|   <li id="prima-id" class="prima-class" draggable="true">primeiro</li>  
|   <li>segundo</li>  
|   <li>terceiro</li>  
</ul>
```

Specificity

- primeiro
- segundo
- terceiro



Type

- Ordem da aplicação dos estilos
 - EmLinha ++++++++
 - Interno +++++
 - Externo ++

Importance

- A regra CSS **!important** é usada para adicionar mais importância que o normal para a propriedade/valor;

```
li[draggable]{  
    color: #chartreuse !important;  
}
```

Texto em Amarelo

Texto em Branco

Texto em Amarelo

Texto em Branco

Exercício 4



Texto em Amarelo

Texto em Branco

Texto em Amarelo

Texto em Branco

```
h1 {  
    color: blue;  
}  
  
.special {  
    color: blue;  
}
```

Combinando Seletores CSS - Grupo

- Para mais de um item que usa o mesmo CSS, os seletores individuais podem ser combinados em uma *lista de seletores* para que a regra seja aplicada a todos os seletores individuais.

```
h1, .special {  
    color: blue;  
}
```

Combinando Seletores CSS - Child

- Seleciona parágrafos que são filhos diretos de <article>elementos usando o combinador filho (>);
- Seleciona o parágrafo que é filho direto da classe .Caixa e aplica a cor blue.

```
article > p {  
}  
}
```

```
/*Child Selector*/  
.caixa > p{  
    color: blue;  
}
```

Combinando Seletores CSS - Descendent

- Combina dois seletores para que os elementos correspondidos pelo segundo seletor sejam selecionados se eles tiverem um elemento ancestral (pai, pai ou mãe, pai ou mãe, etc.) correspondente ao primeiro seletor;

```
<ul>
  <li>
    <div>Item 1</div>
    <ul>
      <li>Subitem A</li>
      <li>Subitem B</li>
    </ul>
  </li>
  <li>
    <div>Item 2</div>
    <ul>
      <li>Subitem A</li>
      <li>Subitem B</li>
    </ul>
  </li>
</ul>
```

```
li {
  list-style-type: disc;
}

li li {
  list-style-type: circle;
}
```

- Item 1
 - Subitem A
 - Subitem B
- Item 2
 - Subitem A
 - Subitem B

Combinando Seletores CSS - Chaining

```
/*Chaining*/
li.usada{
    color: #rgb(165, 42, 118);
    font-weight: bold;
}
```

- Os seletores CSS podem ser encadeados para que os conjuntos de regras se apliquem apenas aos elementos que correspondem a todos os critérios;

Exercício 5

Listas de roupas para doar

Abrigo São José

Roupas

- Camisa Preta
- Calça Amarela
- Moletom

Alimentos

- Feijão
- Arroz
- Carne

Móveis

- Cama
- Cadeira
- Banco
- Sofá

Outros Materiais

Listas de roupas para doar

Abrigo São José

Roupas

- Camisa Preta
- Calça Amarela
- Moletom

Alimentos

- Feijão
- Arroz
- Carne

Móveis

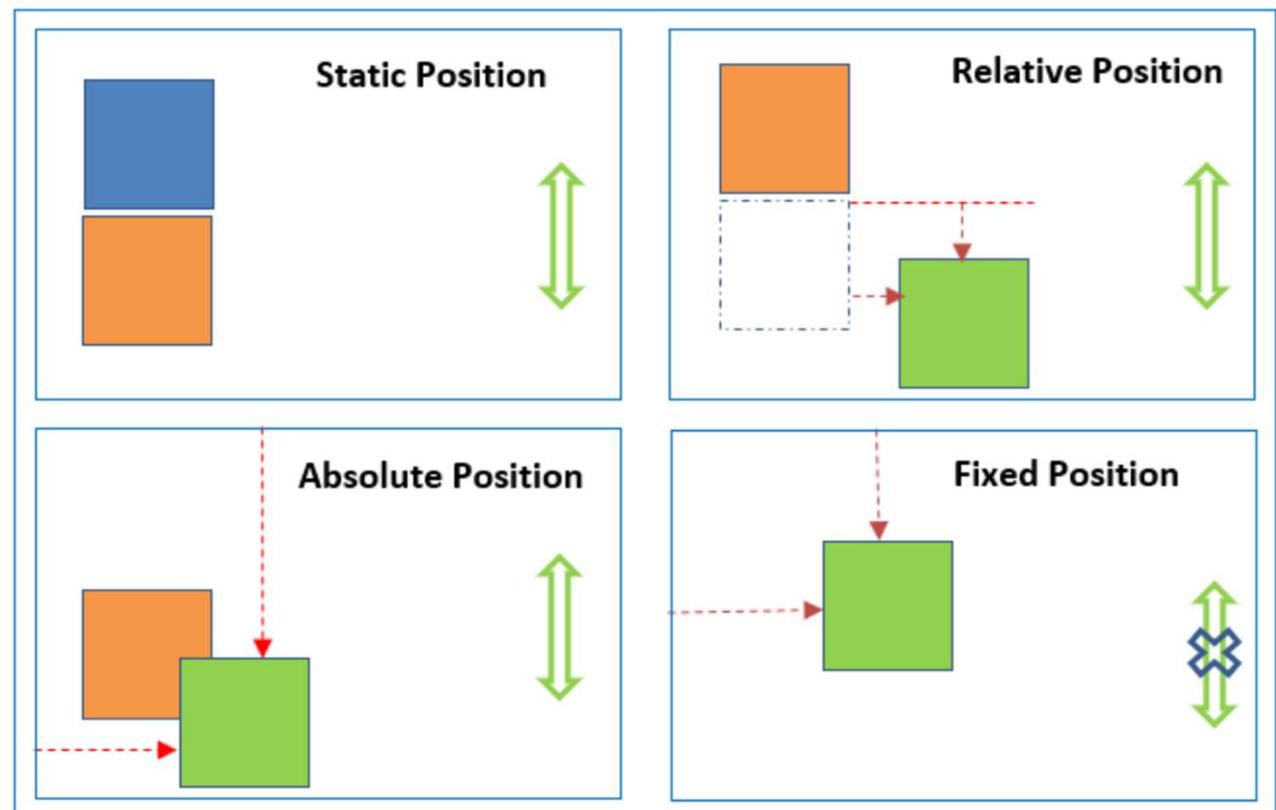
- Cama
- Cadeira
- Banco
- Sofá

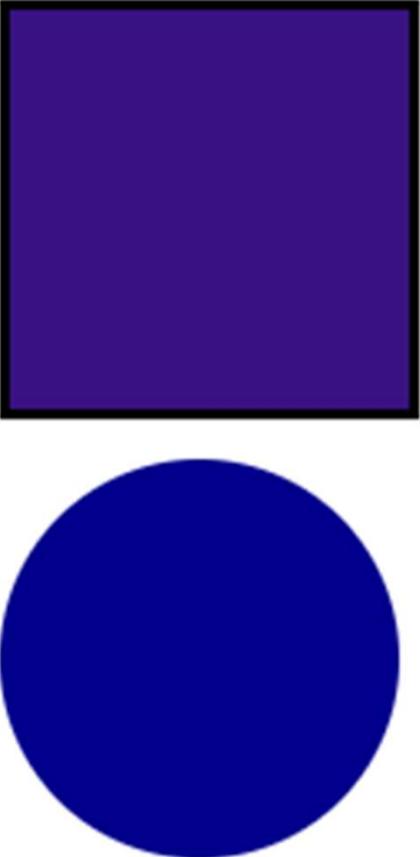
Outros Materiais

Pasta 2.7_combinandoSelectors- não modifique o index.html, só acrescente o arquivo de estilo

Posicionamiento

- Estático
- Relativo
- Absolute
- Fixo

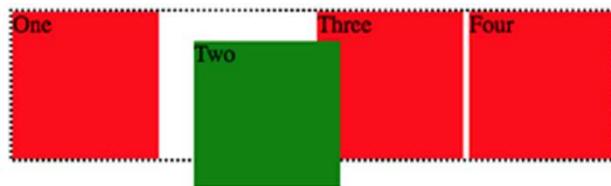




Estático

- Elementos HTML são posicionados de forma estática por padrão;
- Os elementos posicionados de forma estática não são afetados pelas propriedades top, bottom, left e right.
- Está sempre posicionado de acordo com o fluxo normal da página:

Relativo



- Um elemento com `position: relative;` está posicionado em relação à sua posição normal;
- Definir as propriedades `top`, `right`, `bottom` e `left` de um elemento relativamente posicionado fará com que ele seja ajustado fora de sua posição normal;
- Outros conteúdos não serão ajustados para caber em qualquer lacuna deixada pelo elemento.

Absolute

- Um elemento com position: absolute; é posicionado em relação ao ancestral posicionado mais próximo (em vez de posicionado em relação à janela de visualização, como fixo).



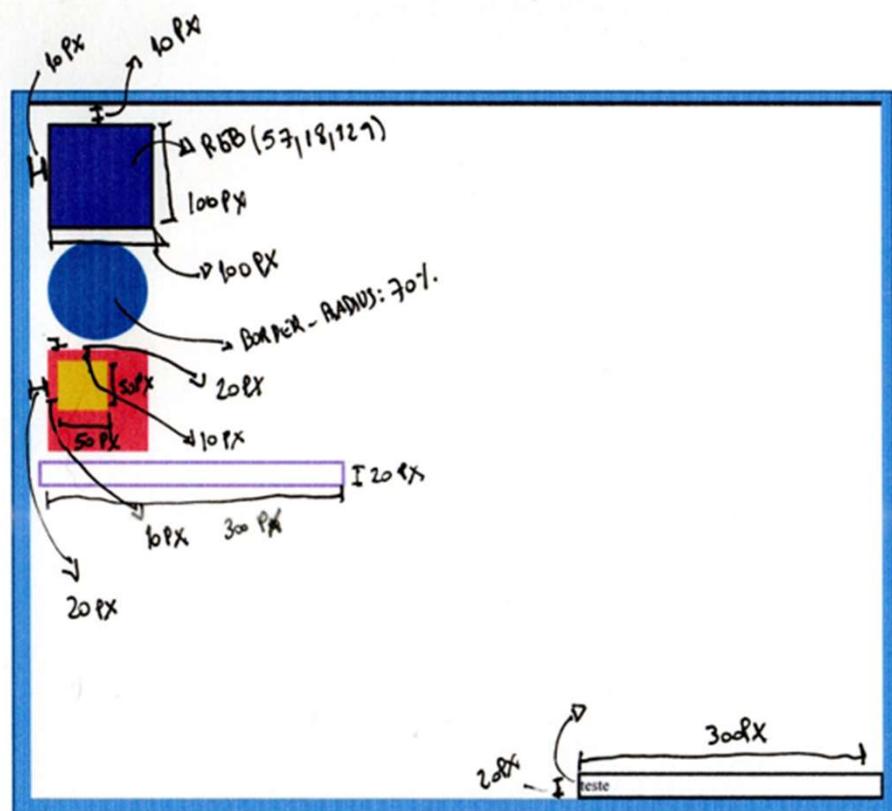
```
#caixa4 {  
    background-color: #rgb(224, 235, 9);  
    width: 50px;  
    height: 50px;  
    position: absolute;  
    left: 10px;  
    top: 10px;  
}
```

Fixo

- Um elemento com position: fixed; é posicionado em relação à janela de visualização, o que significa que sempre permanece no mesmo lugar, mesmo se a página for rolada.
- As propriedades top, right, bottom e left são usadas para posicionar o elemento.

teste

Exercicio 6



Display

- Especifica se/como um elemento é exibido;
- Todo o elemento tem um display default dependendo qual o tipo do elemento. O valor default do display para a maioria dos elementos é block or inline;
- O valor inline não permite utilizar width/height para definir largura e altura;
- O valor None faz com que o elemento não seja exibido.



Block-level Elements

- Um elemento de nível de bloco sempre começa em uma nova linha e ocupa toda a largura disponível (estende-se para a esquerda e para a direita tanto quanto possível).
 - <div>
 - <h1> - <h6>
 - <p>
 - <form>
 - <header>
 - <footer>
 - <section>



Inline Elements



- Um elemento inline não começa em uma nova linha e ocupa apenas a largura necessária.
 -
 - <a>
 -

Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac...



```
img {  
    float: right;  
}
```

float

- A propriedade CSS float especifica como um elemento deve flutuar;
 - Especifica que uma imagem deve flutuar à direita em um texto.

```
img {  
    float: left;  
}
```

float

- Especifica que uma imagem deve flutuar para a esquerda em um texto.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac...

float

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  float: left;
  padding: 15px;
}

.div1 {
  background: red;
}

.div2 {
  background: yellow;
}

.div3 {
  background: green;
}
</style>
</head>
<body>

<h2>Float Next To Each Other</h2>

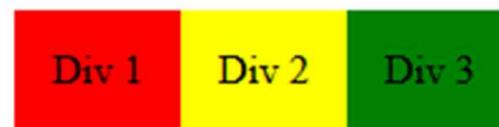
<p>In this example, the three divs will float next to each other.</p>

<div class="div1">Div 1</div>
<div class="div2">Div 2</div>
<div class="div3">Div 3</div>

</body>
</html>
```

Float Next To Each Other

In this example, the three divs will float next to each other.



Exercício 7

World War II Fighters



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.

World War II Fighters

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.

clear

- Quando a propriedade float é usada e é preciso que o próximo elemento abaixo (não à direita ou à esquerda), é preciso usar a propriedade clear;
- A propriedade clear especifica o que deve acontecer com o elemento próximo a um elemento flutuante;
- A propriedade clear pode ter um dos seguintes valores:
 - None: o elemento não é colocado abaixo dos elementos flutuantes à esquerda ou à direita. Este é o padrão;
 - Left: o elemento é empurrado abaixo dos elementos flutuantes à esquerda;
 - Right: o elemento é colocado abaixo dos elementos flutuantes à direita;
 - Both: o elemento é colocado abaixo dos elementos flutuantes esquerdo e direito;
 - Inherit: o elemento herda o valor claro de seu pai.

clear



```
.div1 {  
  float: left;  
  padding: 10px;  
  border: 3px solid #73AD21;  
}  
  
.div2 {  
  padding: 10px;  
  border: 3px solid red;  
}  
  
.div3 {  
  float: left;  
  padding: 10px;  
  border: 3px solid #73AD21;  
}  
  
.div4 {  
  padding: 10px;  
  border: 3px solid red;  
  clear: left;  
}
```

Without clear

div1 div2 - Notice that div2 is after div1 in the HTML code. However, since div1 floats to the left, the text in div2 flows around div1.

With clear

div3

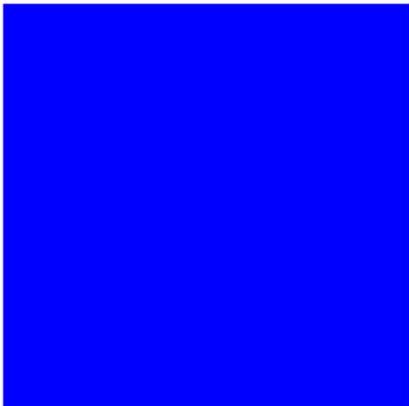
div4 - Here, clear: left; moves div4 down below the floating div3. The value "left" clears elements floated to the left. You can also clear "right" and "both".

Media query

- é uma técnica CSS introduzida no CSS3;
- Ele usa a regra @media para incluir um bloco de propriedades CSS somente se uma determinada condição for verdadeira.

Media query

Media query example



Media query example



```
.caixa {  
    background-color: blue;  
    width: 200px;  
    height: 200px;  
}  
  
/*entre 150px e 200px*/  
@media (min-width: 150px) and (max-width: 300px) {  
    .caixa {  
        background-color: pink;  
        width: 50px;  
        height: 50px;  
    }  
}
```

Exercício 8

- Troque a cor de fundo de acordo com o tamanho da tela dos dispositivos
- @media (max-width:772px)
- @media (min-width:773px) and (max-width:1100px)
- @media (min-width:1101px)

Grid Layout

- O CSS Grid Layout oferece um sistema de layout baseado em grade, com linhas e colunas, que facilita o design de páginas web sem a necessidade de usar flutuadores e posicionamento.
- Um layout de grade consiste em um elemento pai, com um ou mais elementos filhos.

modo estático



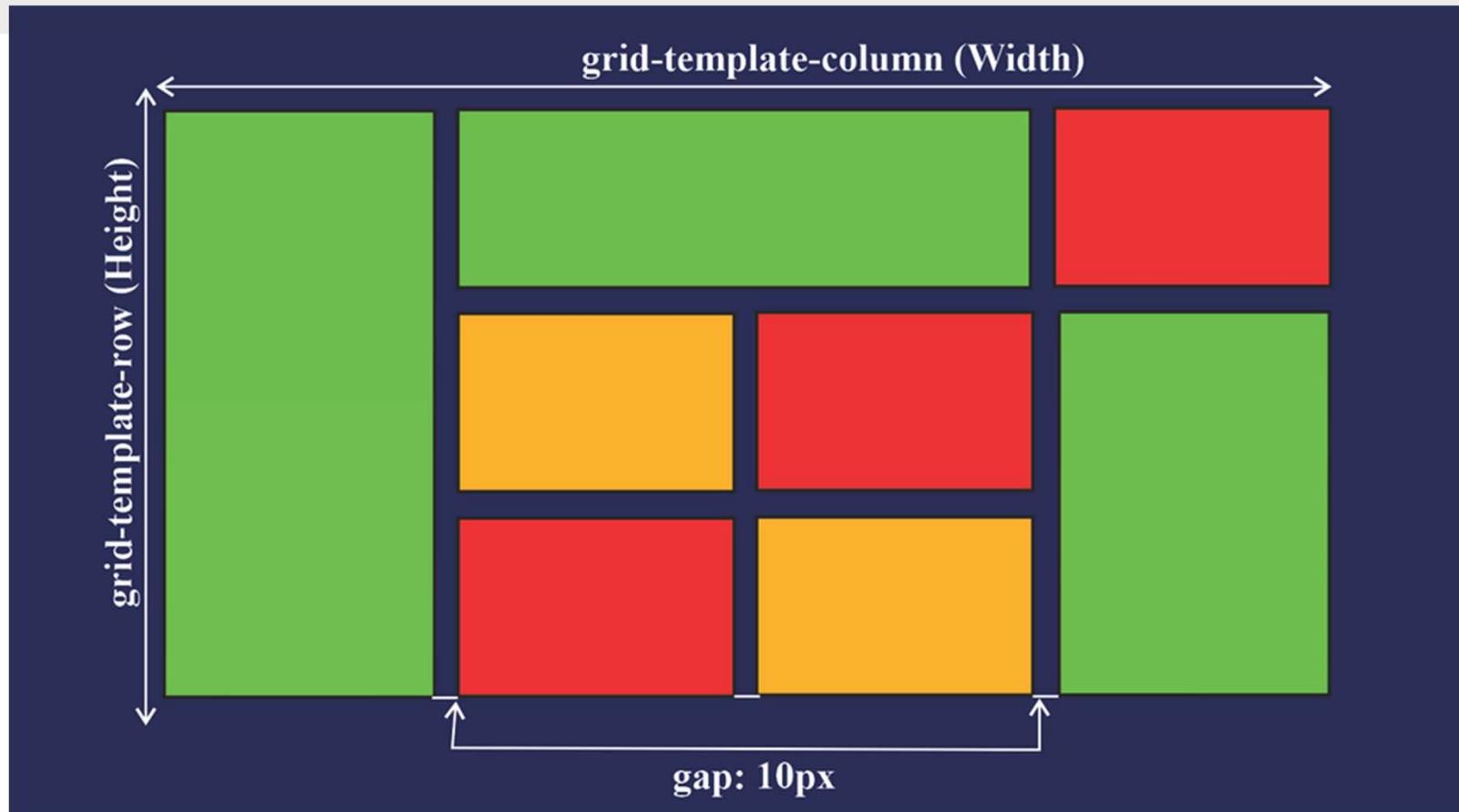
manho: auto



manho: fractional ratio combinado



Grid Layout

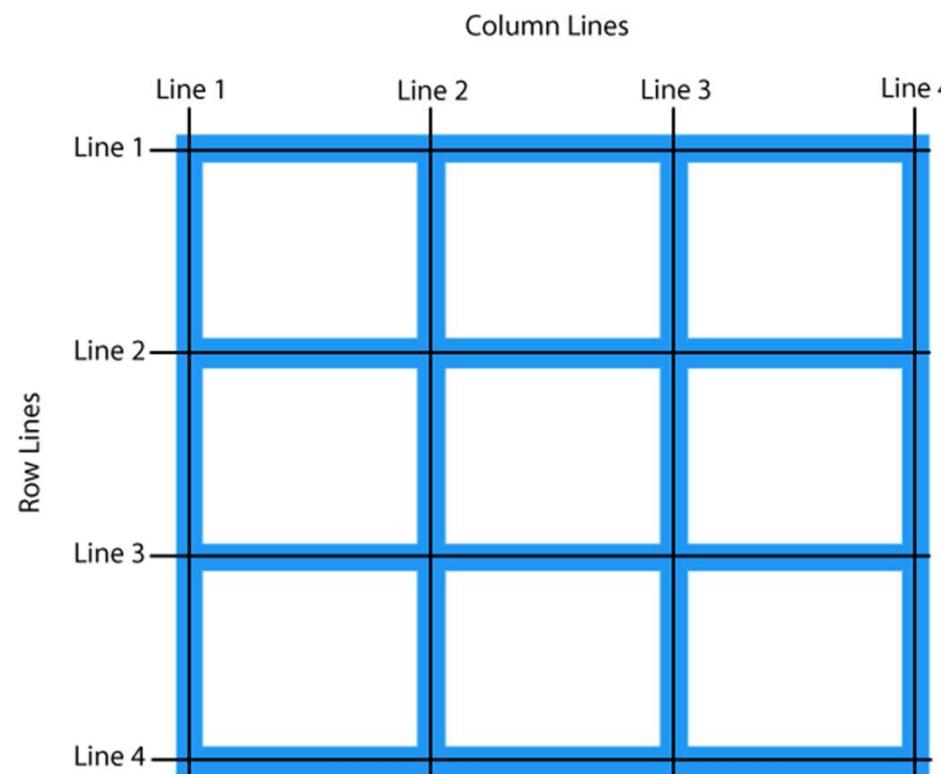


Grid Layout

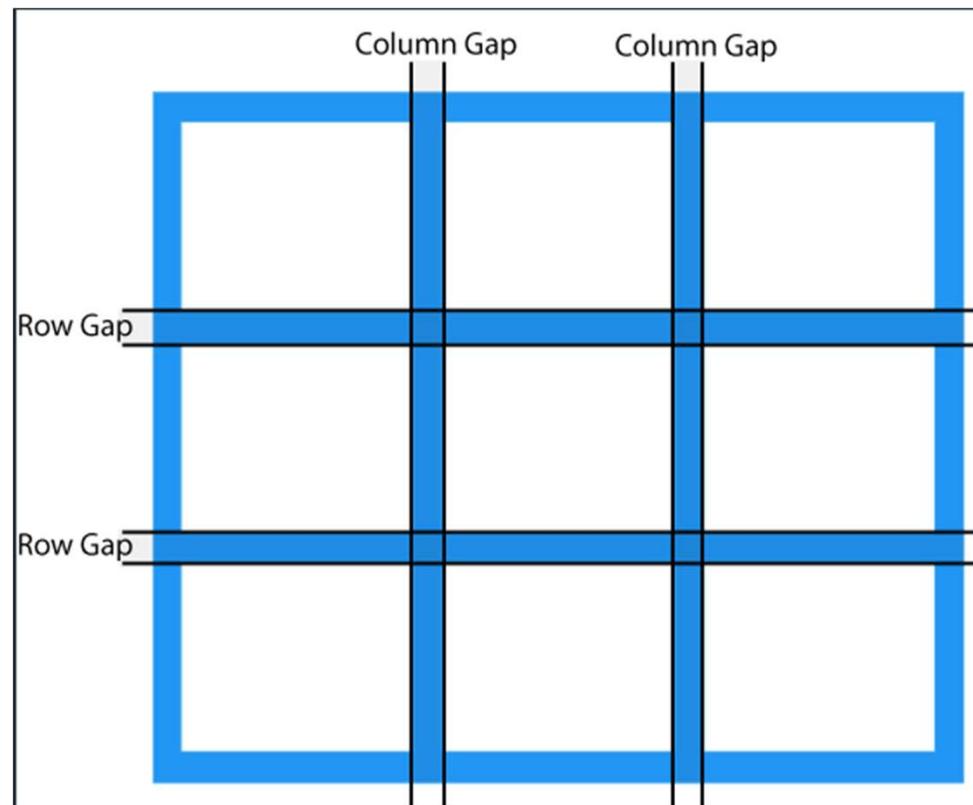
```
<div class="grid-container">
  <div class="grid-item">1</div>
  <div class="grid-item">2</div>
  <div class="grid-item">3</div>
  <div class="grid-item">4</div>
  <div class="grid-item">5</div>
  <div class="grid-item">6</div>
  <div class="grid-item">7</div>
  <div class="grid-item">8</div>
  <div class="grid-item">9</div>
</div>
```

1	2	3
4	5	6
7	8	9

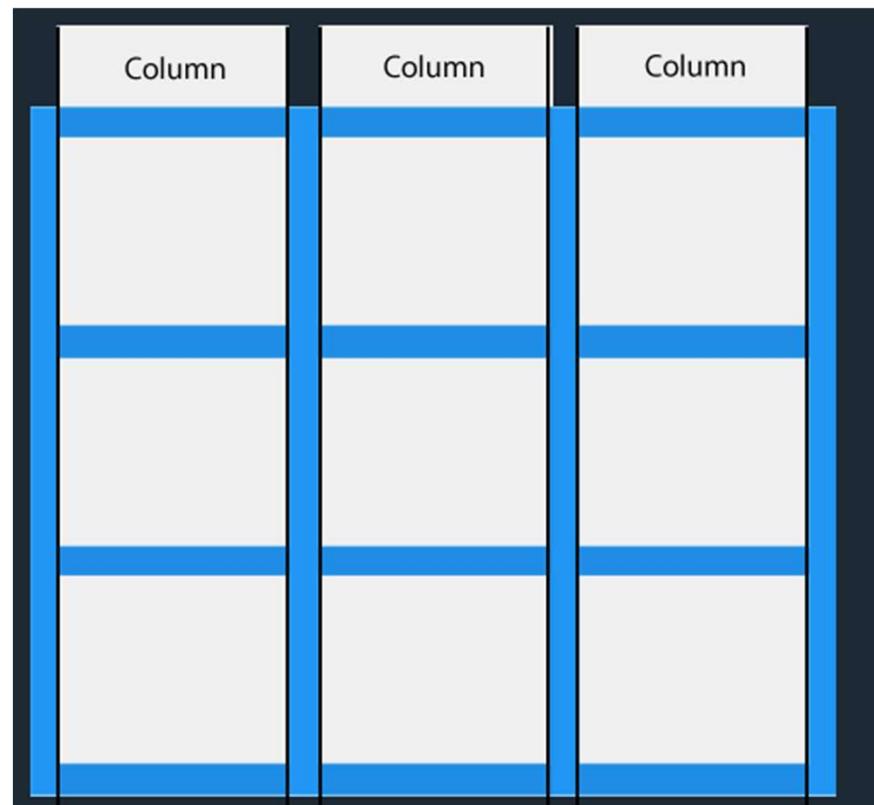
Grid Lines



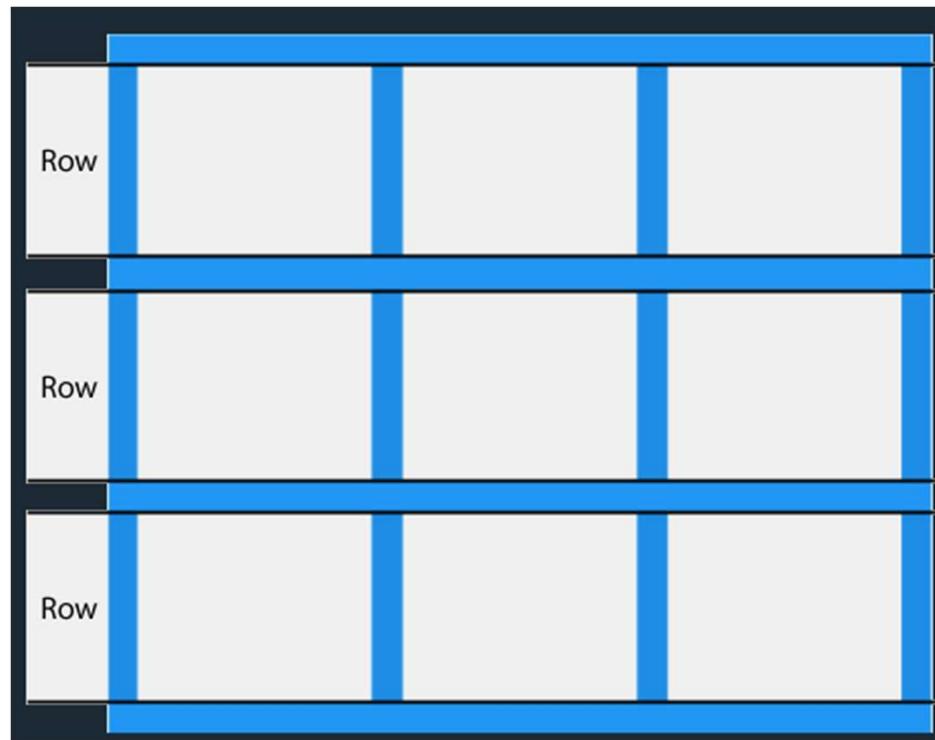
Grid Gaps



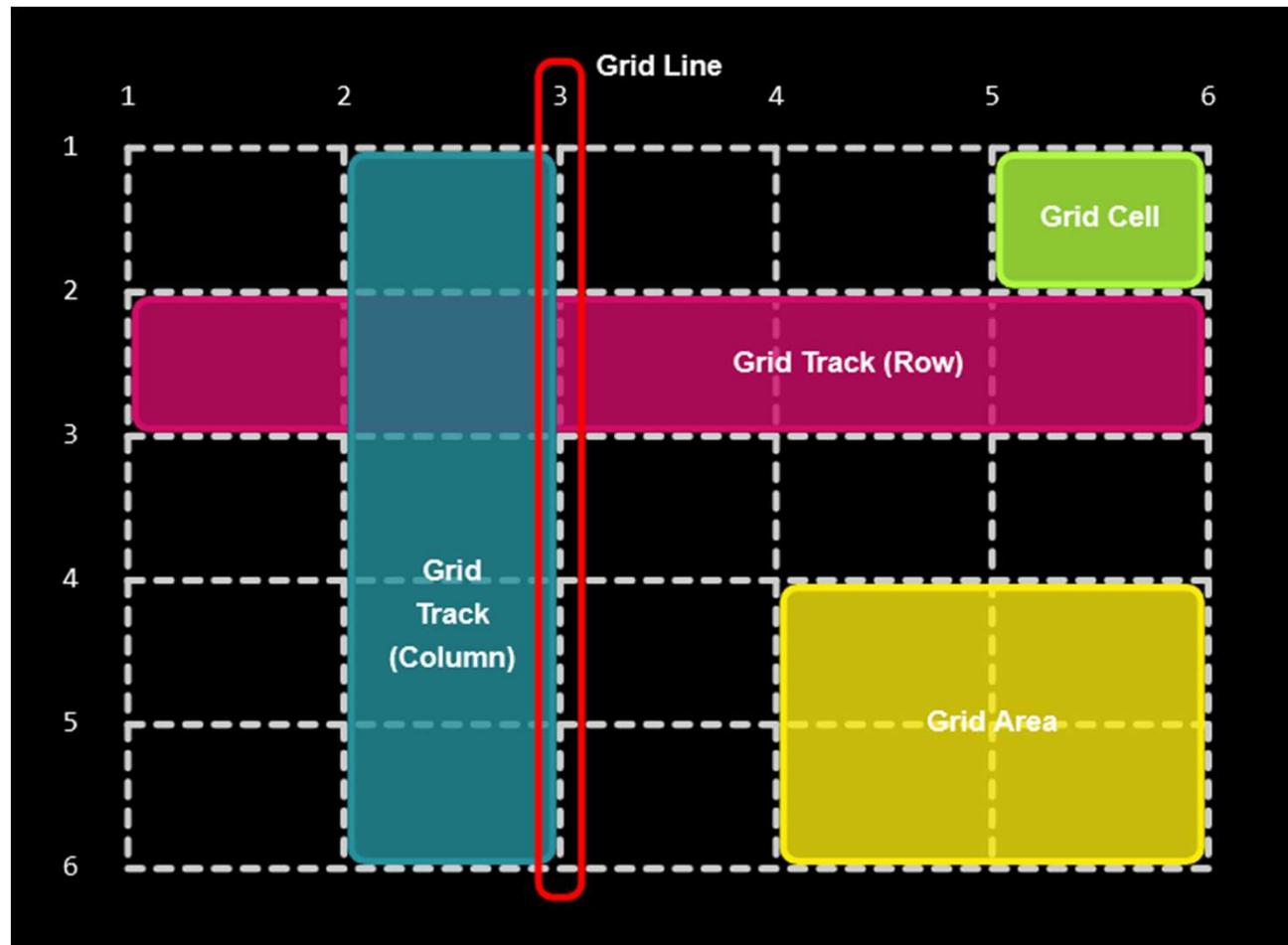
Grid Columns



Grid Rows



Grid Layout



Visão geral das Properties

Container	Item
grid-template-rows	grid-row-start
grid-template-columns	grid-row-end
grid-template-areas	grid-column-start
grid-row-gap	grid-column-end
grid-column-gap	grid-gap
justify-items	justify-self
align-items	align-self
justify-content	
align-content	order
grid-auto-rows	
grid-auto-columns	
grid-auto-flow	

Grid container

- Cria-se um grid container com as declarações CSS display: grid para um elemento da marcação. Assim declarando, todos os elementos filhos diretos daquele container se transformam em grid items;
- Todos os elementos filhos diretos agora são grid items;

```
<div class="wrapper">  
  <div>One</div>  
  <div>Two</div>  
  <div>Three</div>  
  <div>Four</div>  
  <div>Five</div>  
</div>
```

```
.wrapper {  
  display: grid;  
}
```

Grid Layout 3x3

HTML

```
1 <div class="container">
2   <div class="item A1">A1</div>
3   <div class="item A2">A2</div>
4   <div class="item A3">A3</div>
5   <div class="item A4">A4</div>
6   <div class="item A5">A5</div>
7   <div class="item A6">A6</div>
8   <div class="item A7">A7</div>
9   <div class="item A8">A8</div>
10  <div class="item A9">A9</div>
11 </div>
```

CSS

```
1 .container {
2   background-color: #F4BF96;
3   margin: 10px auto;
4   display: grid;
5   grid-template-columns: 50px 50px 50px;
6   grid-template-rows: 100px 100px 100px;
7   grid-row-gap: 20px;
8   grid-column-gap: 20px;
9 }
10
11 .item {
12   padding: 20px;
13   font-size: 20px;
14   font-family: sans-serif;
```

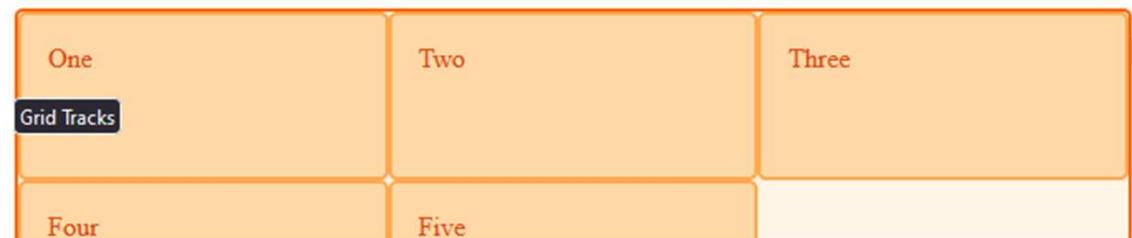
A1	A2	A3
A4	A5	A6
A7	A8	A9

Unidade fr

- A unidade **fr** representa uma fração do espaço disponível no container do grid.

```
<div class="wrapper">  
  <div>One</div>  
  <div>Two</div>  
  <div>Three</div>  
  <div>Four</div>  
  <div>Five</div>  
</div>
```

```
.wrapper {  
  display: grid;  
  grid-template-columns: 1fr 1fr 1fr;  
}
```



Grid Layout 3x3

HTML

```
1 <div class="container">
2   <div class="item A1">A1</div>
3   <div class="item A2">A2</div>
4   <div class="item A3">A3</div>
5   <div class="item A4">A4</div>
6   <div class="item A5">A5</div>
7   <div class="item A6">A6</div>
8   <div class="item A7">A7</div>
9   <div class="item A8">A8</div>
10  <div class="item A9">A9</div>
11 </div>
```

CSS

```
1 .container {
2   background-color: #F4BF96;
3   margin: 10px auto;
4   display: grid;
5   grid-template-columns: 1fr 20% 1fr;
6   grid-template-rows: repeat(3, 100px);
7   grid-row-gap: 20px;
8   grid-column-gap: 20px;
9 }
10
11 .item {
12   padding: 20px;
13   font-size: 20px;
14   font-family: sans-serif;
```

A1

A2

A3

A4

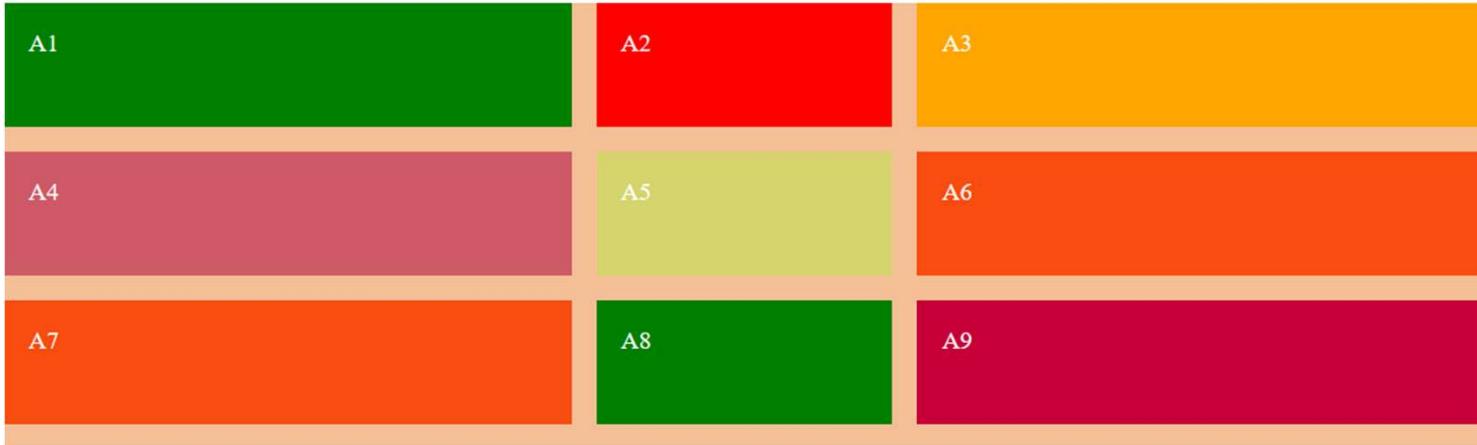
A5

A6

A7

A8

A9

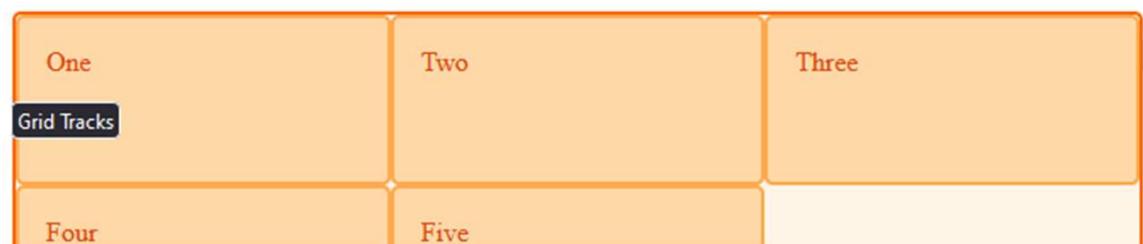


Grid Tracks

- As linhas e as colunas do grid são definidas com as propriedades **grid-template-columns** e **grid-template-rows**. Isso define o grid tracks. Um grid track é o espaço entre duas linhas em um grid.

```
<div class="wrapper">  
  <div>One</div>  
  <div>Two</div>  
  <div>Three</div>  
  <div>Four</div>  
  <div>Five</div>  
</div>
```

```
.wrapper {  
  display: grid;  
  grid-template-columns: 200px 200px 200px;  
}
```



grid implícito e grid explícito

- Ao criar o grid do slide anterior, foi definido a coluna de trilhas com a propriedade **grid-template-columns**;
- Foi deixado que o grid criasse as linhas necessárias para qualquer outro conteúdo;
- Estas linhas são criadas no grid implícito. O grid explícito é constituído por linhas e colunas que é definido com **grid-template-columns** e **grid-template-rows**;

grid-auto-rows: 50px;

HTML

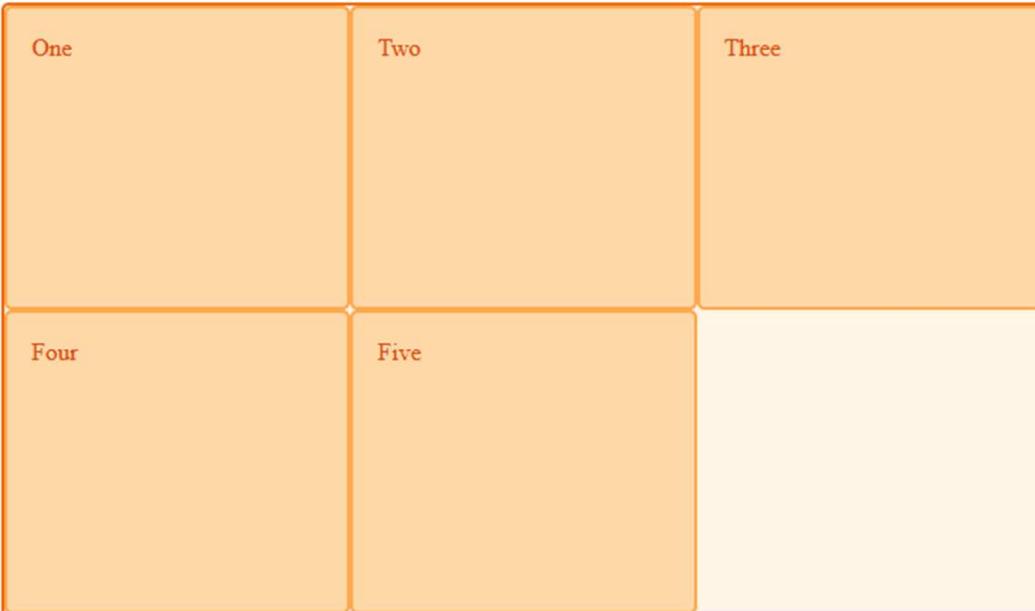
```
1 <div class="container">
2   <div class="item A1">A1</div>
3   <div class="item A2">A2</div>
4   <div class="item A3">A3</div>
5   <div class="item A4">A4</div>
6   <div class="item A5">A5</div>
7   <div class="item A6">A6</div>
8   <div class="item A7">A7</div>
9   <div class="item A8">A8</div>
10  <div class="item A9">A9</div>
11 </div>
```

CSS

```
1 <div class="container" style="display: grid;
2   grid-template-columns: 1fr 1fr;
3   grid-template-rows: repeat(1, 200px);
4   gap: 10px;
5   grid-auto-rows: 50px;">
6     <div class="item A1">A1</div>
7     <div class="item A2">A2</div>
8     <div class="item A3">A3</div>
9     <div class="item A4">A4</div>
10    <div class="item A5">A5</div>
11    <div class="item A6">A6</div>
12    <div class="item A7">A7</div>
13    <div class="item A8">A8</div>
14    <div class="item A9">A9</div>
15  </div>
16  &.item {
17    padding: 20px;
18    font-size: 20px;
19    font-family: sans-serif;
20    color: white;
21    background-color: green;
22    text-align: center;
23    border-radius: 10px;
24  }
25  &.A1 {
26    background-color: green;
27  }
28  &.A2 {
29    background-color: red;
30  }
31  &.A3 {
32    background-color: orange;
33  }
34  &.A4 {
35    background-color: purple;
36  }
37  &.A5 {
38    background-color: yellow;
39  }
40  &.A6 {
41    background-color: orange;
42  }
43  &.A7 {
44    background-color: red;
45  }
46  &.A8 {
47    background-color: green;
48  }
49  &.A9 {
50    background-color: maroon;
51  }
```



grid-template-rows



```
<div class="wrapper">
  <div>One</div>
  <div>Two</div>
  <div>Three</div>
  <div>Four</div>
  <div>Five</div>
</div>
```

```
.wrapper {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  grid-auto-rows: 200px;
}
```

grid-row-start, grid-column-start

Untitled ↗
Jorge

HTML

```
1 <div class="container">
2   <div class="item A1">A1</div>
3   <div class="item A2">A2</div>
4   <div class="item A3">A3</div>
5   <div class="item A4">A4</div>
6   <div class="item A5">A5</div>
7   <div class="item A6">A6</div>
8   <div class="item A7">A7</div>
9   <div class="item A8">A8</div>
10  <div class="item A9">A9</div>
11 </div>
```

CSS

```
22 & .A3 {
23   background-color: orange;
24 }
25 &.A4 {
26   background-color: #CE5A67;
27 }
28 &.A5 {
29   background-color: #D6D46D;
30   grid-row-start: 3;
31   grid-column-start: 3;
32 }
33 &.A6 {
34   background-color: #F94C10;
```

The visual representation shows a 3x3 grid of colored boxes:

- Row 1: A1 (green), A2 (red), A3 (yellow)
- Row 2: A4 (pink), A6 (orange), A7 (orange)
- Row 3: A8 (green), A9 (red), A5 (light green)

Blue arrows point from the text "grid-row-start: 3;" and "grid-column-start: 3;" in the CSS code to the corresponding positions of item A6 in the grid diagram.

grid-row-start, grid-row-end

HTML

```
1 <div class="container">
2   <div class="item A1">A1</div>
3   <div class="item A2">A2</div>
4   <div class="item A3">A3</div>
5   <div class="item A4">A4</div>
6   <div class="item A5">A5</div>
7   <div class="item A6">A6</div>
8   <div class="item A7">A7</div>
9   <div class="item A8">A8</div>
10  <div class="item A9">A9</div>
11 </div>
```

CSS

```
35 background-color:#F94C10;
36 }
37 &.A7 {
38   background-color:#F94C10;
39 }
40 &.A8 {
41   background-color:green;
42 }
43 &.A9 {
44   background-color:#C70039;
45   grid-row-start: 1;
46   grid-row-end: 3;
47 }
```

<div style="

grid-row: 3/4 | grid-column: 3/4;

```
HTML
```

```
1 <div class="container">
2   <div class="item A1">A1</div>
3   <div class="item A2">A2</div>
4   <div class="item A3">A3</div>
5   <div class="item A4">A4</div>
6   <div class="item A5">A5</div>
7   <div class="item A6">A6</div>
8   <div class="item A7">A7</div>
9   <div class="item A8">A8</div>
10  <div class="item A9">A9</div>
11 </div>
```

```
CSS
```

```
35 background-color:#F94C10;
36 }
37 .A7 {
38   background-color:#F94C10;
39 }
40 .A8 {
41   background-color:green;
42   grid-row: 3 / 4; /* grid-row-start: 3, grid-row-end: 4*/
43   grid-column: 3 / 4; /*grid-column-start: 3, grid-column-end: */
44 }
45 .A9 {
46   background-color:#C70039;
47   grid-row-start: 1;
```

A9



A1

A2

A3

A4

A6

A7

A8

grid-area: 1 / 2 / 3 / 4;

HTML

```
1 <div class="container">
2   <div class="item A1">A1</div>
3   <div class="item A2">A2</div>
4   <div class="item A3">A3</div>
5   <div class="item A4">A4</div>
6   <div class="item A5">A5</div>
7   <div class="item A6">A6</div>
8   <div class="item A7">A7</div>
9   <div class="item A8">A8</div>
10  <div class="item A9">A9</div>
11 </div>
```

CSS

```
29 background-color:#000460;
30 grid-row-start: 3;
31 grid-column-start: 3;
32 }
33 &.A6 {
34   background-color:#F94C10;
35 }
36 &.A7 {
37   background-color:#F94C10;
38   grid-area: 1 / 2 / 3 / 4;
39 }
40 &.A8 {
41   background-color:green;
42   grid-row: 3 / 4; /* grid-row-start: 3, grid-row-end: 4*/
43 }
```

A9

A7

A8

A1

A2

A3

A4

A6

A5

A blue arrow points from the CSS code's `grid-area: 1 / 2 / 3 / 4;` declaration to the A7 cell in the grid.

grid-column: 1 / span 3;

HTML

```
1 <div class="container">
2   <div class="item A1">A1</div>
3   <div class="item A2">A2</div>
4   <div class="item A3">A3</div>
5   <div class="item A4">A4</div>
6   <div class="item A5">A5</div>
7   <div class="item A6">A6</div>
8   <div class="item A7">A7</div>
9   <div class="item A8">A8</div>
10  <div class="item A9">A9</div>
11 </div>
```

CSS

```
36 background-color:#F94C10;
37 | grid-column: 1 / span 3;
38 }
39 &.A8 {
40 background-color:green;
41 | grid-row: 3 / 4; /* grid-row-start: 3, grid-row-end: 4*/
42 | grid-column: 3 / 4; /*grid-column-start: 3, grid-column-end: */
43 }
44 &.A9 {
45 background-color:#C70039;
46 }
```

A1

A2

A3

A4

A5

A6

A7

A8



grid-column: 1 / -1;

HTML

```
1 <div class="container">
2   <div class="item A1">A1</div>
3   <div class="item A2">A2</div>
4   <div class="item A3">A3</div>
5   <div class="item A4">A4</div>
6   <div class="item A5">A5</div>
7   <div class="item A6">A6</div>
8   <div class="item A7">A7</div>
9   <div class="item A8">A8</div>
10  <div class="item A9">A9</div>
11 </div>
```

CSS

```
22 }
23 &.A3 {
24   background-color:orange;
25 }
26 &.A4 {
27   background-color:#CE5A67;
28   grid-column: 1 / -1;
29 }
30 &.A5 {
31   background-color:#D6D46D;
32 }
33 &.A6 {
34   background-color:#F94C10;
35 }
```



`grid-auto-flow: row | column;`

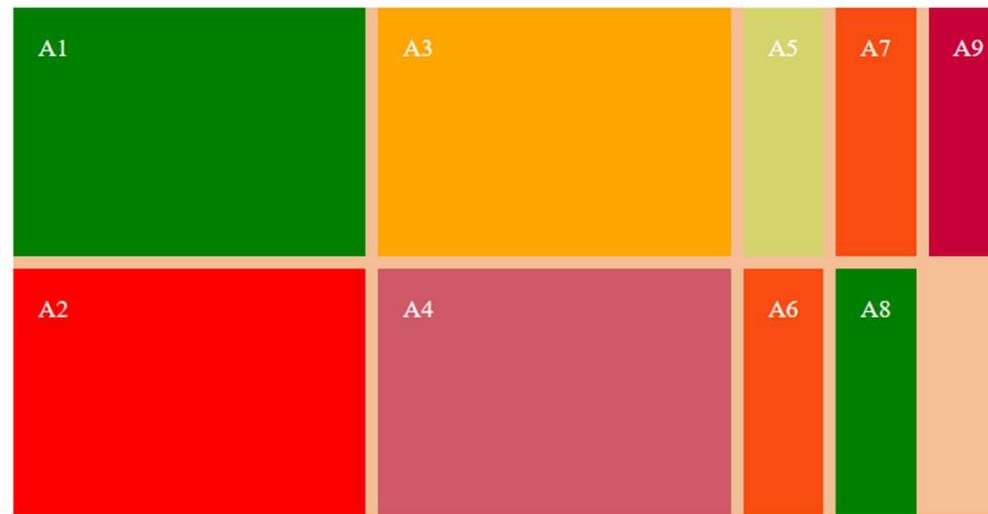
The image shows a code editor interface with two tabs: 'HTML' and 'CSS'. The 'HTML' tab is active, displaying the following code:

```
<div class="container">
  <div class="item A1">A1</div>
  <div class="item A2">A2</div>
  <div class="item A3">A3</div>
  <div class="item A4">A4</div>
  <div class="item A5">A5</div>
  <div class="item A6">A6</div>
  <div class="item A7">A7</div>
  <div class="item A8">A8</div>
  <div class="item A9">A9</div>
</div>
```

The 'CSS' tab is also active, displaying the following CSS code:

```
.container{
  width: 800px;
  background-color: #F4BF96;
  margin: 10px auto;
  display: grid;
  grid-template-columns: 1fr 1fr;
  grid-template-rows: repeat(2, 200px);
  gap:10px;
  grid-auto-rows: 50px;
  grid-auto-flow:column;
}

.item {
  padding: 20px;
```



align-items: center justify-items: center;

```
HTML
1 <div class="container">
2   <div class="item A1">A1</div>
3   <div class="item A2">A2</div>
4   <div class="item A3">A3</div>
5   <div class="item A4">A4</div>
6   <div class="item A5">A5</div>
7   <div class="item A6">A6</div>
8   <div class="item A7">A7</div>
9   <div class="item A8">A8</div>
10  <div class="item A9">A9</div>
11 </div>
```

```
CSS
10  grid-auto-flow:row;
11  /*alinha na vertical*/
12  align-items: center; /*default - stretch*/
13  /*alinha na horizontal*/
14  justify-items: center;
15 }
16
17 .item {
18   padding: 20px;
19   font-size: 20px;
20   font-family: sans-serif;
21   color: white;
22
23 &.A1 {
```



align-self: start;

```
HTML
1 <div class="container">
2   <div class="item A1">A1</div>
3   <div class="item A2">A2</div>
4   <div class="item A3">A3</div>
5   <div class="item A4">A4</div>
6   <div class="item A5">A5</div>
7   <div class="item A6">A6</div>
8   <div class="item A7">A7</div>
9   <div class="item A8">A8</div>
10  <div class="item A9">A9</div>
11 </div>
```

```
CSS
15 }
16
17 .item {
18   padding: 20px;
19   font-size: 20px;
20   font-family: sans-serif;
21   color: white;
22
23 &.A1 {
24   background-color: green;
25   grid-area: 1/1/3/2;
26   align-self: start;
27 }
28 &.A2 {
```

A1

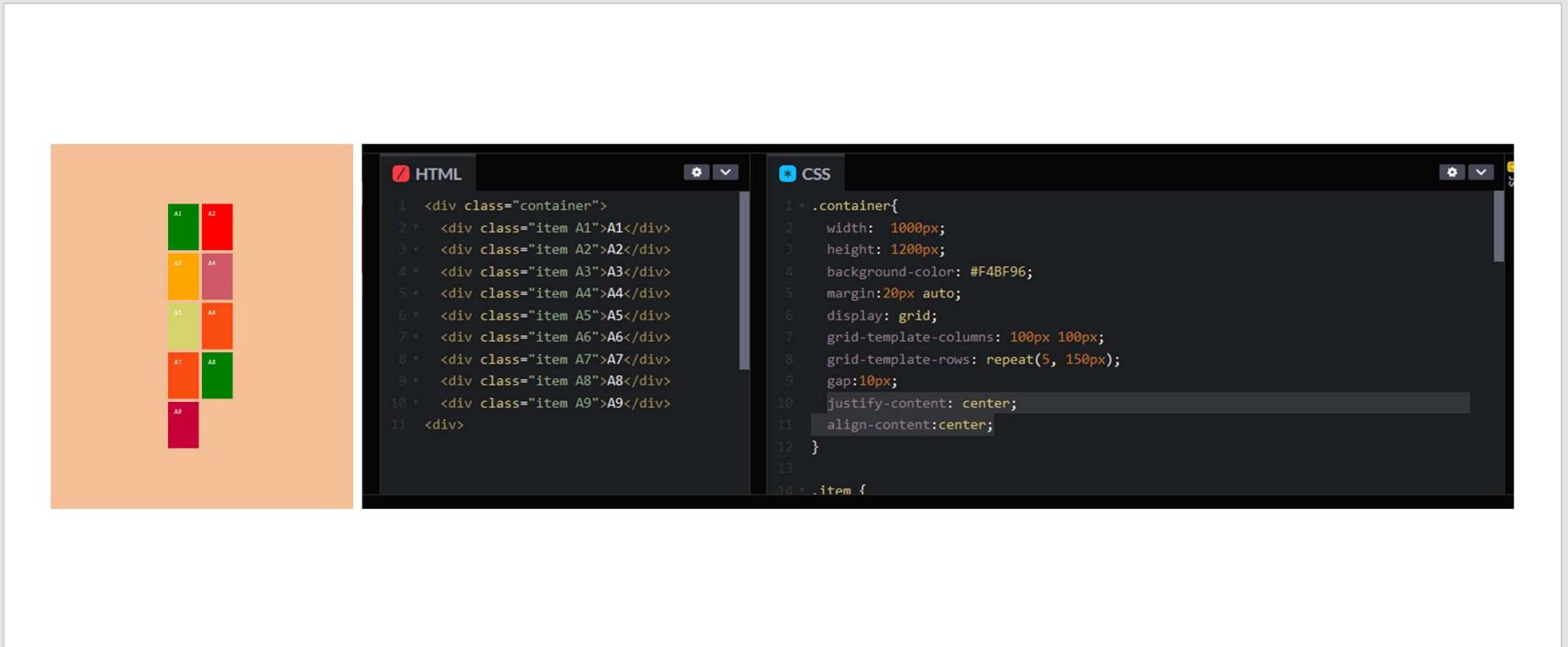
A2

A3

Exercício 9



justify-content: center; align-content: center;



The image shows a screenshot of a code editor with two tabs: HTML and CSS. The HTML tab contains a single-line doctype declaration and a script tag. The CSS tab contains the following code:

```
1 * .container{  
2   width: 1000px;  
3   height: 1200px;  
4   background-color: #F48F96;  
5   margin: 20px auto;  
6   display: grid;  
7   grid-template-columns: 100px 100px;  
8   grid-template-rows: repeat(5, 150px);  
9   gap: 10px;  
10  justify-content: center;  
11  align-content: center;  
12 }  
13  
14 * .item {
```

To the left of the code editor is a preview window showing a 2x5 grid of colored squares. The grid has 2 columns and 5 rows. The items are labeled A1 through A10. The first column contains A1 (green), A2 (red), A3 (yellow), A5 (light green), and A7 (orange). The second column contains A4 (pink), A6 (light orange), A8 (dark green), and A9 (dark red). The last row, A10, is empty.

max-content

HTML

```
1 <div class="container">
2   <div class="item A1">Residencia</div>
3   <div class="item A2">A2</div>
4   <div class="item A3">A3</div>
5   <div class="item A4">A uesc é
       uma universidade do sul da
       Bahia</div>
6   <div class="item A5">A5</div>
7   <div class="item A6">A6</div>
```

CSS

```
* { margin: 20px auto; }
* { display: grid;
  grid-template-columns: max-content 100px;
  grid-template-rows: repeat(5, 150px);
  gap: 10px;
  justify-content: center;
  align-content: center;
}
.item {
  padding: 20px;
```

Residencia	A2
A3	A uesc é uma universidade do sul da Bahia
A5	A6

min-content

HTML

```
1 <div class="container">
2   <div class="item A1">Residencia</div>
3   <div class="item A2">A2</div>
4   <div class="item A3">A3</div>
5   <div class="item A4">A uesc é
       uma universidade do sul da
       Bahia</div>
6   <div class="item A5">A5</div>
7   <div class="item A6">A6</div>
```

CSS

```
* margin: 20px auto;
* display: grid;
* grid-template-columns: max-content min-content;
* grid-template-rows: repeat(5, 150px);
* gap: 10px;
* justify-content: center;
* align-content: center;
}

.item {
  padding: 20px;
```

The image shows a 3x2 grid of colored boxes. The columns have different widths: the first column is wider than the second. The rows also vary in height. The boxes are colored: the top-left is green, the top-right is red, the middle-left is orange, the middle-right is maroon, the bottom-left is light green, and the bottom-right is orange-red. The text content of the boxes corresponds to the items listed in the HTML code above, demonstrating how the 'min-content' property affects the size of the grid items.

Residencia	A2
A3	A uesc é uma universidade do sul da Bahia
A5	A6

minmax

HTML

```
1 <div class="container">
2   <div class="item A1">Residencia</div>
3   <div class="item A2">A2</div>
4   <div class="item A3">A3</div>
5   <div class="item A4">A
  Universidade de Santa Cruz está
  localizada no sul da
  Bahia</div>
6   <div class="item A5">A5</div>
```

CSS

```
6   display: grid;
7   grid-template-columns: max-content min-content;
8   grid-template-rows: repeat(5, 150px);
9   gap: 10px;
10  justify-content: center;
11  align-content: center;
12 }
13 .item {
14   padding: 20px;
```



Residencia	A2
A3	A Universidade de Santa Cruz está localizada no sul da Bahia
A5	A6

minmax(200px, min-content)

HTML

```
1 <div class="container">
2   <div class="item A1">Residencia</div>
3   <div class="item A2">A2</div>
4   <div class="item A3">A3</div>
5   <div class="item A4">A
6 Universidade de Santa Cruz está
7 localizada no sul da
8 Bahia</div>
9   <div class="item A5">A5</div>
```

CSS

```
6   display: grid;
7   grid-template-columns: max-content min-content;
8   grid-template-rows: repeat(5, minmax(100px, min-content));
9   gap: 10px;
10  justify-content: center;
11  align-content: center;
12 }
13
14 .item {
15   padding: 20px;
```



Referências

- CSS Tutorial
 - <https://www.w3schools.com/css/default.asp>
- Aprenda a estilizar HTML utilizando CSS
 - <https://developer.mozilla.org/pt-BR/docs/Learn/CSS>
- CSS Grid Layout: A New Layout Module for the Web
 - <https://webkit.org/blog/7434/css-grid-layout-a-new-layout-module-for-the-web/>
- A Complete Guide to CSS Grid
 - <https://css-tricks.com/snippets/css/complete-guide-grid/>