

RESPONSIVIDADE

FAPESC – DESENVOLVEDORES PARA TECNOLOGIA DA INFORMAÇÃO

HERCULANO DE BIASI

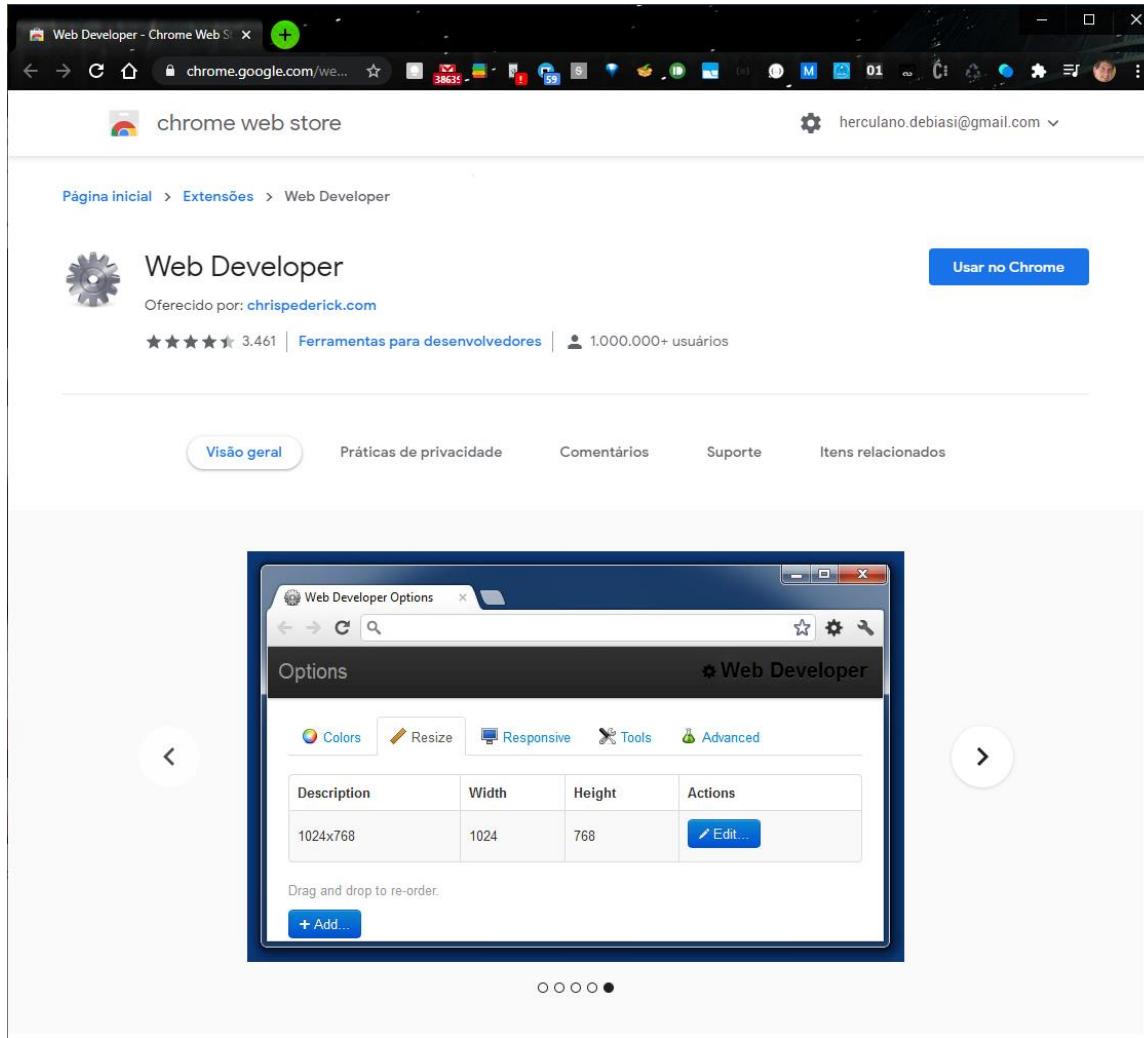
herculano.debiasi@unoesc.edu.br

TÓPICOS

- Extensões para navegadores
- Exemplos de sites responsivos
- Tipos de *design*
- Modelo de caixa (*box model*)
- Posicionamento
- **Tipografia**
- **Fontes**
- **Media types**
- **Media queries**
- Flexbox
- **Grid layout**

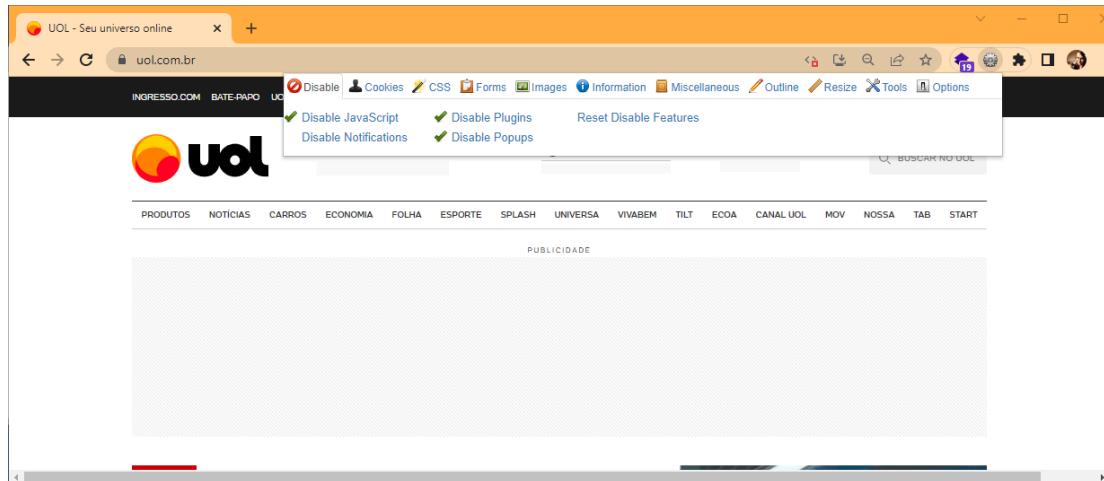
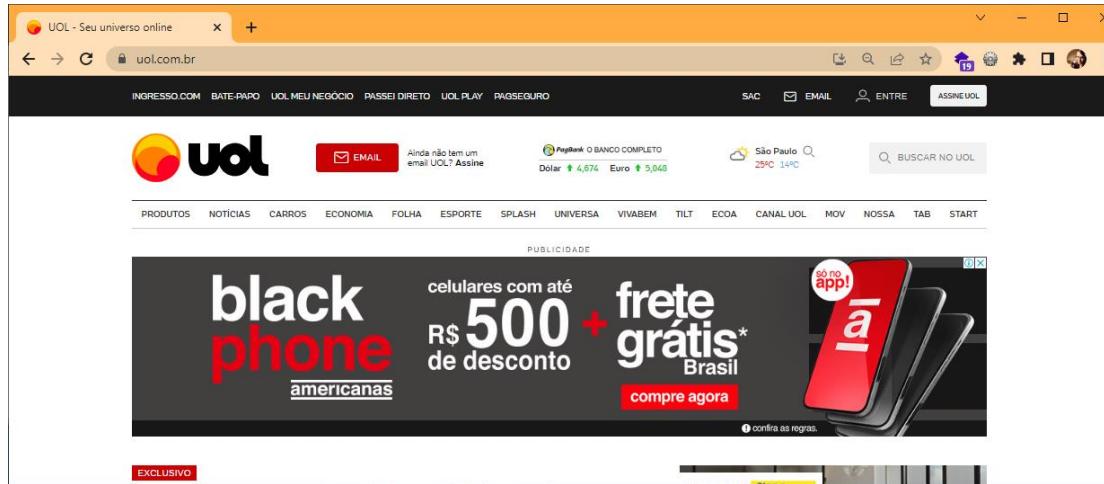
EXTENSÕES PARA NAVEGADORES

- A extensão Web Developer (Chrome/Firefox/Opera) faz várias análises



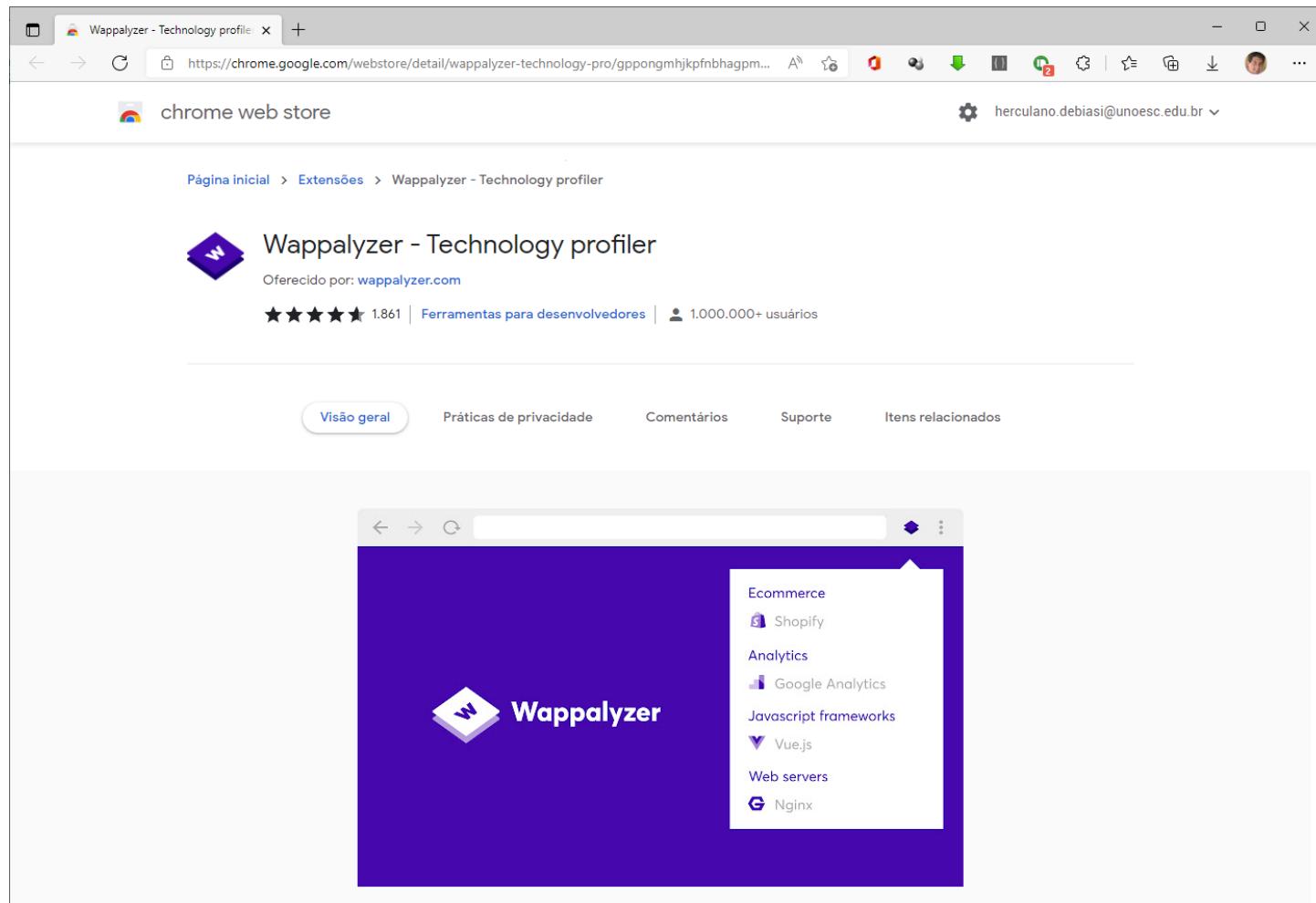
EXTENSÕES PARA NAVEGADORES

■ Desabilitando o JavaScript com a extensão Web Developer



EXTENSÕES PARA NAVEGADORES

- A extensão Wappalyzer indica as tecnologias utilizadas em sites



EXTENSÕES PARA NAVEGADORES

- A extensão Wappalyzer indica as tecnologias utilizadas em sites

The screenshot shows a web browser window with the URL acad.unoesc.edu.br/academic/mainMenu.jspa. The page is the main menu of the EDUCARE system for the UNOESC academic portal. The Wappalyzer extension is active, displaying a purple sidebar with the following information:

- TECHNOLOGIES**: Script de Fonte (Font Awesome), UI Frameworks (Bootstrap), Linguagem de Programação (Java).
- MORE INFO**: A link to "Something wrong or missing?".
- Generate sales leads**: A call-to-action button with the text "Find new prospects by the technologies they use. Reach out to customers of Shopify, Magento, Salesforce and others." and a "Create a lead list →" button.

EXTENSÕES PARA NAVEGADORES

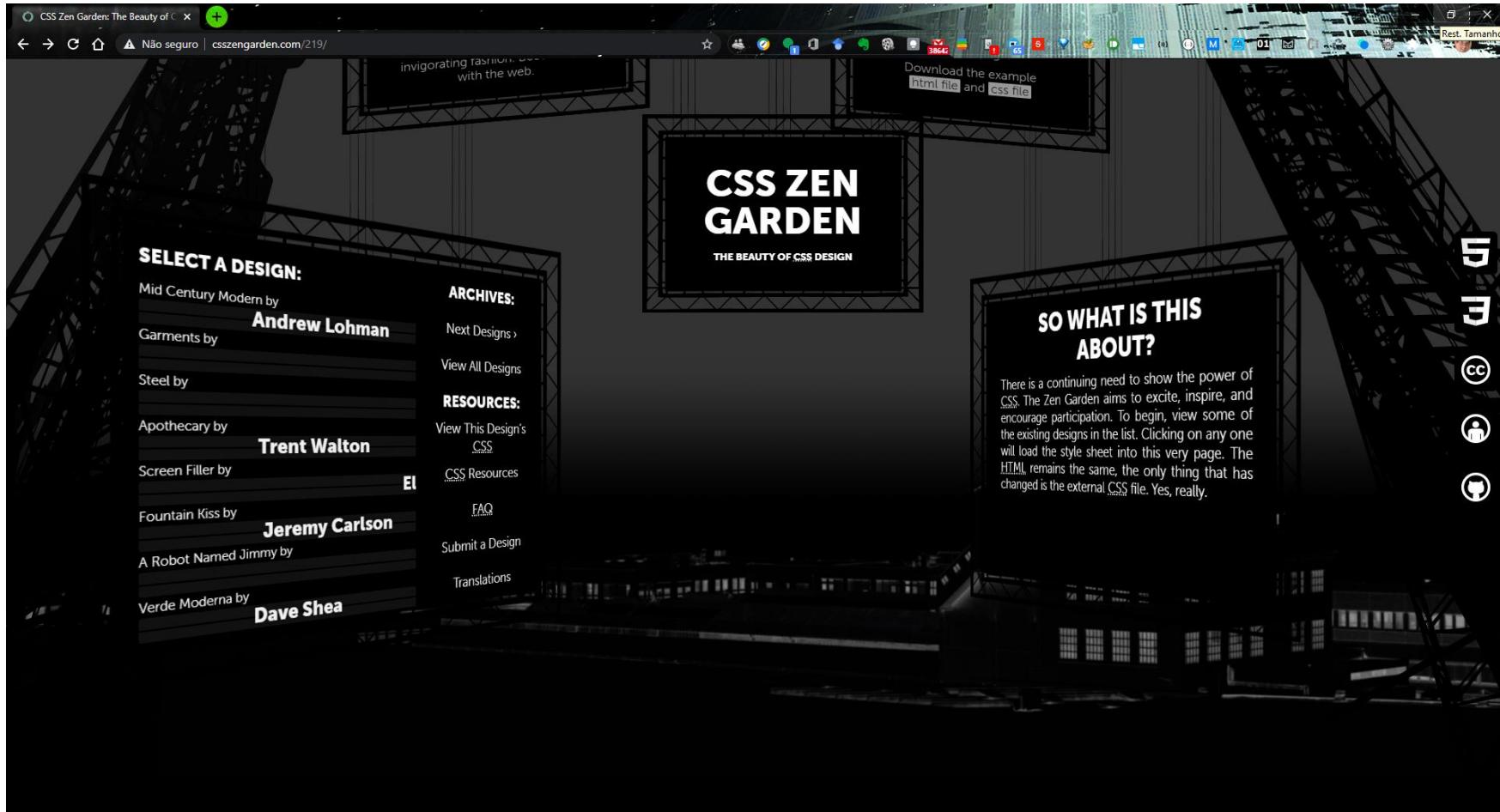
- A extensão Wappalyzer indica as tecnologias utilizadas em sites

The screenshot shows a web browser window with the Unoesc website (unoesc.edu.br) loaded. The Wappalyzer extension is active, overlaying the page with a sidebar containing a comprehensive list of technologies detected on the site. The sidebar includes sections for CMS (ExpressionEngine), Linguagem de Programação (PHP), Widget (Facebook), Automação de Marketing (HubSpot), Ferramenta Estatística (Matomo Analytics, Google Ads Conversion Tracking, HubSpot Analytics, Google Analytics), Gestor de Tags (Google Tag Manager), Chat Direto (WhatsApp Business Chat), Biblioteca JavaScript (core-js, jQuery), Reprodutor de Vídeo (YouTube), Script de Fonte, and Cookie compliance (HubSpot Cookie Policy Banner). The main content of the website features the Unoesc logo, a search bar, navigation menus for 'Vestibular e Seletivo' (with a 100% scholarship offer), 'Notícias por campus', and various course sections like 'Graduação', 'Graduação EAD', 'Especialização', and 'Mestrado e Doutorado'.

Tecnologia	Descrição
CMS	ExpressionEngine
Linguagem de Programação	PHP
Widget	Facebook
Automação de Marketing	HubSpot
Ferramenta Estatística	Matomo Analytics, Google Ads Conversion Tracking, HubSpot Analytics, Google Analytics
Gestor de Tags	Google Tag Manager
Chat Direto	WhatsApp Business Chat
Biblioteca JavaScript	core-js, jQuery
Reprodutor de Vídeo	YouTube
Script de Fonte	
Cookie compliance	HubSpot Cookie Policy Banner

EXTENSÕES PARA NAVEGADORES

■ CSS Zen Garden com CSS



EXTENSÕES PARA NAVEGADORES

■ CSS Zen Garden sem CSS



CSS Zen Garden

The Beauty of CSS Design

A demonstration of what can be accomplished through CSS-based design. Select any style sheet from the list to load it into this page.

Download the example [html file](#) and [css file](#)

The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, broken CSS support, and abandoned browsers.

We must clear the mind of the past. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP, and the major browser creators.

The CSS Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the time-honored techniques in new and invigorating fashion. Become one with the web.

So What is This About?

There is a continuing need to show the power of CSS. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The HTML remains the same, the only thing that has changed is the external CSS file. Yes, really.

CSS allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets people excited is by demonstrating what it can truly be, once the reins are placed in the hands of those able to create beauty from structure. Designers and coders alike have contributed to the beauty of the web; we can always push it further.

Participation

Strong visual design has always been our focus. You are modifying this page, so strong CSS skills are necessary too, but the example files are commented well enough that even CSS novices can use them as starting points. Please see the [CSS Resource Guide](#) for advanced tutorials and tips on working with CSS.

You may modify the style sheet in any way you wish, but not the HTML. This may seem daunting at first if you've never worked this way before, but follow the listed links to learn more, and use the sample files as a guide.

Download the sample [HTML](#) and [CSS](#) to work on a copy locally. Once you have completed your masterpiece (and please, don't submit half-finished work) upload your CSS file to a web server under your control. [Send us a link](#) to an archive of that file and all associated assets, and if we choose to use it we will download it and place it on our server.

Benefits

Why participate? For recognition, inspiration, and a resource we can all refer to showing people how amazing CSS really can be. This site serves as equal parts inspiration for those working on the web today, learning tool for those who will be tomorrow, and gallery of future techniques we can all look forward to.

Requirements

Where possible, we would like to see mostly CSS 1.&2 usage. CSS 3.&4 should be limited to widely-supported elements only, or strong fallbacks should be provided. The CSS Zen Garden is about functional, practical CSS and not the latest bleeding-edge tricks viewable by 2% of the browsing public. The only real requirement we have is that your CSS validates.

Luckily, designing this way shows how well various browsers have implemented CSS by now. When sticking to the guidelines you should see fairly consistent results across most modern browsers. Due to the sheer number of user agents on the web these days — especially when you factor in mobile — pixel-perfect layouts may not be possible across every platform. That's okay, but do test in as many as you can. Your design should work in at least IE9+ and the latest Chrome, Firefox, iOS and Android browsers (run by over 90% of the population).

We ask that you submit original artwork. Please respect copyright laws. Please keep objectionable material to a minimum, and try to incorporate unique and interesting visual themes to your work. We're well past the point of needing another garden-related design.

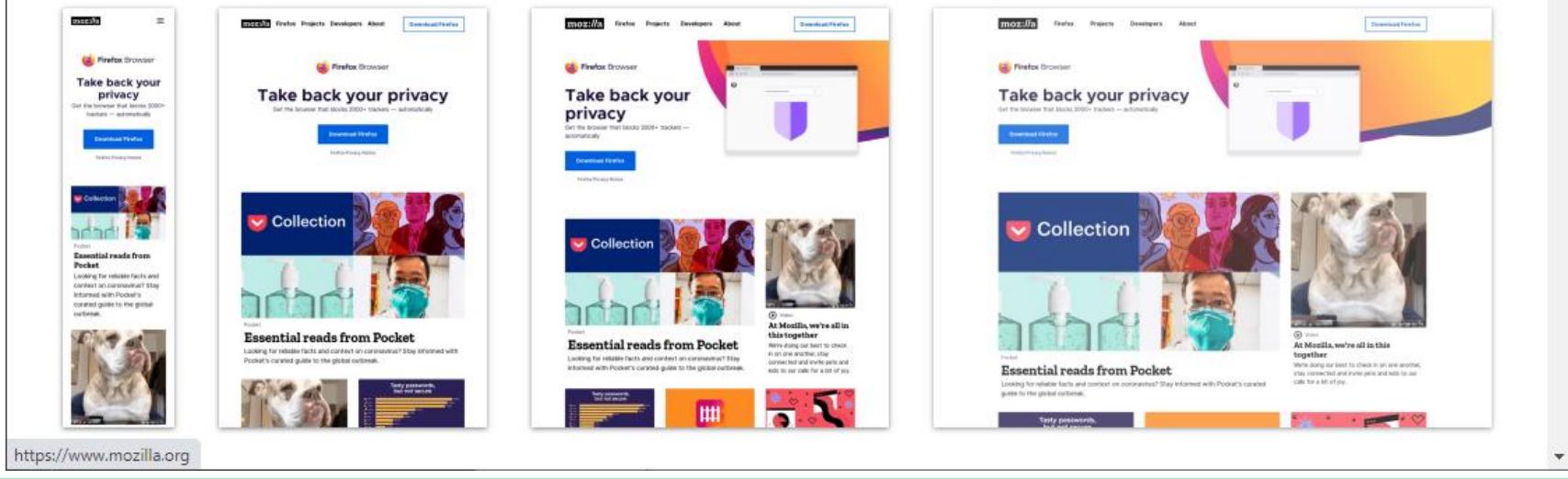
This is a learning exercise as well as a demonstration. You retain full copyright on your graphics (with limited exceptions, see [submission guidelines](#)), but we ask you release your CSS under a Creative Commons license identical to the [one on this site](#) so that others may learn from your work.

EXEMPLOS DE SITES RESPONSIVOS/ADAPTATIVOS

■ Site Media Queries



Mozilla



EXEMPLOS DE SITES RESPONSIVOS/ADAPTATIVOS

■ finans

The screenshot displays the homepage of the Finans website. The top navigation bar includes links for Home, Recursos, Benefícios, Preço, and Entrar. A large yellow banner features the text "Suas contas, descomplicadas" and a subtext stating that the service is used by over 1 million people. It also highlights that Finans is a secure online tool. Below the banner is a search bar labeled "Seu e-mail" and a blue "Cadastre-se" button. To the right of the search bar, there's a placeholder for device compatibility with icons for smartphone and tablet. The main content area shows a woman smiling and holding a gold coin, with a bar chart graphic overlaid on the background.

Suas contas,
descomplicadas

Usado por mais de 1 milhão de pessoas, o Finans é uma ferramenta online que vai facilitar sua vida financeira.

Seu e-mail

Cadastre-se

Disponível para

Saiba para onde vai seu dinheiro

Com o Finans, você categoriza todos os seus lançamentos. Com gráficos simples, você sabe de onde vem e para onde vai o seu dinheiro.

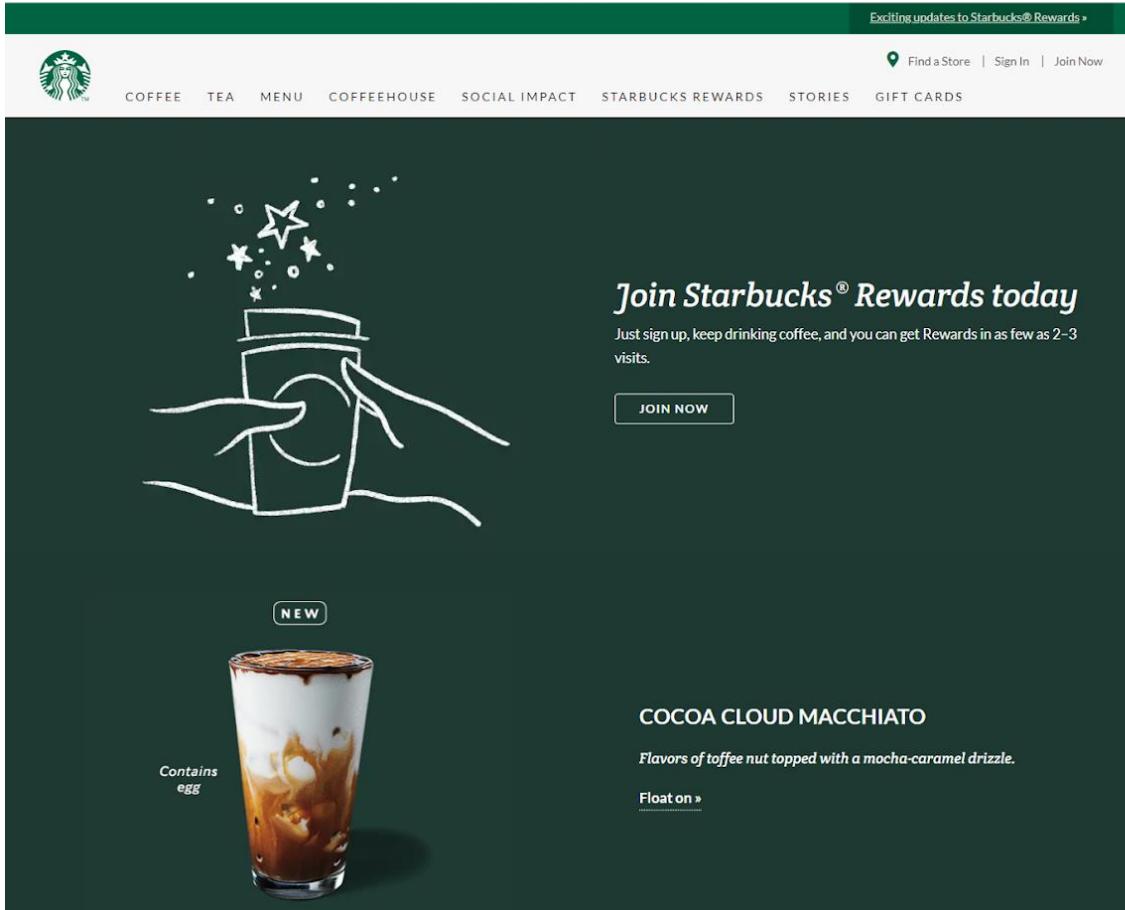
Veja mais

A small graphic of a mobile device screen displaying a bar chart and a pie chart, representing financial data visualization.

EXEMPLOS DE SITES RESPONSIVOS/ADAPTATIVOS

■ Starbucks

DESKTOP VIEW



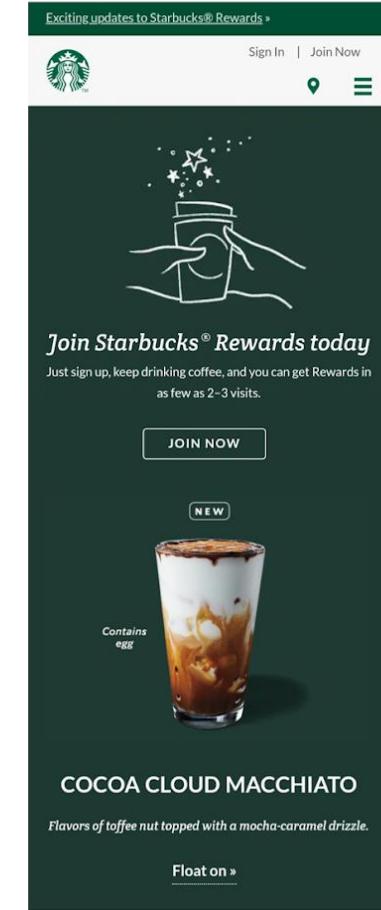
The Starbucks desktop website features a dark green header with the Starbucks logo and navigation links: COFFEE, TEA, MENU, COFFEEHOUSE, SOCIAL IMPACT, STARBUCKS REWARDS, STORIES, and GIFT CARDS. A prominent call-to-action button says "JOIN NOW". Below the main image, there's a "NEW" badge next to a Cocoa Cloud Macchiato image.

Join Starbucks® Rewards today
Just sign up, keep drinking coffee, and you can get Rewards in as few as 2-3 visits.

JOIN NOW

COCOA CLOUD MACCHIATO
Flavors of toffee nut topped with a mocha-caramel drizzle.
[Float on »](#)

MOBILE VIEW



The Starbucks mobile website has a similar layout to the desktop version but is optimized for a smaller screen. It includes a header with the Starbucks logo and links for Find a Store, Sign In, and Join Now. The main content area features a "JOIN NOW" button and a "NEW" badge next to a Cocoa Cloud Macchiato image.

Join Starbucks® Rewards today
Just sign up, keep drinking coffee, and you can get Rewards in as few as 2-3 visits.

JOIN NOW

COCOA CLOUD MACCHIATO
Flavors of toffee nut topped with a mocha-caramel drizzle.
[Float on »](#)

EXEMPLOS DE SITES RESPONSIVOS/ADAPTATIVOS

■ Adidas

The screenshot shows the Adidas website for the Al Rihla Pack. The background features a repeating yellow hexagonal pattern with small white stars inside. In the center, there's a large circular frame containing a soccer boot. To the right of the frame is a smaller rectangular frame showing hands holding soccer boots. At the top, the Adidas logo is on the left, followed by navigation links: MEN, WOMEN, KIDS, SPORTS, BRANDS, COLLECTIONS, and OUTLET. On the far right are links for help, exchanges & returns, order tracker, become a member, adiclub, log in, and a flag icon. Below the navigation is a search bar with a magnifying glass icon and user icons for profile, heart, and shopping bag. A black banner at the top says "FREE RETURNS & EXCHANGE". On the left, there's a "SHOP NOW →" button. In the bottom left corner, there are video control icons for pausing and volume.

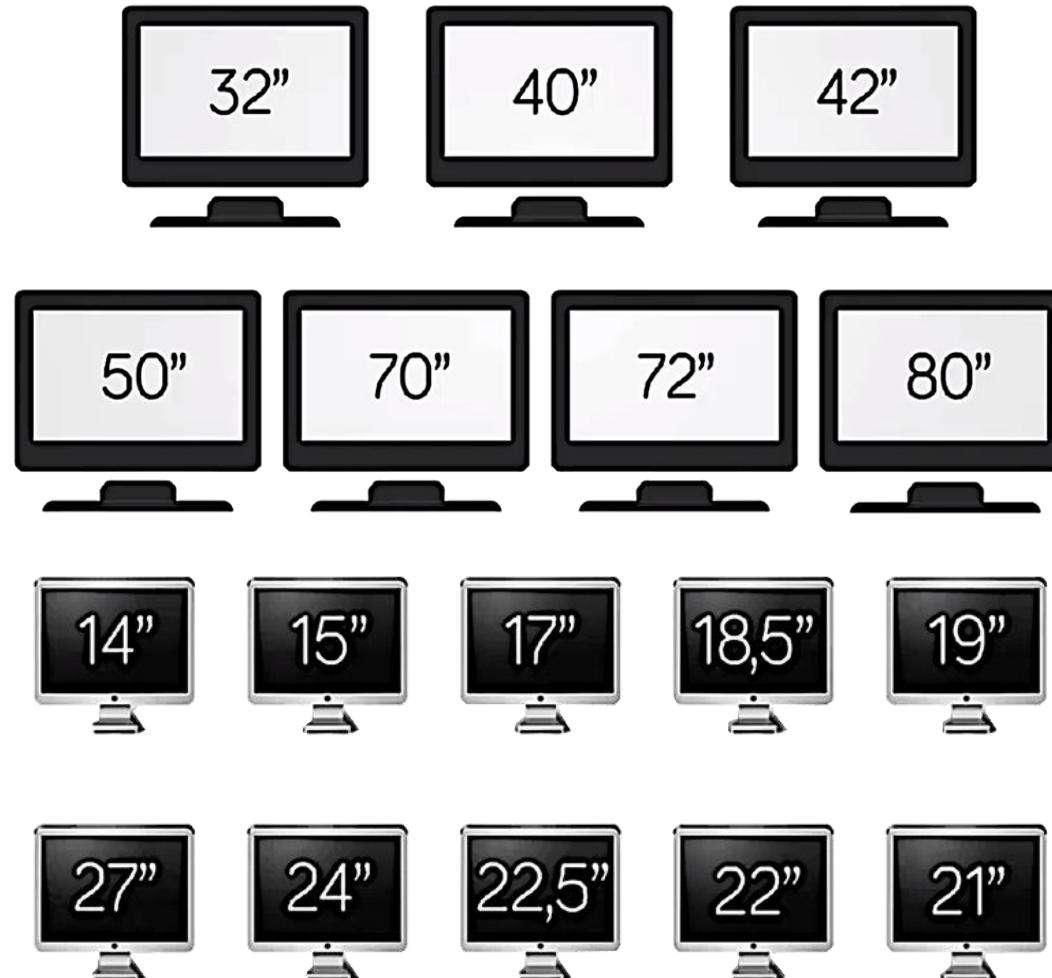
EXEMPLOS DE SITES RESPONSIVOS/ADAPTATIVOS

Nasa

The screenshot displays the official NASA website (nasa.gov) in a browser window. The top navigation bar includes links for Missions, Galleries, NASA TV, Follow NASA, Downloads, About, and NASA Audiences, along with a search bar. Below this, a secondary navigation bar offers categories like Humans in Space, Moon to Mars, Earth, Space Tech, Flight, Solar System and Beyond, STEM Engagement, History, and Benefits to You. The main content area features a large image of a rocket launching, with a caption below it: "Commercial Crew" and "NASA's SpaceX Crew-5 Launches to International Space Station". To the right, there's a "NASA Explorers" section titled "ARTEMIS GENERATION" with an article about astronauts thinking like geologists. Further down, a "Space Tech" section shows a large, circular heat shield, with a related news item: "NASA Invites Media to Learn About Inflatable Heat Shield Demo". On the far right, a sidebar lists "NASA Events" for October 5 and 6, 2023, including coverage of the SpaceX Crew-5 mission and a science leadership town hall. At the bottom, links for "NASA TV Schedule" and "Launches and Landings" are visible.

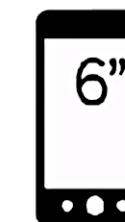
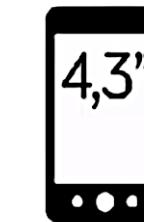
TIPOS DE DESIGN

- Diferentes tamanhos de dispositivos



TIPOS DE DESIGN

- Diferentes tamanhos de dispositivos



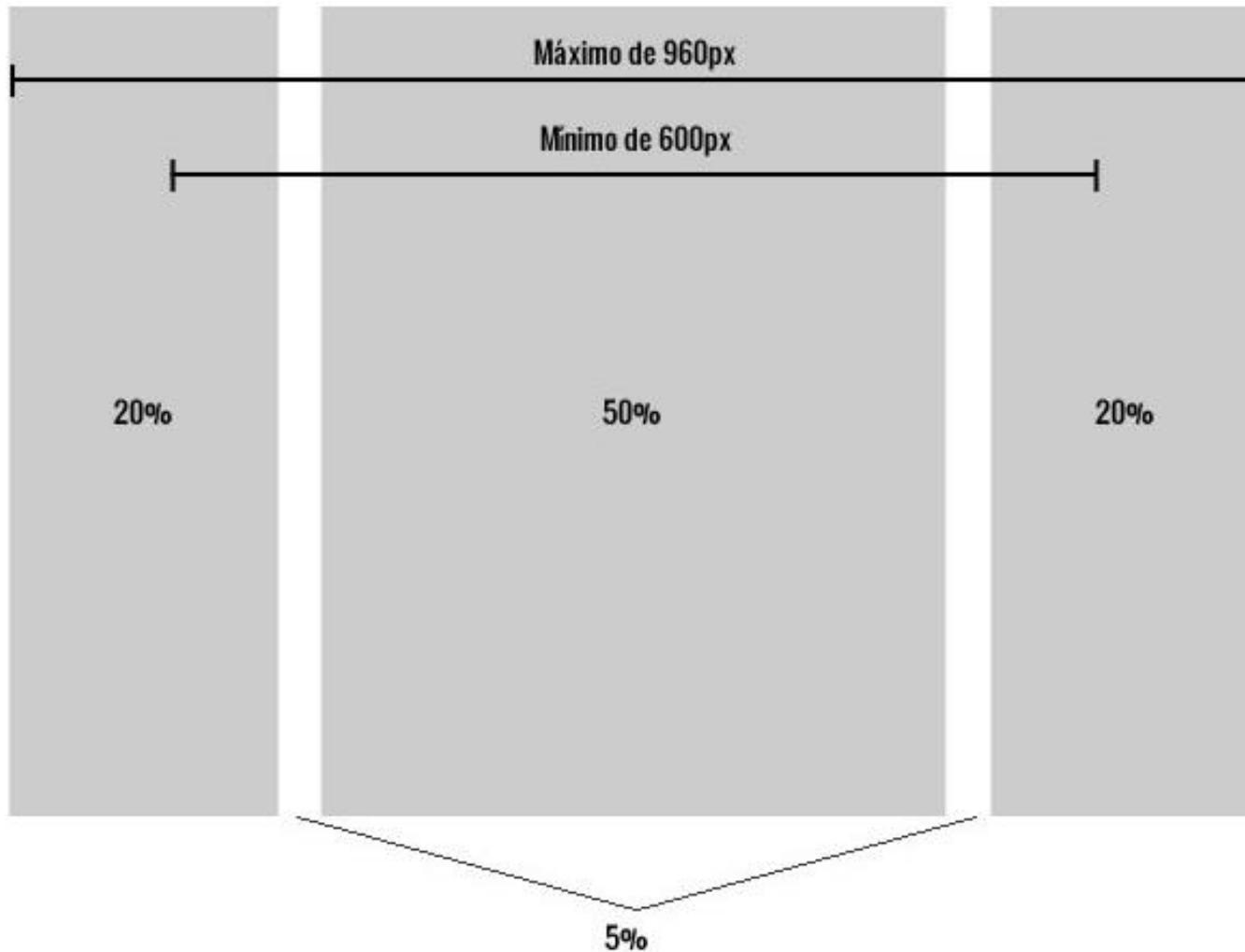
TIPOS DE DESIGN

■ Design fluido



TIPOS DE DESIGN

■ Design fluido



TIPOS DE DESIGN

■ Design fluido: Definindo tamanhos máximos e mínimos dos elementos

A screenshot of a browser window titled "Tabela". The address bar shows the URL "127.0.0.1:5500/tabelas1....". The page content is a table with the following data:

Código	Produto	Preço
001	Notebook	R\$ 4.999,99
002	Tablet	R\$ 10.999,99
Total		R\$ 15.999,98

```
1  <!DOCTYPE html>
2  <html lang="pt-BR">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Tabela</title>
8
9      <style>
10         table, th, td {
11             border: 2px solid black;
12             border-collapse: collapse;
13             padding: 2px;
14             margin: 0 auto;
15             min-width: 150px;
16             max-width: 800px;
17         }
18     </style>
19 </head>
20 <body>
21     <table style="width: 80%;">
22         <caption>Lista de Produtos</caption>
23         <thead>
24             <tr>
25                 <th width="20%">Código</th>
26                 <th width="50%">Produto</th>
27                 <th width="30%">Preço</th>
28             </tr>

```

A screenshot of a browser window titled "Tabela". The address bar shows the URL "127.0.0.1:5500/tabelas1.html". The page content is a table with the following data:

Código	Produto	Preço
001	Notebook	R\$ 4.999,99
002	Tablet	R\$ 10.999,99
Total		R\$ 15.999,98

```
43             <tr>
44                 <td colspan="2" align="center">Total</td>
45                 <td align="right">R$ 15.999,98</td>
46             </tr>
47         </tfoot>
48     </table>
49 </body>
50 </html>
```

TIPOS DE DESIGN

■ Web design responsivo



TIPOS DE DESIGN

- *Mobile first:* Conceito no qual se faz a criação de projetos web e sites primeiro em dispositivos móveis, para depois fazer adaptações para o *desktop* e outras plataformas, ao contrário do que ainda acontece na maioria das vezes
- *Design adaptativo:* Criar uma versão “totalmente” diferente, com linhas de código, imagens, e estrutura feitas somente para dispositivos móveis

Desktop first



Mobile first



TIPOS DE DESIGN

■ Tag meta com atributo viewport

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

- Atributo width=device-width define o tamanho da tela do dispositivo
- Atributo initial-scale=1.0 define o nível de zoom inicial da página ao ser carregada

Sem a tag meta com atributo viewport



Com a tag meta com atributo viewport



Placeholder text: "Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet domino..."

MODELO DE CAIXA (Box Model)

- Os estilos dos elementos HTML estão estruturados dentro de uma caixa chamada de “Box Model”, sendo que dentro dessa mesma caixa existe a hierarquia

Margin Box

Border Box

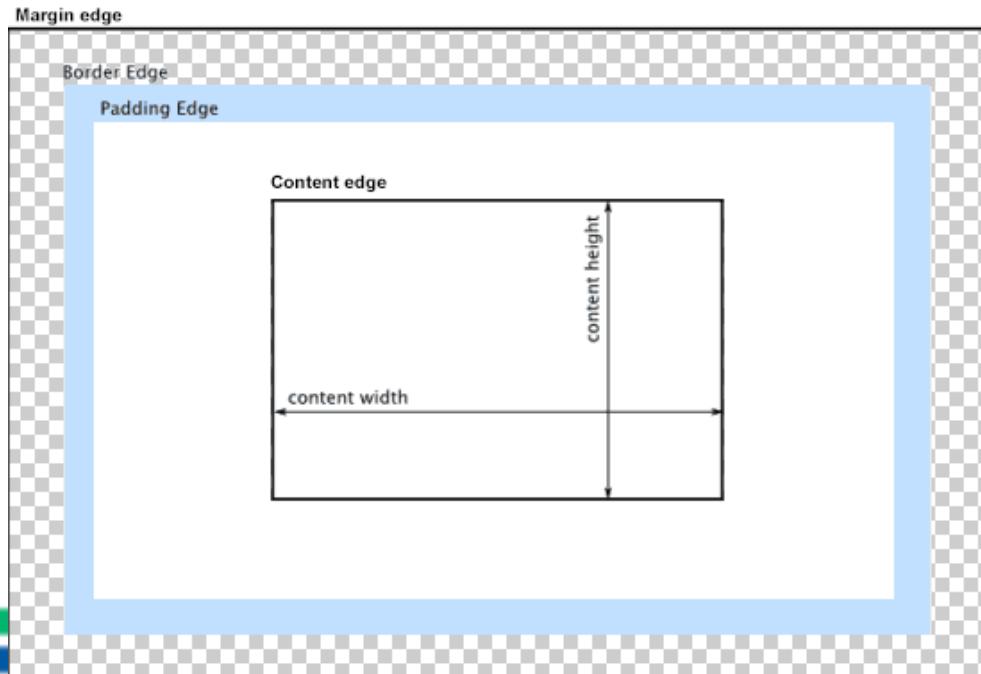
Padding Box

Element Box (O elemento `em si`, `div`, `span`, entre outros)

- O *padding* é contabilizado no tamanho total do elemento

MODELO DE CAIXA (Box Model)

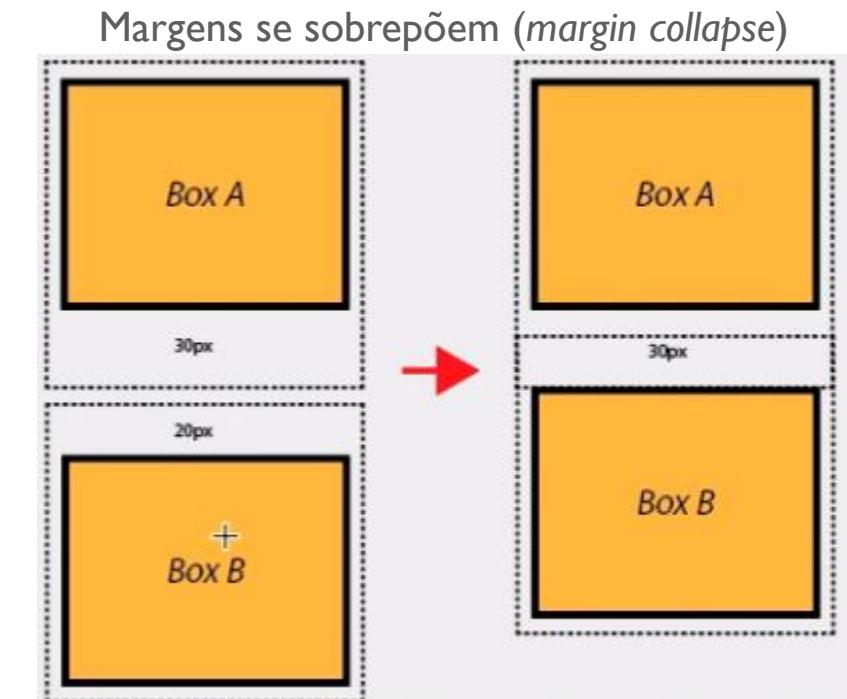
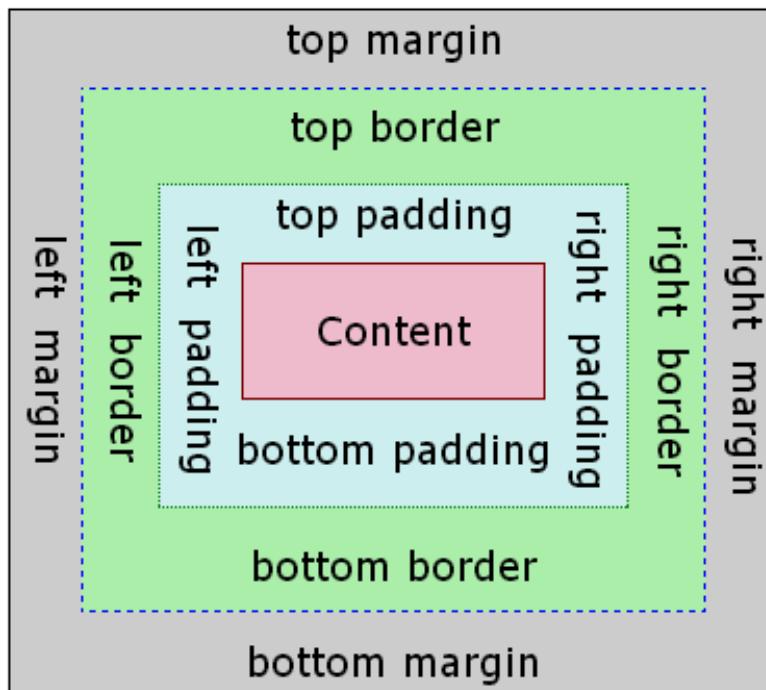
- Cada caixa possui 4 edges (bordas)
 - Margin edge: A área de margem (*margin area*) estende a área de borda com um espaço vazio utilizado para separar o elemento dos elementos vizinhos
 - Border edge: A área de borda (*border area*) estende a área de preenchimento para a área que contém as bordas
 - Padding edge: A área de preenchimento (*padding area*) estende-se para a borda em torno do enchimento; o preenchimento está localizado dentro do *padding edge*, e suas dimensões são a largura do *padding-box* e a altura do *padding-box*
 - Content edge: A área de conteúdo (*content area*) é a área ocupada pelo conteúdo real do elemento e é localizada dentro do *content edge*



MODELO DE CAIXA (Box MODEL)

■ Diferença entre *margin* e *padding*

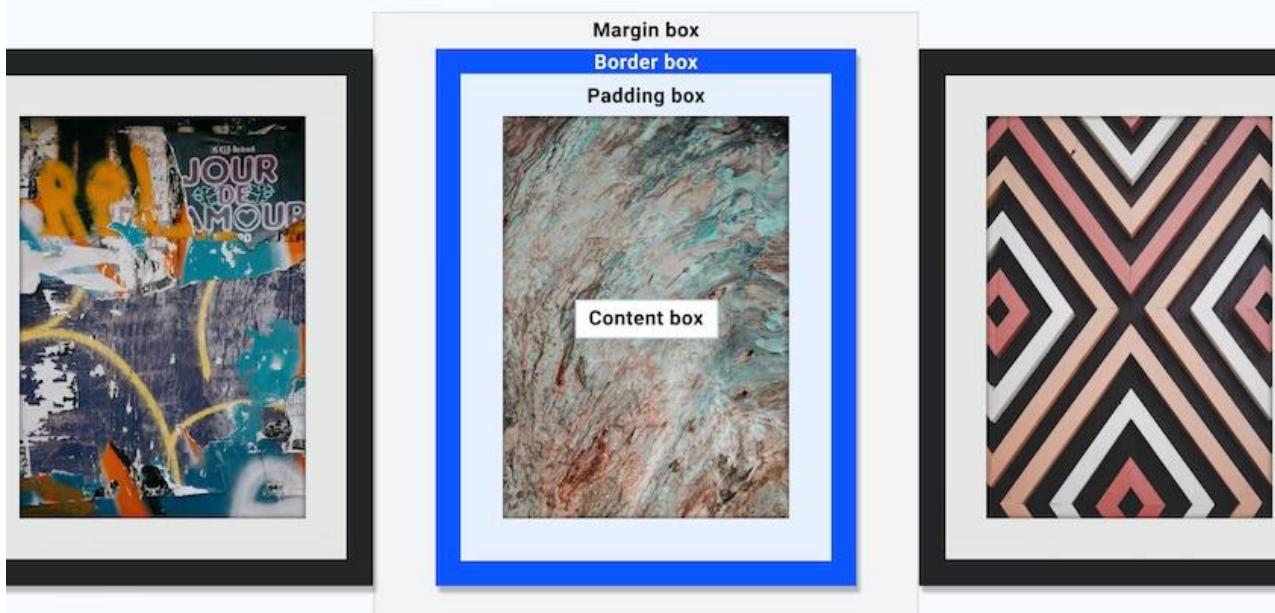
- *Margin* (margem): Espaçamento fora do conteúdo
- *Padding* (preenchimento): Espaçamento dentro dos limites do conteúdo



MODELO DE CAIXA (Box Model)

■ Analogia com um quadro

- A caixa de conteúdo é a obra de arte
- A caixa de preenchimento (*padding*) é o fosco branco, entre a moldura e a arte
- A caixa de borda é a moldura, fornecendo uma borda literal para a obra de arte
- A caixa de margem é o espaço entre cada quadro
- A sombra ocupa o mesmo espaço que a caixa de margem



MODELO DE CAIXA (Box Model)

■ Exemplo

```
aula4 - box_model1.html

1 <!DOCTYPE html>
2 <html lang="pt-BR">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta http-equiv="X-UA-Compatible" content="IE=edge">
7     <meta name="viewport" content="width=device-width, initial-scale=1.0">
8     <title>Box Model</title>
9
10    <style>
11        .conteudo1 {
12            /* display: inline; */
13            border: 1px solid red;
14            width: 300px;
15
16            /* padding-top: 20px;
17            padding-right: 20px;
18            padding-left: 20px;
19            padding-bottom: 20px; */
20            padding: 20px;
21            padding: 20px 5px;
22            padding: 20px 20px 20px 20px;
23
24            /* margin-top: 20px;
25            margin-right: 20px;
26            margin-bottom: 20px;
27            margin-left: 20px; */
28            margin: 20px
29        }
30    </style>
31 </head>
32
33 <body>
34     <div class="conteudo1">
35         Lorem ipsum, dolor sit amet consectetur adipisicing elit. Quas rem placeat, veritatis obcaecati, laudantium voluptate
36         modi temporibus ducimus maxime, aspernatur odio veniam molestiae nihil beatae. Possimus officia esse enim cumque!
37     </div>
38     <div class="conteudo1">
39         Lorem ipsum dolor sit amet consectetur adipisicing elit. Modi, odit! Omnis quibusdam perferendis eius ab! Officiis in
40         ventore dolorem repellendus blanditiis! Unde laboriosam, rem magni corporis quisquam eius laborum dolorum adipisci?
41     </div>
42     Lorem ipsum dolor sit amet consectetur adipisicing elit. Unde et, aliquam harum, est consectetur alias, fugiat eveniet fa
43         cilis corporis recusandae excepturi perspiciatis. Placeat tempore a rerum voluptatum aspernatur consequuntur perferendis.
44 </body>
45 </html>
```

POSICIONAMENTO

- Estão disponíveis 4 diferentes tipos de posicionamento
 - Estático (static)
 - Relativo (relative)
 - Absoluto (absolute)
 - Fixo (fixed)

POSICIONAMENTO

■ Exemplo dos posicionamentos estático e relativo

```
● ● ● aula4 - posicionamento1.html
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>Posicionamento estático</title>
9
10 <style>
11   body {
12     background: #007fc7;
13   }
14
15   #container {
16     background: white;
17     width: 500px;
18     padding: 5px 15px 1px;
19     margin: 10px auto;
20   }
21
22   h1 {
23     position: static;
24     margin: 0px;
25   }
26
27   p {
28     position: static;
29   }
30 </style>
31 </head>
32
33 <body>
34   <div id="container">
35     <h1>Posicionamentos</h1>
36     <p>P1: Lorem ipsum dolor sit amet consectetur, adipisicing elit. Rerum harum sint enim laborum iusto itaque qui  
usui impedit blanditiis magni quia. Vero doloribus optio quisquam deserunt persiciatis! Facilis eius modi aut.</p>
37
38     <p>P2: Lorem ipsum dolor sit amet consectetur, adipisicing elit. Rerum harum sint enim laborum iusto itaque qui  
usui impedit blanditiis magni quia. Vero doloribus optio quisquam deserunt persiciatis! Facilis eius modi aut.</p>
39
40     <p>P: Lorem ipsum dolor sit amet consectetur, adipisicing elit. Rerum harum sint enim laborum iusto itaque qui  
asi impedit blanditiis magni quia. Vero doloribus optio quisquam deserunt persiciatis! Facilis eius modi aut.</p>
41   </div>
42 </body>
43 </html>
```

```
● ● ● aula4 - posicionamento2.html
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>Posicionamento relativo</title>
9
10 <style>
11   body {
12     background: #007fc7;
13   }
14
15   #container {
16     background: white;
17     width: 500px;
18     padding: 5px 15px 1px;
19     margin: 10px auto;
20   }
21
22   h1 {
23     position: static;
24     margin: 0px;
25   }
26
27   p {
28     position: static;
29     width: 200px;
30   }
31
32   .relativo {
33     border: 2px solid red;
34     position: relative;
35     top: 10px;
36     bottom: 10px;
37     left: 100px;
38     right: 10px;
39   }
40 </style>
41 </head>
42
43 <body>
44   <div id="container">
45     <h1>Posicionamentos</h1>
46     <p>P1: Lorem ipsum dolor sit amet consectetur, adipisicing elit. Rerum harum sint enim laborum iusto itaque qui  
usui impedit blanditiis magni quia. Vero doloribus optio quisquam deserunt persiciatis! Facilis eius modi aut.</p>
47
48     <p class="relativo">P2: Lorem ipsum dolor sit amet consectetur, adipisicing elit. Rerum harum sint enim labor  
um iusto itaque quasi impedit blanditiis magni quia. Vero doloribus optio quisquam deserunt persiciatis! Facilis ei  
us modi aut.</p>
49
50     <p>P: Lorem ipsum dolor sit amet consectetur, adipisicing elit. Rerum harum sint enim laborum iusto itaque qui  
asi impedit blanditiis magni quia. Vero doloribus optio quisquam deserunt persiciatis! Facilis eius modi aut.</p>
51   </div>
52 </body>
53
54 </html>
```

POSICIONAMENTO

■ Exemplo do posicionamento absoluto

```
● ● ●
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta http-equiv="X-UA-Compatible" content="IE=edge">
7     <meta name="viewport" content="width=device-width, initial-scale=1.0">
8     <title>Posicionamento absoluto</title>
9
10    <style>
11        body {
12            background: #007fc7;
13        }
14
15        #container {
16            background: white;
17            width: 500px;
18            padding: 5px 15px 1px;
19            margin: 10px auto;
20        }
21
22        h1 {
23            margin: 0px;
24        }
25
26        p {
27            width: 200px;
28        }
29
30        .absoluto {
31            border: 2px solid red;
32            background: #cdcdcd;
33            position: absolute;
34            top: 150px;
35            left: 150px;
36        }
37    </style>
38 </head>
39
40 <body>
41     <div id="container">
42         <h1>Posicionamentos</h1>
43         <p>P1: Lorem ipsum dolor sit amet consectetur, adipisicing elit. Rerum harum sint enim laborum iusto itaque quasi impedit blanditiis magni quia. Vero doloribus optio quisquam deserunt perspiciatis! Facilis eius modi aut.</p>
44
45         <p class="absoluto">P2: Lorem ipsum dolor sit amet consectetur, adipisicing elit. Rerum harum sint enim laborum iusto itaque quasi impedit blanditiis magni quia. Vero doloribus optio quisquam deserunt perspiciatis! Facilis eius modi aut.</p>
46
47         <p>P: Lorem ipsum dolor sit amet consectetur, adipisicing elit. Rerum harum sint enim laborum iusto itaque quasi impedit blanditiis magni quia. Vero doloribus optio quisquam deserunt perspiciatis! Facilis eius modi aut.</p>
48     </div>
49 </body>
50
51 </html>
```

POSICIONAMENTO

■ Exemplo do posicionamento fixo

```
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>Posicionamento fixo</title>
9
10  <style>
11    body {
12      background: #007fc7;
13    }
14
15    #container {
16      background: white;
17      width: 500px;
18      padding: 5px 15px 1px;
19      margin: 10px auto;
20    }
21
22    h1 {
23      margin: 0px;
24    }
25
26    p {
27      width: 200px;
28    }
29
30    .fixo {
31      border: 2px solid red;
32      background: #cccccc;
33      position: fixed;
34      top: 150px;
35      left: 300px;
36    }
37  </style>
38 </head>
39
40 <body>
41   <div id="container">
42     <h1>Posicionamentos</h1>
43     <p>P1: Lorem ipsum dolor sit amet consectetur, adipisicing elit. Rerum harum sint enim laborum iusto itaque quasi impedit blanditiis magni quia. Vero doloribus optio quisquam deserunt perspiciatis! Facilis eius modi aut.</p>
44
45     <p class="fixo">P2: Lorem ipsum dolor sit amet consectetur, adipisicing elit. Rerum harum sint enim laborum iusto itaque quasi impedit blanditiis magni quia. Vero doloribus optio quisquam deserunt perspiciatis! Facilis eius modi aut.</p>
46
47     <p>P: Lorem ipsum dolor sit amet consectetur, adipisicing elit. Rerum harum sint enim laborum iusto itaque quasi impedit blanditiis magni quia. Vero doloribus optio quisquam deserunt perspiciatis! Facilis eius modi aut.</p>
48
49     <p>P: Lorem ipsum dolor sit amet consectetur, adipisicing elit. Rerum harum sint enim laborum iusto itaque quasi impedit blanditiis magni quia. Vero doloribus optio quisquam deserunt perspiciatis! Facilis eius modi aut.</p>
50
51     <p>P: Lorem ipsum dolor sit amet consectetur, adipisicing elit. Rerum harum sint enim laborum iusto itaque quasi impedit blanditiis magni quia. Vero doloribus optio quisquam deserunt perspiciatis! Facilis eius modi aut.</p>
52
53     <p>P: Lorem ipsum dolor sit amet consectetur, adipisicing elit. Rerum harum sint enim laborum iusto itaque quasi impedit blanditiis magni quia. Vero doloribus optio quisquam deserunt perspiciatis! Facilis eius modi aut.</p>
54   </div>
55 </body>
56
57 </html>
```

POSICIONAMENTO

- Para o posicionamento dos itens há também a propriedade `display`, que pode assumir valores como
 - `block`: Mostra um elemento como um elemento bloco (como um `<p>`, `<div>`, `<table>`), um abaixo do outro, iniciando uma nova linha e assumindo toda a largura da janela
 - `inline`: Exibe um elemento como se estivesse *inline* – na mesma linha (como um ``, `<a>`, ``), com a largura sendo definida pelo conteúdo do elemento – propriedades de largura e altura não têm nenhum efeito
 - `inline-block`: Permite definir a largura e altura do elemento e não adiciona uma quebra de linha depois do elemento, permitindo que ele possa ficar ao lado de outros, respeitando a margem do bloco
 - `none`: O elemento é completamente removido
 - `table`: O elemento irá se comportar como uma tabela
 - `list-item`: O elemento irá se comportar como um elemento de uma lista
 - `flex`: Mostra o elemento como um contêiner do tipo *flex*
 - `grid`: Mostra o elemento como um contêiner do tipo *grid*

POSICIONAMENTO

■ Exemplos

TIPOGRAFIA

■ Conceitos

- px = pixels
- in = polegadas
- cm = centímetros
- em e rem (*root em*)

■ No Bootstrap, como a base é de 16 px, então

- 2 em equivale a 32 px
- 0,5 em equivale a 8 px

FONTES

■ Google Fonts

- Plataforma de fontes mais utilizada
- Mais de 800 fontes

■ Font Awesome

- 2^a plataforma de fontes mais utilizada
- Incorporada ao Bootstrap

Google Fonts



MEDIA TYPES

■ Tipos de mídias

- **all:** Todos os dispositivos
- **aural:** Sintetizadores de voz
- **braille:** Leitores de Braille
- **embossed:** Impressoras de Braille
- **handheld:** Dispositivos de mão, por exemplo, celulares com telas pequenas
- **print:** Impressoras convencionais
- **projection:** Apresentações de slides
- **screen:** Monitores coloridas
- **tty:** Teleimpressores e terminais
- **tv:** Televisores

MEDIA TYPES

- A utilização é feita através da definição de folhas de estilo específicas para cada mídia e posterior aplicação na página HTML, como mostrado abaixo

```
<link rel="stylesheet" media="print" href="print.css" />
```

MEDIA TYPES

■ Exemplo



aula4 - media_types.html

```
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3 <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <link rel="stylesheet" type="text/css" media="screen"
8         href="assets/css/estilo-tela.css">
9     <link rel="stylesheet" type="text/css" media="print"
10        href="assets/css/estilo-impressao.css">
11     <title>Media Types</title>
12 </head>
13 <body>
14     <p>
15         Lorem ipsum dolor sit, amet consectetur adipisicing elit. Quia inventore quidem fuga ex deleniti iste autem rerum, necessitatibus recusandae, soluta excepturi eligendi, sed debitis fugiat minima ducimus eum dolore accusamus
16     </p>
17 </body>
18 </html>
```



aula4 - estilo-tela.css

```
1 p {
2     font-size: 40px;
3     color: blue;
4 }
```

aula4 - estilo-impressao.css

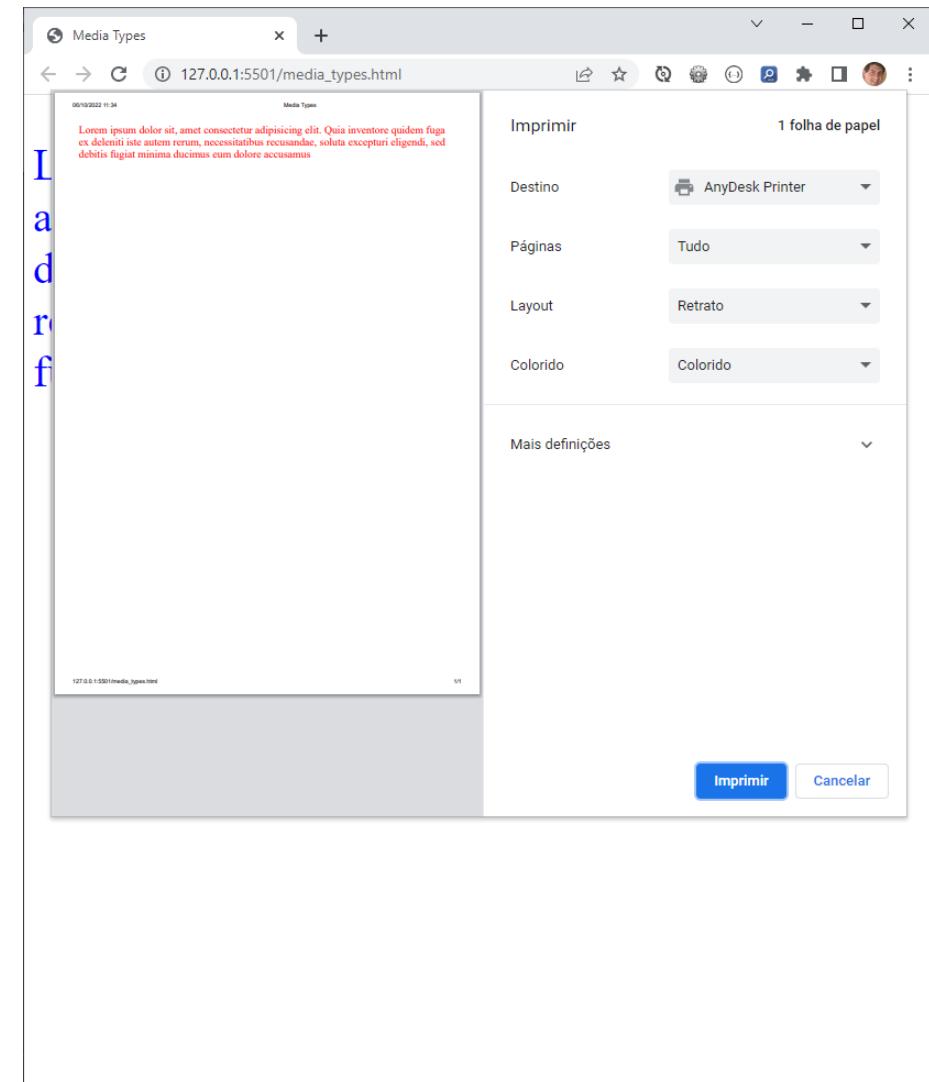
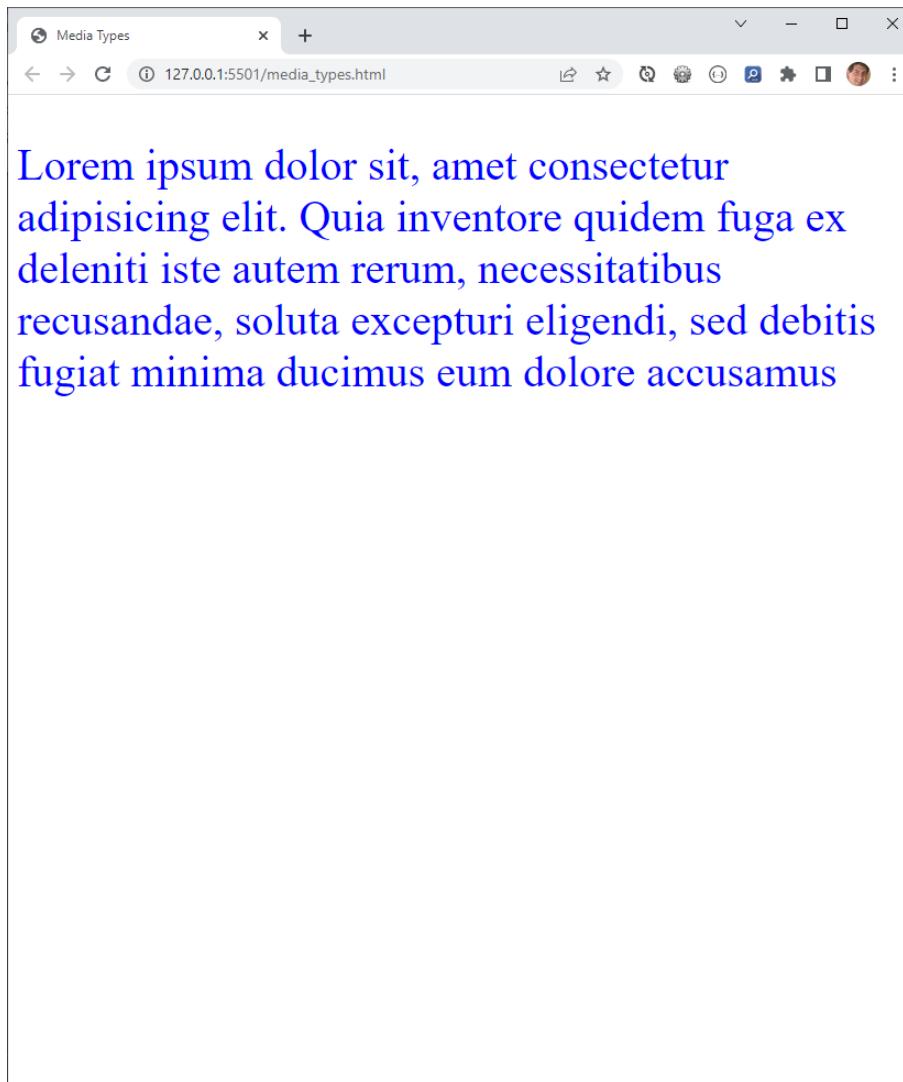
```
1 p {
2     font-size: 20px;
3     color: red;
4 }
```

Apoadores:



MEDIA TYPES

■ Exemplo de saída para tela e impressora



MEDIA QUERIES

- Media queries são uma evolução dos *media types*
 - Base de tecnologias como o Bootstrap
 - Possibilitar aplicar formatações diferentes de acordo com a resolução do dispositivo
- Exemplo de utilização de uma *media query*

```
@media (min-width: 576px) { ... }
```

MEDIA QUERIES

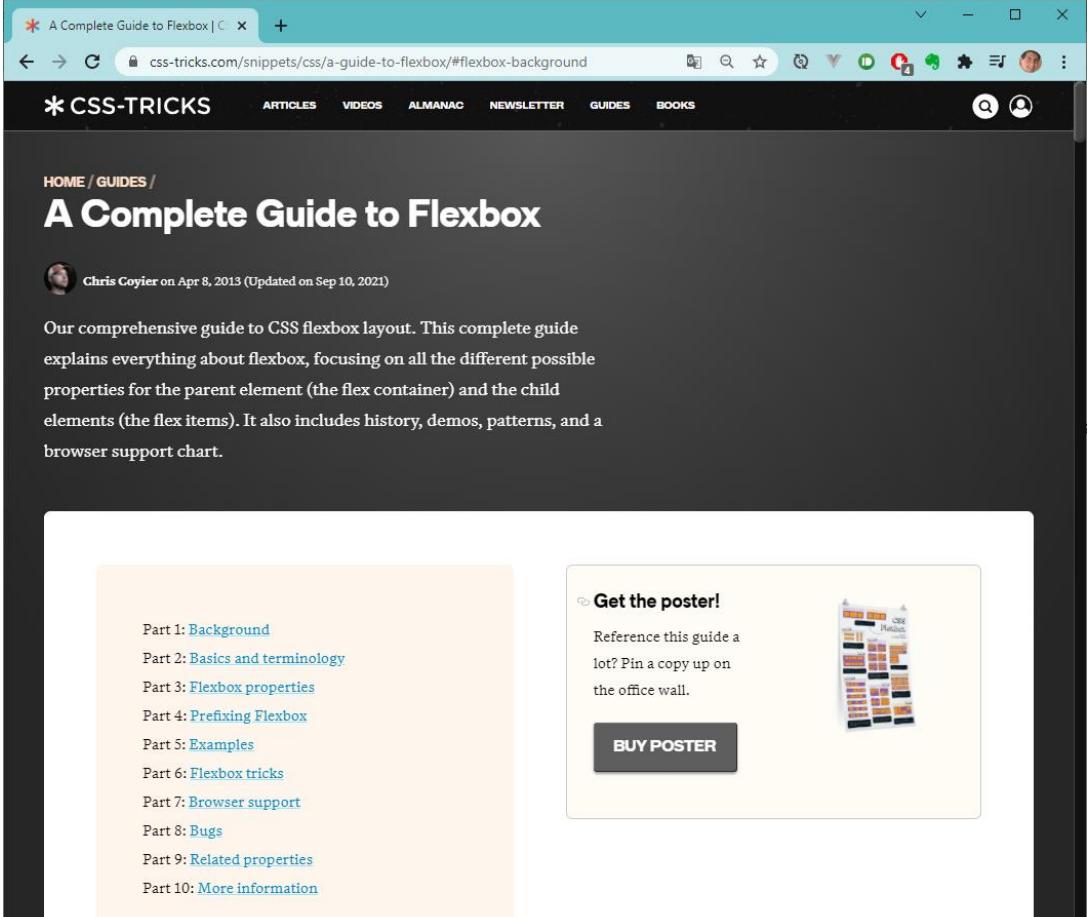
■ Exemplos de resoluções de telas

- 320 pixels: Smartphones no modo retrato
- 480 pixels: Smartphones no modo paisagem
- 600 pixels: Tablets pequenos, ex: Amazon Kindle (600×800)
- 768 pixels: Tablets maiores em modo retrato, ex: iPad (768×1024)
- 1024 pixels: Tablets maiores em modo paisagem, monitores antigos
- 1200 pixels: Monitores wide-screen

FLEXBOX

■ Flexbox (Flexible Box Layout – Layout de Caixa Flexível): Sistema de *layout* do CCS3 – lançado em 2017

- A página ao lado fornece um guia completo
- Este método é ideal quando se tem um conjunto de itens de tamanhos diferentes e se deseja que eles se encaixem confortavelmente em uma ou mais fileiras, com itens menores ocupando menos espaço e itens maiores ocupando mais espaço



The screenshot shows a web browser displaying the article 'A Complete Guide to Flexbox' from CSS-Tricks. The page has a dark background with white text. At the top, there's a navigation bar with links for HOME, GUIDES, ARTICLES, VIDEOS, ALMANAC, NEWSLETTER, GUIDES, and BOOKS. Below the navigation, the title 'A Complete Guide to Flexbox' is prominently displayed. A small profile picture and the author's name, Chris Coyier, are shown with the date of publication (Apr 8, 2013) and the last update (Sep 10, 2021). The main content area contains a paragraph explaining what the guide covers, mentioning it's a comprehensive guide to CSS flexbox layout, focusing on properties for parent and child elements, history, demos, patterns, and browser support. To the right of the main content, there's a sidebar with a section titled 'Get the poster!' featuring a thumbnail of a poster and a 'BUY POSTER' button. On the left side of the main content, there's a list of ten parts or sections related to the guide.

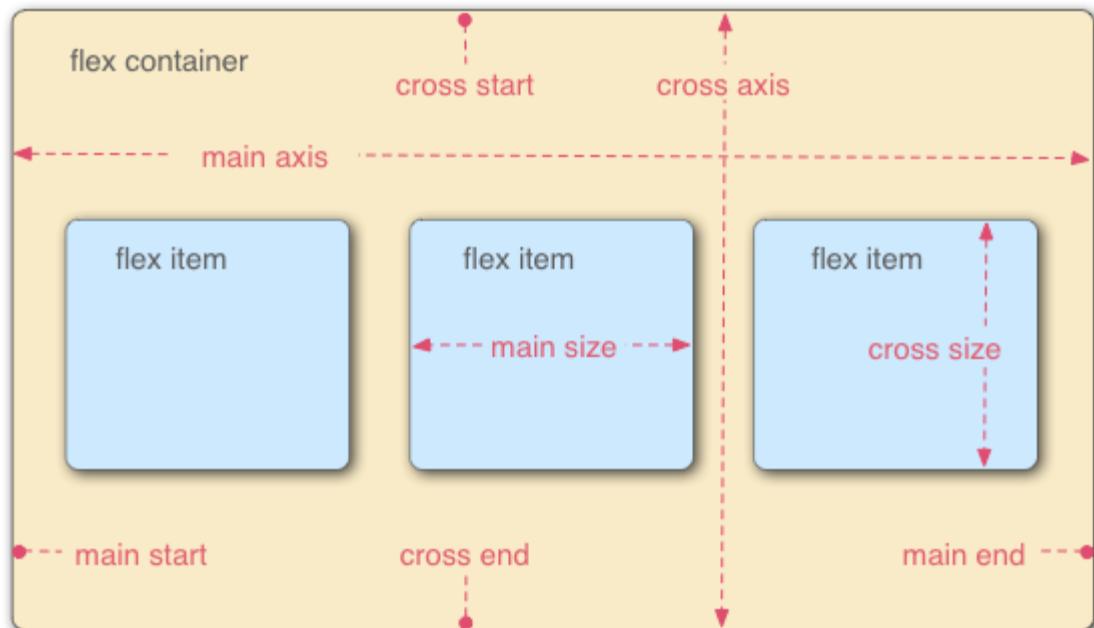
Part 1: [Background](#)
Part 2: [Basics and terminology](#)
Part 3: [Flexbox properties](#)
Part 4: [Prefixing Flexbox](#)
Part 5: [Examples](#)
Part 6: [Flexbox tricks](#)
Part 7: [Browser support](#)
Part 8: [Bugs](#)
Part 9: [Related properties](#)
Part 10: [More information](#)

FLEXBOX

■ Propriedades

■ Contêiner Flex

- display
- flex-direction
- flex-wrap
- flex-flow
- justify-content
- align-items
- align-content



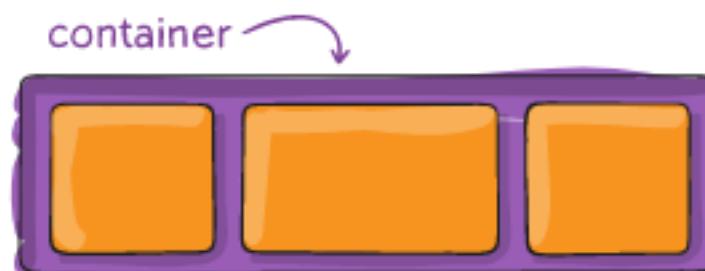
■ Item Flex

- flex-grow
- flex-basis
- flex-shrink
- flex
- order
- align-self

FLEXBOX

- Um contêiner Flex é o elemento que envolve sua estrutura
 - Este contêiner é definido com a propriedade `display` e os valores `flex` ou `inline-flex`

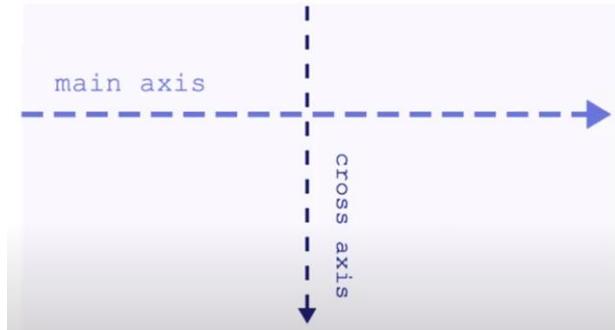
```
<div class="flex-container">  
  <div>1</div>          .flex-container {  
  <div>2</div>          display: flex;  
  <div>3</div>          }  
</div>
```



FLEXBOX

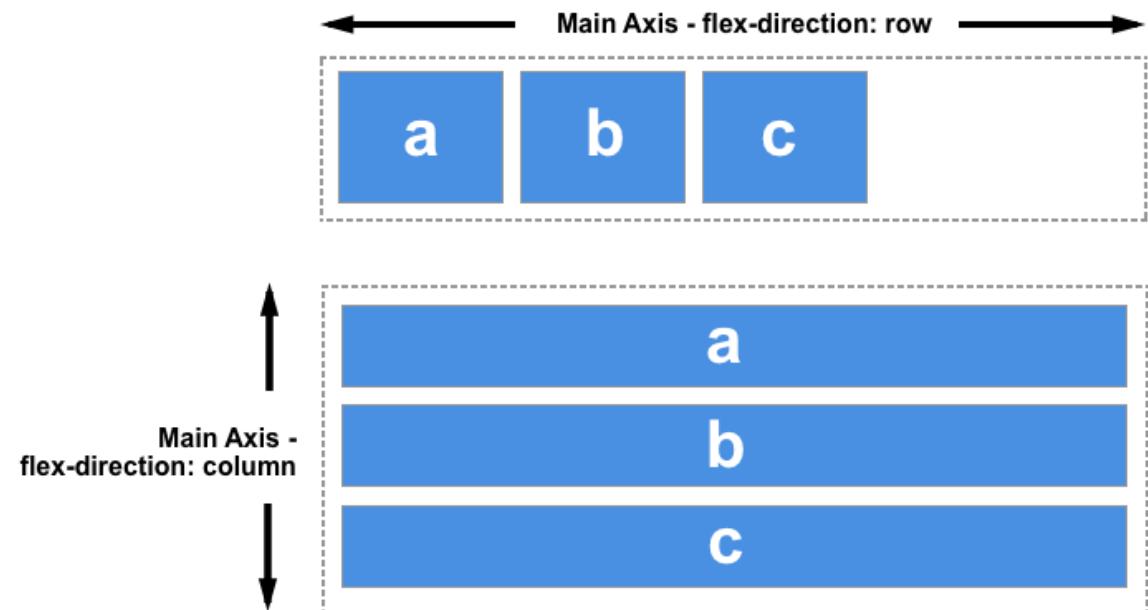
■ Flexbox trabalha com um sistema de dois eixos

- *Main axis* (eixo principal): Eixo primário de orientação, é definido na propriedade `flex-direction`
- *Cross axis* (eixo secundário ou transversal): É o eixo perpendicular (ortogonal) ao eixo principal



■ A propriedade `flex-direction` pode assumir os seguintes valores

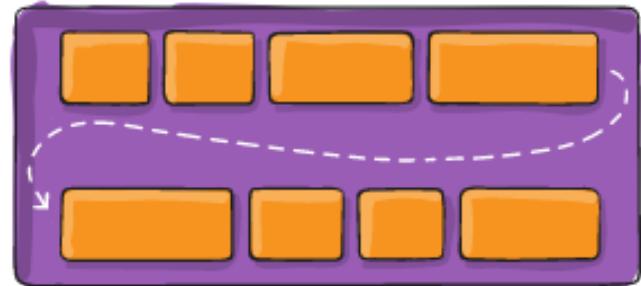
- `row` (padrão)
- `row-reverse`
- `column`
- `column-reverse`



FLEXBOX

■ Propriedade `flex-wrap`: Define se os itens devem quebrar ou não a linha

- Por padrão eles não quebram linha, isso faz com que os itens flex sejam exibidos além do limite do conteúdo



■ Essa propriedade pode assumir os seguintes valores

- nowrap (padrão)
- wrap
- wrap-reverse

`flex-wrap: nowrap;`

`TesteDoItem1` `TesteDoItem2` `TesteDoItem3`

`flex-wrap: wrap;`

`TesteDoItem1` `TesteDoItem2`
`TesteDoItem3`

`flex-wrap: wrap-reverse;`

`TesteDoItem3`
`TesteDoItem1` `TesteDoItem2`

FLEXBOX

- Propriedade `justify-content`: Alinha os elementos de acordo com a direção e só funciona se os itens atuais não ocuparem todo o contêiner

- Pode assumir os seguintes valores

- `flex-start` (*default*)
- `flex-end`
- `center`
- `space-between`
- `space-around`
- `space-evenly`



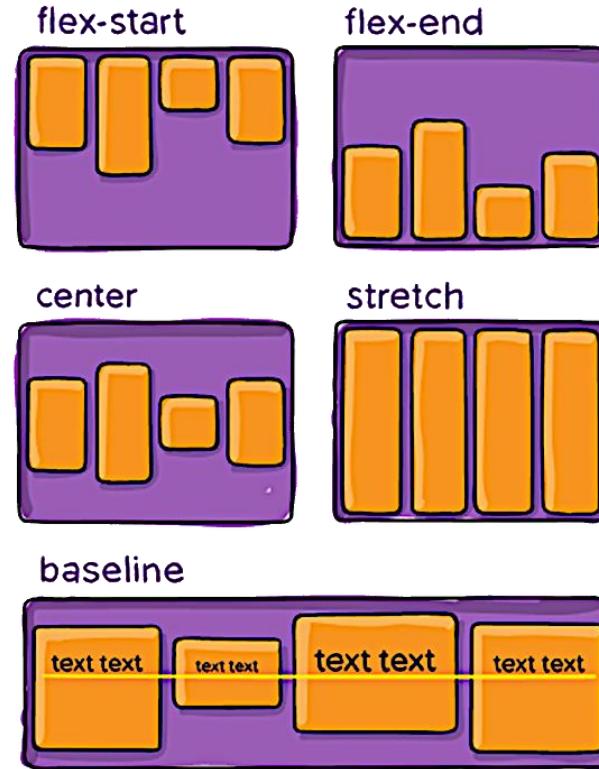
FLEXBOX

- Propriedade align-items: Define como será o alinhamento dos elementos ao longo do eixo secundário (cross-axis) do contêiner

- Isso permite o alinhamento no eixo vertical

- Pode assumir os seguintes valores

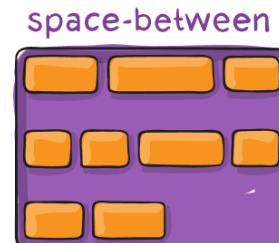
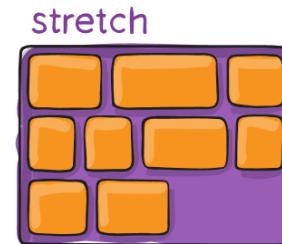
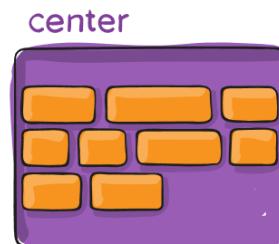
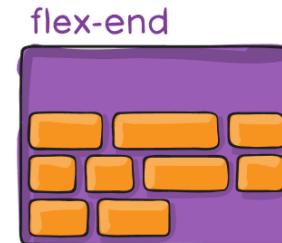
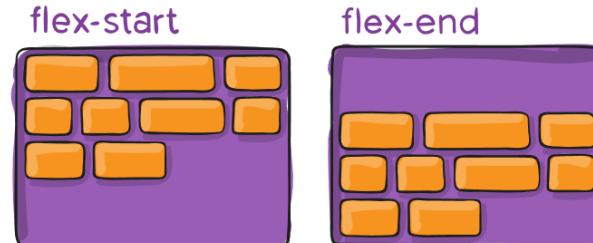
- stretch (*default*)
 - flex-start
 - flex-end
 - center
 - baseline



FLEXBOX

- Propriedade align-content: Define o comportamento do alinhamento dos elementos ao longo do eixo secundário quando há mais de uma linha
 - É similar à propriedade justify-content para alinhamento no eixo principal

- Pode assumir os seguintes valores
 - normal (*default*)
 - flex-start
 - flex-end
 - center
 - stretch
 - space-between
 - space-around
 - space-evenly



FLEXBOX

- Propriedade `flex-grow`: Define a habilidade de um item se expandir
 - Por padrão o valor é zero, assim os itens ocupam um tamanho máximo relacionado ao conteúdo interno deles
 - Ao definir 1 para todos itens, eles tentarão ter a mesma largura e vão ocupar 100% do espaço do contêiner
 - Se houver uma linha com 4 itens, sendo três `flex-grow: 1` e um `flex-grow: 2`, este último tentará ocupar 2 vezes mais espaço extra do que os outros elementos



FLEXBOX

- Propriedade `flex-basis`: Indica o tamanho inicial do item *flex* antes da distribuição do espaço livre restante
- Pode assumir os seguintes valores
 - Um tamanho: 20%, 5rem, 100px, etc
 - `auto` (padrão): Faz com que a largura da base seja igual a do item
 - 0: Se o `grow` for igual ou maior que 1, ele irá tentar manter todos os elementos com a mesma largura, independente do conteúdo (por isso 0 é o valor mais comum do `flex-basis`, caso contrário o item terá a largura do seu conteúdo)

FLEXBOX

- Propriedade `flex-shrink`: Define a capacidade de redução de tamanho do item
- Pode assumir os seguintes valores
 - 1 (padrão): Permite que os itens tenham os seus tamanhos (seja esse tamanho definido a partir de `width` ou `flex-basis`) reduzidos para caber no contêiner
 - 0: Não permite a diminuição dos itens
 - Número: Um item com valor 3 diminuirá 3 vezes mais do que um item com 1

FLEXBOX

■ Propriedade flex: Atalho para as propriedades flex-grow, flex-shrink e flex-basis combinadas

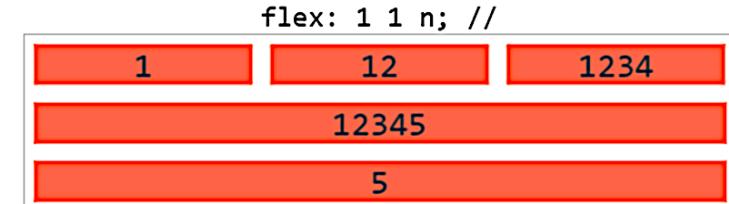
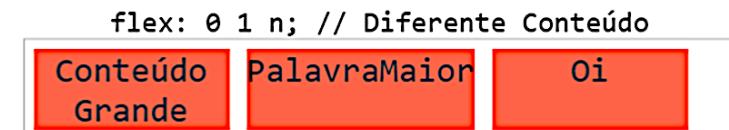
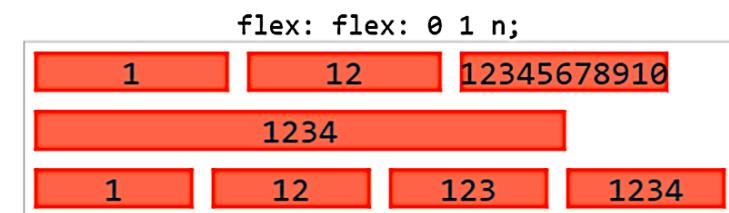
- É recomendado utilizar esta propriedade em vez de definir as propriedades individualmente pois ela altera todos os valores de forma inteligente

■ Valores

- initial: Reseta o item ao tamanho inicial
- auto: Item pode crescer e encher o contêiner
- none: Item totalmente inflexível

■ Exemplo

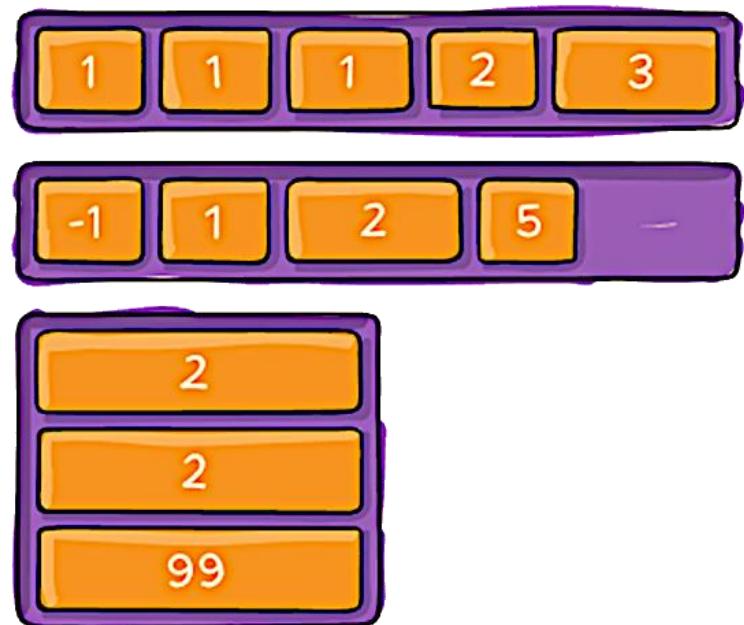
- `flex: 3 2 300px;`
- `// flex-grow: 3`
- `// flex-shrink: 2`
- `// flex-basis: 300px;`



FLEXBOX

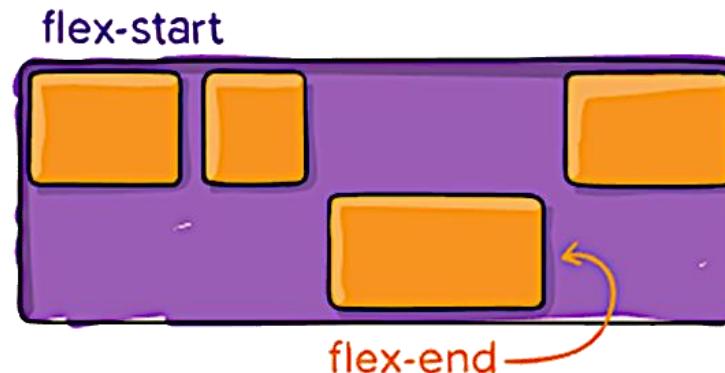
- Propriedade `order`: Modifica a ordem dos itens *flex*
 - A ordem sempre do menor para o maior, neste modo um item com `order: 1`, aparece na frente de outro item com `order: 5`

- Valores
 - Número: Número para modificar a ordem padrão, pode ser negativo
 - 0: Valor padrão, o que significa que a ordem dos itens será aquela que aparece no arquivo HTML



FLEXBOX

- Propriedade `align-self`: Define o alinhamento específico de um único item `flex` dentro do contêiner
- Pode assumir os seguintes valores
 - `stretch` (*default*)
 - `flex-start`
 - `flex-end`
 - `center`
 - `baseline`



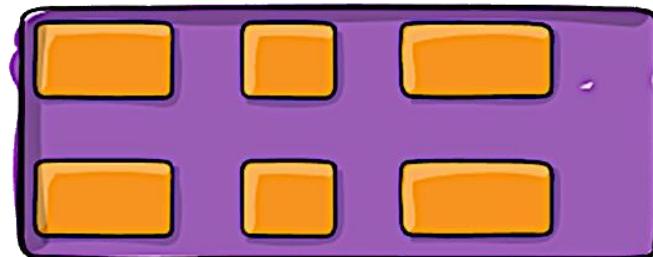
FLEXBOX

- Propriedades gap, row-gap e column-gap: Controlam o espaço entre os itens

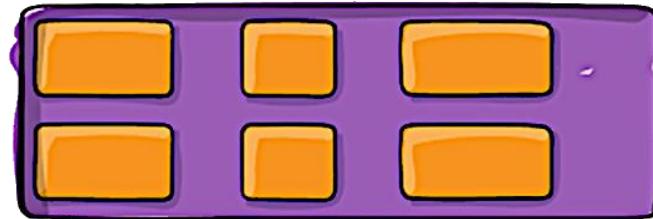
gap: 10px



gap: 30px



gap: 10px 30px



Apoadores:



FLEXBOX

Exemplo: display: block

The screenshot displays a Brackets code editor on the left and a web browser window on the right, illustrating the use of CSS Flexbox.

Brackets Editor:

- File: flex_01.html (flexbox) - Brackets
- Arquivo, Editar, Localizar, Visualizar, Navegar, Depurar, Ajuda, Emmet
- flexbox - dropdown menu
- flex_01.html
- Code View:

```
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3 <head>
4   <meta charset="UTF-8">
5   <title>CSS Flex</title>
6 <style>
7   .item {
8     margin: 5px;
9     padding: 5px;
10    background: #447;
11    text-align: center;
12    font-size: 1em;
13    color: #fff;
14  }
15
16  .container {
17    max-width: 500px;
18    margin: 0 auto;
19    border: 1px solid #000;
20    display: block;
21  }
22
23  body {
24    font-family: arial;
25    color: #000;
26  }
27 </style>
28 </head>
29
30 <body>
31  <div class="container">
32    <div class="item">Curso 1</div>
33    <div class="item">Curso 2</div>
34    <div class="item">Curso 3</div>
35    <div class="item">Curso 4</div>
36    <div class="item">Curso 5</div>
37    <div class="item">Curso 6</div>
38  </div>
39 </body>
40 </html>
```

- Linha 1, Coluna 1 — 40 linhas
- Erros: 10, INS, UTF-8, HTML, Tamanho da tabulação: 4

Browser Window:

- CSS Flex
- 127.0.0.1:27233/flex_01.html
- Curso 1
- Curso 2
- Curso 3
- Curso 4
- Curso 5
- Curso 6

FLEXBOX

■ Exemplo: display: flex

The screenshot shows a development environment with a code editor and a browser preview.

Code Editor: The left pane displays the file `flex_01.html` in Brackets. The code defines a simple layout using CSS Flexbox:

```
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3 <head>
4   <meta charset="UTF-8">
5   <title>CSS Flex</title>
6 <style>
7   .item {
8     margin: 5px;
9     padding: 5px;
10    background: #447;
11    text-align: center;
12    font-size: 1em;
13    color: #fff;
14  }
15
16  .container {
17    max-width: 500px;
18    margin: 0 auto;
19    border: 1px solid #000;
20    display: flex;
21  }
22
23  body {
24    font-family: arial;
25    color: #000;
26  }
27 </style>
28 </head>
29
30 <body>
31  <div class="container">
32    <div class="item">Curso 1</div>
33    <div class="item">Curso 2</div>
34    <div class="item">Curso 3</div>
35    <div class="item">Curso 4</div>
36    <div class="item">Curso 5</div>
37    <div class="item">Curso 6</div>
38  </div>
39 </body>
40 </html>
```

Browser Preview: The right pane shows a browser window titled "CSS Flex" with the URL `127.0.0.1:27233/flex_01.html`. It displays a horizontal row of six blue rectangular buttons, each containing white text:

Curso 1 | Curso 2 | Curso 3 | Curso 4 | Curso 5 | Curso 6

FLEXBOX

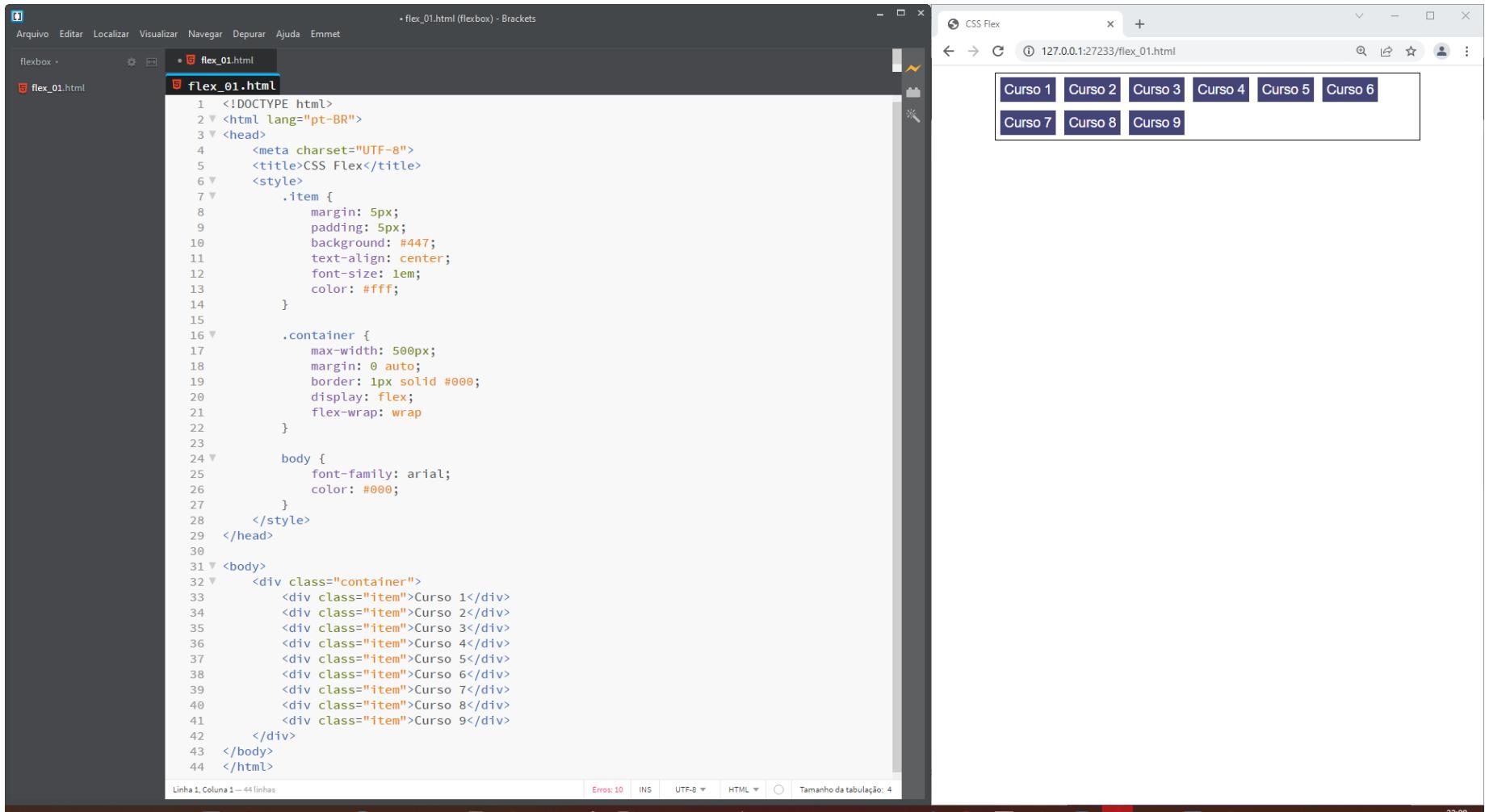
■ Problema: Ultrapassando o container

The screenshot shows a Windows desktop environment. On the left, a Brackets code editor window is open, displaying the file 'flex_01.html'. The code implements a flexbox layout with 9 items, each containing the text 'Curso' followed by a number from 1 to 9. The items are styled with a black background, white text, and 5px margins and padding. The container has a border and a max-width of 500px. On the right, a browser window titled 'CSS Flex' shows the rendered output where the 9 items are displayed in a single row, exceeding the container's width and causing horizontal scroll bars.

```
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3 <head>
4   <meta charset="UTF-8">
5   <title>CSS Flex</title>
6 <style>
7   .item {
8     margin: 5px;
9     padding: 5px;
10    background: #447;
11    text-align: center;
12    font-size: 1em;
13    color: #fff;
14  }
15
16  .container {
17    max-width: 500px;
18    margin: 0 auto;
19    border: 1px solid #000;
20    display: flex;
21  }
22
23  body {
24    font-family: arial;
25    color: #000;
26  }
27 </style>
28 </head>
29
30 <body>
31   <div class="container">
32     <div class="item">Curso 1</div>
33     <div class="item">Curso 2</div>
34     <div class="item">Curso 3</div>
35     <div class="item">Curso 4</div>
36     <div class="item">Curso 5</div>
37     <div class="item">Curso 6</div>
38     <div class="item">Curso 7</div>
39     <div class="item">Curso 8</div>
40     <div class="item">Curso 9</div>
41   </div>
42 </body>
```

FLEXBOX

Exemplo: flex-wrap: wrap



The screenshot displays a development environment with two main windows. On the left is a Brackets code editor showing the file `flex_01.html`. The code defines a CSS flexbox layout with items wrapped to multiple lines. On the right is a browser window titled "CSS Flex" showing the rendered output of the code.

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
<meta charset="UTF-8">
<title>CSS Flex</title>
<style>
.item {
    margin: 5px;
    padding: 5px;
    background: #447;
    text-align: center;
    font-size: 1em;
    color: #fff;
}

.container {
    max-width: 500px;
    margin: 0 auto;
    border: 1px solid #000;
    display: flex;
    flex-wrap: wrap
}

body {
    font-family: arial;
    color: #000;
}
</style>
</head>

<body>
<div class="container">
    <div class="item">Curso 1</div>
    <div class="item">Curso 2</div>
    <div class="item">Curso 3</div>
    <div class="item">Curso 4</div>
    <div class="item">Curso 5</div>
    <div class="item">Curso 6</div>
    <div class="item">Curso 7</div>
    <div class="item">Curso 8</div>
    <div class="item">Curso 9</div>
</div>
</body>
</html>
```

The browser preview shows a grid of nine purple rectangular boxes arranged in three rows and three columns. The boxes are labeled "Curso 1" through "Curso 9". The "flex-wrap: wrap" property causes the items to wrap onto the next line when there is no more space in the current row.

FLEXBOX

■ Centralizando

The screenshot shows a Windows desktop environment. On the left, a Brackets code editor window is open, displaying the file `flex_02.html`. The code uses CSS Flexbox to center nine course items within a container. On the right, a web browser window titled "CSS Flex" shows the rendered output of the code, displaying nine course items arranged in two rows of four, with the ninth item centered below them.

```
flex_02.html (flexbox) - Brackets
Arquivo Editar Localizar Visualizar Navegar Depurar Ajuda Emmet
flex box flex_01.html grid_01.html flex_02.html grid_02.html
flex_01.html flex_02.html grid_01.html grid_02.html
flex_01.html
flex_02.html
grid_01.html
grid_02.html

flex_02.html
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3 <head>
4   <meta charset="UTF-8">
5   <title>CSS Flex</title>
6 <style>
7   .item {
8     margin: 5px;
9     padding: 5px;
10    background: #447;
11    text-align: center;
12    font-size: 1em;
13    color: #fff;
14  }
15
16  .container {
17    display: flex;
18    width: 70%;
19    margin: auto;
20    border: 1px solid #000;
21    flex-wrap: wrap;
22    justify-content: center;
23  }
24
25  body {
26    font-family: arial;
27    color: #000;
28  }
29 </style>
30 </head>
31
32 <body>
33  <div class="container">
34    <div class="item">Curso 1</div>
35    <div class="item">Curso 2</div>
36    <div class="item">Curso 3</div>
37    <div class="item">Curso 4</div>
38    <div class="item">Curso 5</div>
39    <div class="item">Curso 6</div>
40    <div class="item">Curso 7</div>
41    <div class="item">Curso 8</div>
42    <div class="item">Curso 9</div>
43  </div>
44 </body>
Linha 1, Coluna 1 — 45 linhas
Erros: 53 INS UTF-8 HTML Tamanho da tabulação: 4
22:36 04/03/2022
```

CSS Flex

127.0.0.1:27233/flex_02.html

Curso 1	Curso 2	Curso 3	Curso 4
Curso 5	Curso 6	Curso 7	Curso 8
Curso 9			

GRID LAYOUT

- CSS grid layout é um sistema para criar *layouts* complexos em forma de grade que sejam responsivos de forma mais fácil e consistente entre diversos navegadores

One

We can use CSS Grid to overwrite older methods.

Two

We can use CSS Grid to overwrite older methods.

We can use CSS Grid to overwrite older methods.

Three

We can use CSS Grid to overwrite older methods.

Four

We can use CSS Grid to overwrite older methods.

Five

We can use CSS Grid to overwrite older methods.

Six

We can use CSS Grid to overwrite older methods.

GRID LAYOUT

■ Exemplo de layout ‘Holy grail’

The screenshot shows a Windows desktop environment with a dark blue header bar at the top. Below it is a green bar, followed by a white header bar with a search icon. The main area contains a Brackets code editor on the left and a browser window on the right.

Brackets Editor: The title bar says "grid_01.html (flexbox) - Brackets". The sidebar shows files "flex_01.html" and "grid_01.html". The code editor displays the following HTML and CSS:

```
grid_01.html (flexbox) - Brackets
Arquivo Editar Localizar Visualizar Navegar Depurar Ajuda Emmet
flexbox flex_01.html grid_01.html
flex_01.html grid_01.html
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3   <head>
4     <meta charset="UTF-8">
5     <title>CSS Flex</title>
6     <style>
7       div { border: 1px solid; }
8       body {
9         text-align: center;
10        display: grid;
11        grid-template-columns: 10em auto 10em;
12        grid-template-areas:
13          "header header header"
14          "left   middle  right"
15          "footer footer footer";
16      }
17    </style>
18  </head>
19  <body>
20    <div style="grid-area: header">Cabeçalho</div>
21    <div style="grid-area: footer">Rodapé</div>
22    <div style="grid-area: left">Painel da esquerda</div>
23    <div style="grid-area: middle; height: 200px">Área principal</div>
24    <div style="grid-area: right">Painel da direita</div>
25  </body>
26 </html>
```

Browser Window: The title bar says "CSS Flex" and the address bar shows "127.0.0.1:27233/grid_01.html". The page content is a grid layout with the following structure:

Cabeçalho		
Painel da esquerda	Área principal	Painel da direita
Rodapé		

The browser taskbar at the bottom shows various pinned icons, and the system tray indicates the date and time as "04/03/2022 22:25".

GRID LAYOUT

■ Exemplo com disposição em linhas

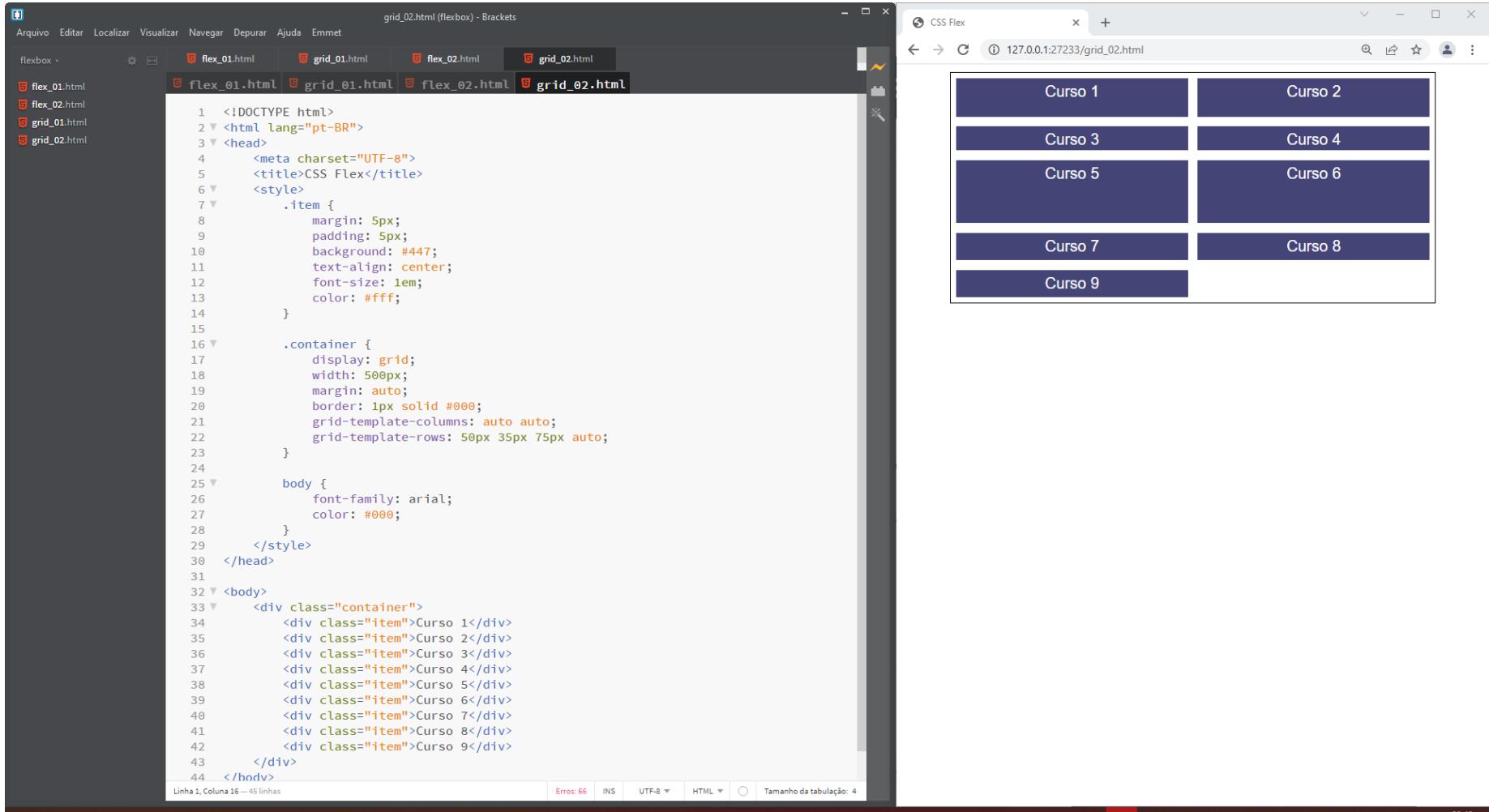
The screenshot shows a Windows desktop environment. On the left, a Brackets code editor window is open, displaying the file `grid_02.html`. The code implements a grid layout using CSS Grid. The editor interface includes tabs for other files like `flex_01.html`, `grid_01.html`, `flex_02.html`, and `grid_02.html`. The code itself defines a container with 3 columns and 3 rows, each containing a dark blue box with white text.

The right side of the screen shows a web browser window with the URL `127.0.0.1:27233/grid_02.html`. The browser displays a 3x3 grid of boxes. The first column contains "Curso 1", "Curso 3", "Curso 5", "Curso 7", and "Curso 9". The second column contains "Curso 2", "Curso 4", "Curso 6", and "Curso 8". The third column is empty.

```
grid_02.html (flexbox) - Brackets
Arquivo Editar Localizar Visualizar Navegar Depurar Ajuda Emmet
flexbox flex_01.html grid_01.html flex_02.html grid_02.html
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3 <head>
4   <meta charset="UTF-8">
5   <title>CSS Flex</title>
6   <style>
7     .item {
8       margin: 5px;
9       padding: 5px;
10      background: #447;
11      text-align: center;
12      font-size: 1em;
13      color: #fff;
14    }
15
16   .container {
17     display: grid;
18     width: 500px;
19     margin: auto;
20     border: 1px solid #000;
21     grid-template-columns: auto auto;
22   }
23
24   body {
25     font-family: arial;
26     color: #000;
27   }
28 </style>
29 </head>
30
31 <body>
32   <div class="container">
33     <div class="item">Curso 1</div>
34     <div class="item">Curso 2</div>
35     <div class="item">Curso 3</div>
36     <div class="item">Curso 4</div>
37     <div class="item">Curso 5</div>
38     <div class="item">Curso 6</div>
39     <div class="item">Curso 7</div>
40     <div class="item">Curso 8</div>
41     <div class="item">Curso 9</div>
42   </div>
43 </body>
44 </html>
```

GRID LAYOUT

Modificando o tamanho das linhas



The screenshot displays a development environment with a code editor and a browser window.

Code Editor (Brackets):

- File menu: Arquivo, Editar, Localizar, Visualizar, Navegar, Depurar, Ajuda, Emmet.
- Project: flexbox
- Open files: flex_01.html, grid_01.html, flex_02.html, grid_02.html.
- Content:

```
grid_02.html (flexbox) - Brackets
Arquivo Editar Localizar Visualizar Navegar Depurar Ajuda Emmet
flexbox flex_01.html grid_01.html flex_02.html grid_02.html
flex_01.html grid_01.html flex_02.html grid_02.html
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3 <head>
4   <meta charset="UTF-8">
5   <title>CSS Flex</title>
6   <style>
7     .item {
8       margin: 5px;
9       padding: 5px;
10      background: #447;
11      text-align: center;
12      font-size: 1em;
13      color: #fff;
14    }
15
16   .container {
17     display: grid;
18     width: 500px;
19     margin: auto;
20     border: 1px solid #000;
21     grid-template-columns: auto auto;
22     grid-template-rows: 50px 35px 75px auto;
23   }
24
25   body {
26     font-family: arial;
27     color: #000;
28   }
29 </style>
30 </head>
31
32 <body>
33   <div class="container">
34     <div class="item">Curso 1</div>
35     <div class="item">Curso 2</div>
36     <div class="item">Curso 3</div>
37     <div class="item">Curso 4</div>
38     <div class="item">Curso 5</div>
39     <div class="item">Curso 6</div>
40     <div class="item">Curso 7</div>
41     <div class="item">Curso 8</div>
42     <div class="item">Curso 9</div>
43   </div>
44 </body>
```

Linha 1, Coluna 16 — 45 linhas

Erros: 66 INS UTF-8 HTML Tamanho da tabulação: 4

Browser:

 - Title: CSS Flex
 - Address: 127.0.0.1:27233/grid_02.html
 - Content: A 3x3 grid layout with alternating row heights. The first two rows have height 50px, and the third row has height 75px. Each grid item contains the text "Curso" followed by a number from 1 to 9.

Curso 1	Curso 2
Curso 3	Curso 4
Curso 5	Curso 6
Curso 7	Curso 8
Curso 9	