



CSS

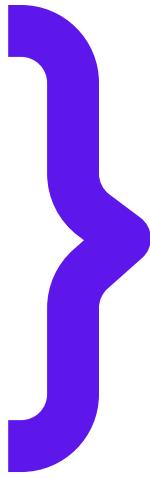
Cascading Style Sheets

BYTES4FUTURE

}{ O que é CSS?

- É uma **linguagem de estilização**;
- Um acrônimo para **Cascading Style Sheets**;
- Frequentemente **utilizada com o HTML**;
- Podemos posicionar elementos, criar animações e **reproduzir layouts na web**;





Em prática, é o que nos permite passar disto...

Cozinha da avó

Home Receitas



Descrição

Lorem ipsum dolor sit amet consectetur adipisicing elit. Odit ducimus, asperiores, cumque deleniti doloribus eos natus quos totam suscipit rem eaque! Impedit quos veniam, perspiciatis voluptate ex dolor adipisci illo.

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Commodi est assumenda molestias quasi rem nisi hic quisquam aperiam blanditiis vero magnam asperiores sequi fuga deleniti adipisci, minima voluptate recusandae quibusdam!

Ingredientes

- ovo 4 unid.
- açúcar 120 g
- cenoura ralada 300 g
- óleo vegetal 1c. de sopa
- farinha 80 g

Preparação

1. Pré-aqueça o forno a 180 °C.
2. Unte uma forma com manteiga e farinha.
3. Numa tigela bata as gemas, o açúcar, o óleo e a cenoura ralada.
4. Junte a farinha e misture com um garfo.
5. Envolva as claras batidas em castelo.
6. Deite o preparado e leve ao forno durante 30 minutos.
7. Retire do forno e desenforme o bolo.

... Para isto

COZINHA DA AVÓ

Home Receitas

Bolo de cenoura



Vamos mostar como pode fazer um ótimo bolo de cenoura, em muito pouco tempo.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Odit ducimus, asperiores, cumque deleniti doloribus eos natus quos totam suscipit rem eaque! Impedit quos veniam, perspiciatis voluptate ex dolor adipisci illo.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Odit ducimus, asperiores, cumque deleniti doloribus eos natus quos totam suscipit rem eaque! Impedit quos veniam, perspiciatis voluptate ex dolor adipisci illo.

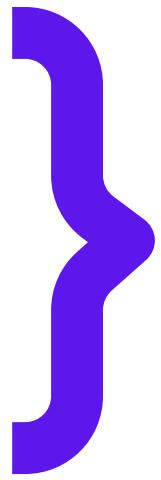
Lorem ipsum, dolor sit amet consectetur adipisicing elit. Commodo est assumenda molestias quasi rem nisi hic quisquam aperiam blanditiis vero magnam asperiores sequi fuga deleniti adipisci, minima voluptate recusandae quibusdam!

Ingredientes

- ovo 4 unid.
- açucar 120 g
- cenoura ralada 300 g
- óleo vegetal 1c. de sopa
- farinha 80 g

Preparação

1. Pré-aqueça o forno a 180 °C.
2. Unte uma forma com manteiga e farinha.
3. Numa tigela bata as gemas, o açúcar, o óleo e a cenoura ralada.
4. Junte a farinha e misture com um garfo.
5. Envolve as claras batidas em castelo.
6. Deite o preparado e leve ao forno durante 30 minutos.
7. Retire do forno e desenforme o bolo.



Sintaxe

- Para estilizar elementos vamos sempre seguir alguns passos;
- Escolher a tag (por meio do nome, classe ou id);
- Abrir e fechar chavetas, delimitando o espaço dos estilos do elemento;
- Escrever as regras de estilização;
- Todas devem terminar com um ponto e vírgula(;).

```
seletor {  
  propriedade: valor;  
}
```

Sintaxe

- **Seletor:** é um elemento da marcação HTML Identificado pelo seu nome(tag), classe (nome-da-classe) ou id(#nome-do-id);
- **Declaração:**
 - **Propriedade:** define o que mudará no elemento HTML (tamanho da fonte, cor do texto, altura do elemento).
 - **Valor:** é a característica que o elemento html irá assumir.(letra tipo Arial, cor azul, fundo verde)

```
1 <body> {
2     background-color: #000;
3 }
4
```

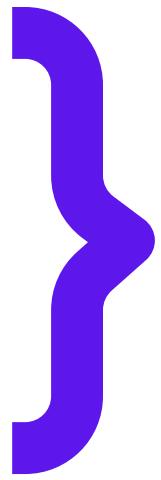
Esse exemplo define a cor de fundo preta para o site

Formas de adicionar o CSS

Temos três maneiras para adicionar CSS numa página:

- **Inline:** onde o CSS fica na tag;
- **Internal CSS:** onde o CSS fica no head da página;
- **External CSS:** onde o CSS fica em um arquivo externo;

Está última forma é a mais indicada, e devemos sempre tentar utilizá-la;

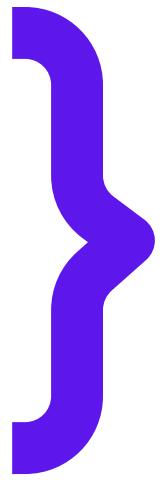


CSS Inline

Tal como o nome indica, o CSS inline é escrito ‘dentro da linha’ do elemento de HTML.

Por exemplo, se quisermos que um título fique vermelho, escrevemos assim:

```
<h1 style="color: red; font-size: 30px;">Era uma vez</h1>
```



CSS Interno

Para utilizarmos CSS interno, usamos a tag <style> dentro da tag <head> do documento HTML.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Três métodos de ligar CSS</title>
    <style>
      h1 {
        color: red;
      }
    </style>
  </head>
  <body>
    <h1>Era uma vez</h1>
  </body>
</html>
```

CSS Externo

Para podermos utilizar CSS externo, precisamos de conectar o nosso ficheiro HTML com um ficheiro externo de CSS. Para isso, utilizamos a tag `<link>` dentro da `<head>` no nosso ficheiro HTML.



The diagram shows a snippet of HTML code within a dark-themed code editor. A yellow arrow points from the text 'Liga Tipo de relação' to the 'rel' attribute of the `<link>` tag. Another yellow arrow points from the text 'Caminho ao ficheiro' to the 'href' attribute, which is underlined to indicate it is a hyperlink.

```
<head>
  <meta charset="UTF-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Três métodos de ligar CSS</title>
  <link rel="stylesheet" href="css-externo.css" />
</head>
```

Liga Tipo de relação

Caminho ao ficheiro

Selectors

Os selectors, ou selectores, tal como o nome indica é o que nos permite selecionar os elementos HTML, aos quais queremos modificar propriedades.

Temos formas além das tags de selecionar elementos:

- **Por classes:** quando queremos afetar mais elementos com uma regra;
- **Por identificadores:** quando queremos afetar um único elemento;

Estas formas são referentes aos atributos **class** que no CSS representado por **.** (ponto) e **id** que no CSS é representado por **#** (cardinal);

Especificidade

Caso tenhamos mais que uma regra para um mesmo elemento, a especificidade é um valor invisível que irá calcular que regra de CSS é que se aplica a esse mesmo elemento.

De maior especificidade para menor, temos:

- CSS Inline
- Selector de ID
- Selector de Class
- Selector de Elemento

}

Cores

Em CSS uma das propriedades que podemos aplicar são as cores.

Há 3 formas de utilização das mesmas:

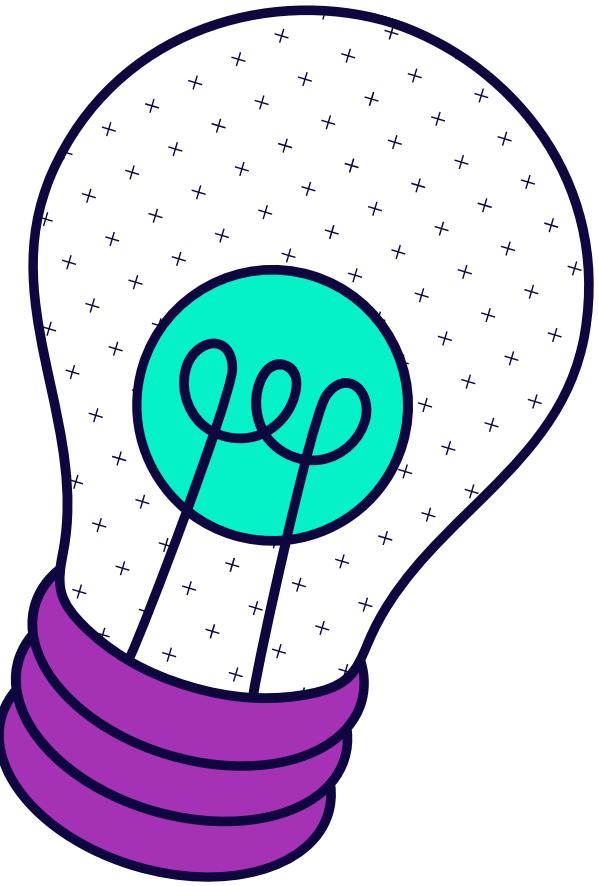
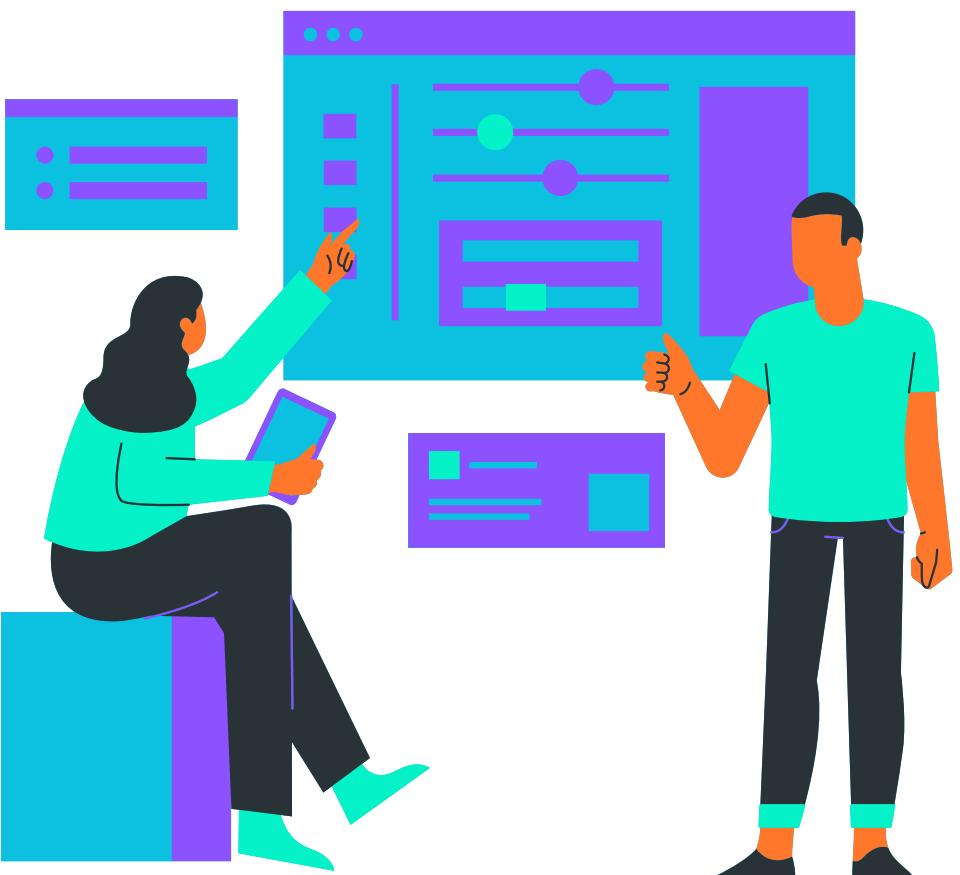
- Nome da cor
- Código Hexadecimal
- Código RGB

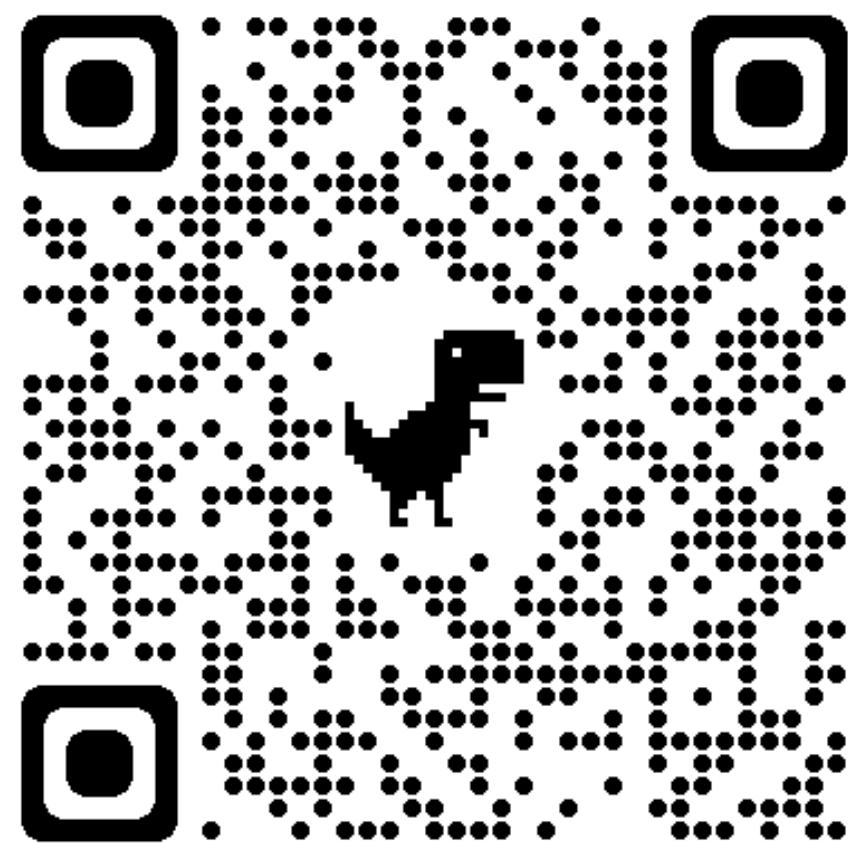


Nomes de cores
suportados por todos os
navegadores

search

QUIZ





<https://www.w3schools.com/css/>

Formatação de texto

Um dos principais elementos das páginas Web. Vejamos algumas propriedades que podemos alterar:

- **color** - altera a cor do texto
- **background-color** - altera a cor do fundo do elemento onde o texto se encontra
- **text-align** - alinha o texto (**center, left, right, justify**)
- **text-decoration:**
 - **none** - retira o sublinhado dos links
 - **overline** - linha acima do texto
 - **line-through** - riscia o texto
 - **underline** - sublinha o texto

Formatação de texto

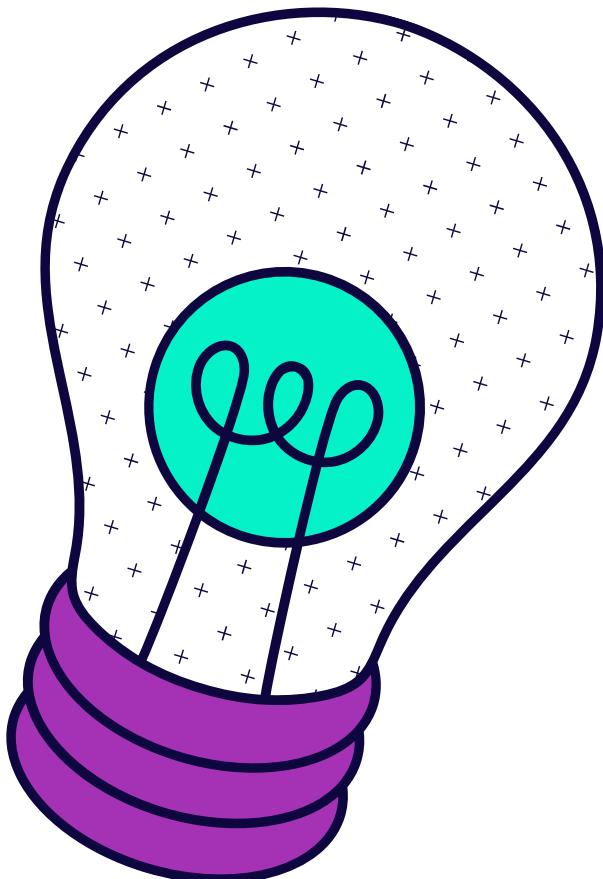
- **text-transform:**
 - uppercase - coloca todas as palavras em maiúscula
 - lowercase - coloca todas as palavras em minúscula
 - capitalize - coloca todas as palavras a iniciar com letra maiúscula
- **text-indent** - faz efeito parágrafo
- **letter-spacing** - altera o espaçamento das letras
- **line-height** - altera o espaçamento entre linhas
- **word-spacing** - altera o espaçamento entre palavras
- **text-shadow** - cria efeito de sombra no texto



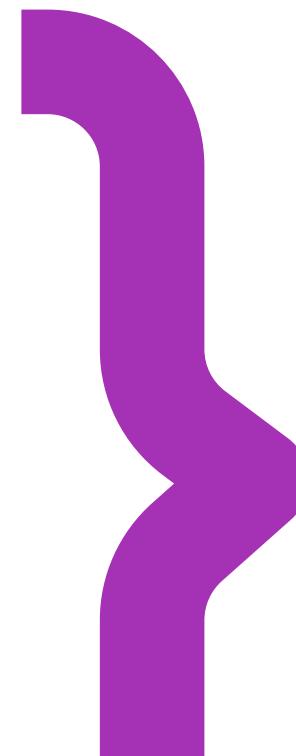
CSS Text

search

QUIZ



Vamos praticar



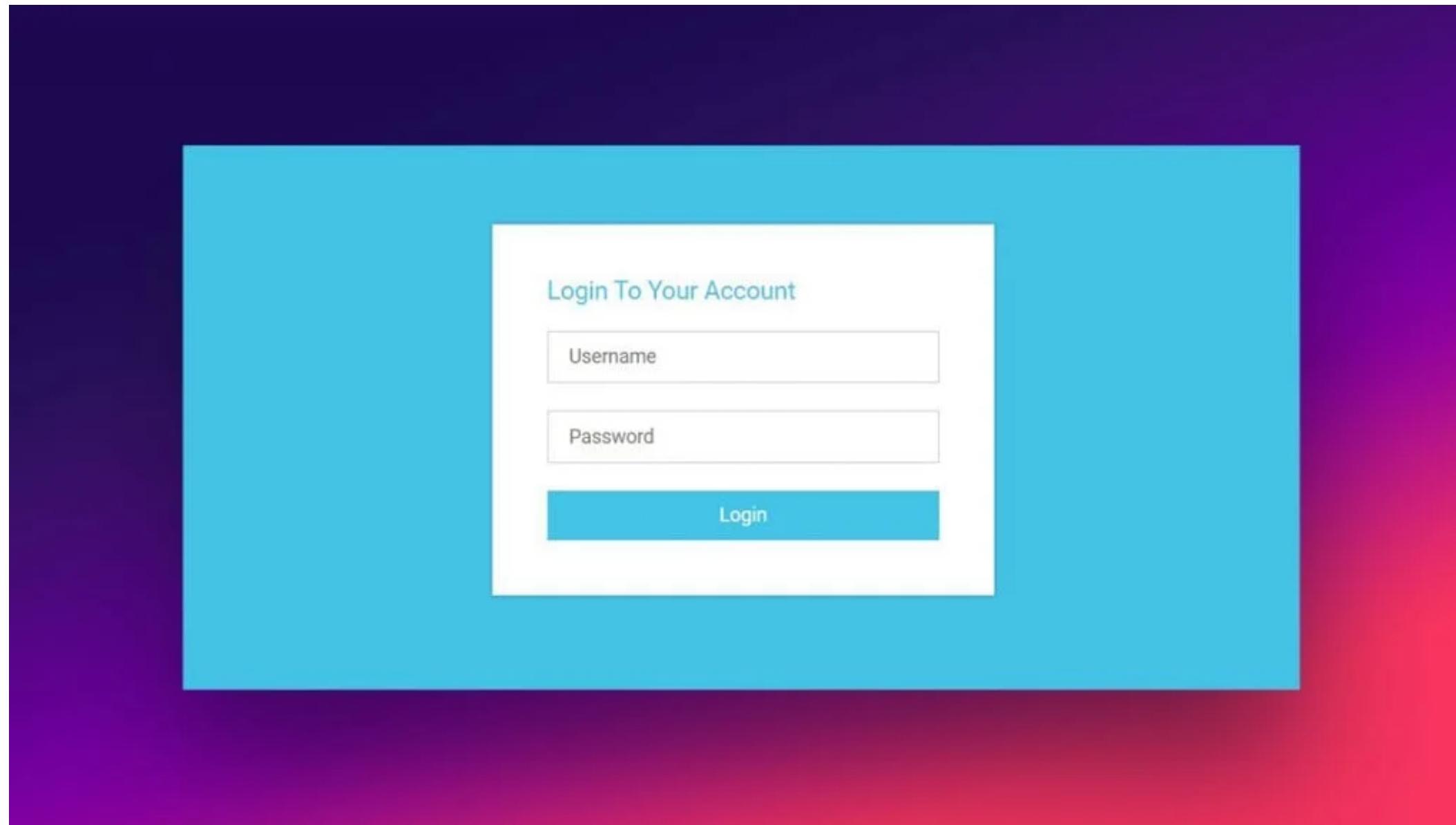
Exercício:

- Utilize a página da sua “**receita de bolo**” criada anteriormente na aula de HTML e aplique os elementos de formatação de texto e cor que vimos nesta aula.
- Na mesma página, aplique tipos de letras variados.

Exercício:

- Agora vão fazer um formulário simples. Não se preocupem ainda com o posicionamento dos vários elementos, iremos ver isso mais à frente.
- Estiliza o fundo do formulário.

Exercício:

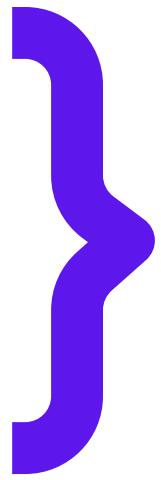




css

Cascading Style Sheets

BYTES4FUTURE

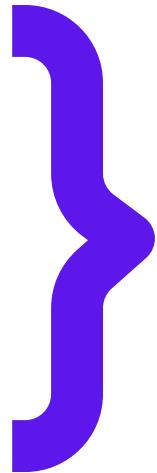


Fundos

Para alterar a cor de fundo, vimos que usamos o background-color.

Mas podemos também aplicar uma **imagem** como fundo da página.
Vamos ver como:

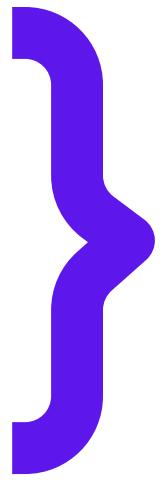
background-image: url("imagem.jpg");



Fundos

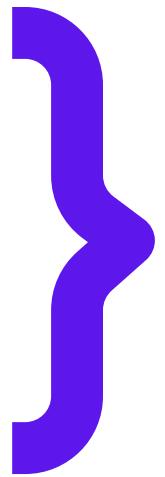
É possível associar as seguintes propriedades:

- **background-repeat: repeat-x;** -> repete a imagem na horizontal
- **background-repeat: repeat-y;** -> repete a imagem na vertical
- **background-repeat: no-repeat;** -> não repete a imagem
- **background-attachment:**
 - **fixed;** -> mantém imagem fixa quando deslizamos a página
 - **scroll;** -> imagem desliza enquanto navegamos a página



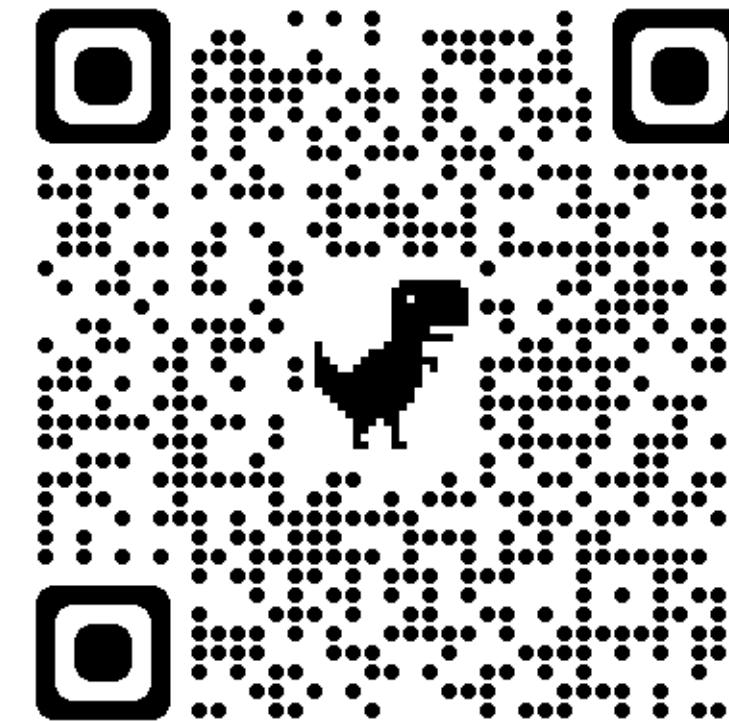
Fundos

- **background-size:**
 - **auto;** - A imagem de fundo será exibida em seu tamanho original.
 - **cover;** - será dimensionada para cobrir completamente o container, mantendo a proporção e cortando partes da imagem, se necessário.
 - **contain;** - será dimensionada para caber completamente dentro do container, mantendo a proporção e deixando espaço vazio se necessário.

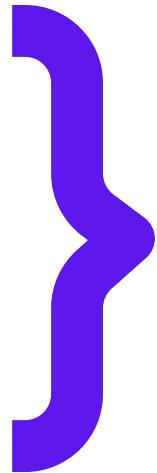


Fundos

Há muitas outras propriedades que vocês podem explorar, podem verificar em:



CSS background



Fontes

Para alterar a fonte, deve-se usar a propriedade **font-family**.

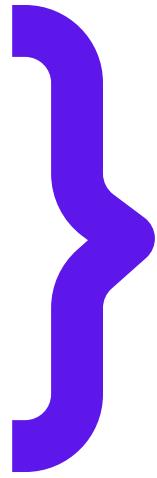
Podem ser ainda aplicadas outras propriedades:

- **font-style** - para definir se letra em normal ou itálico
- **font-weight** - para definir se letra normal ou negrito
- **font-size** - permite alterar o tamanho da fonte

É possível também importar fontes, através do **Google Fonts**.

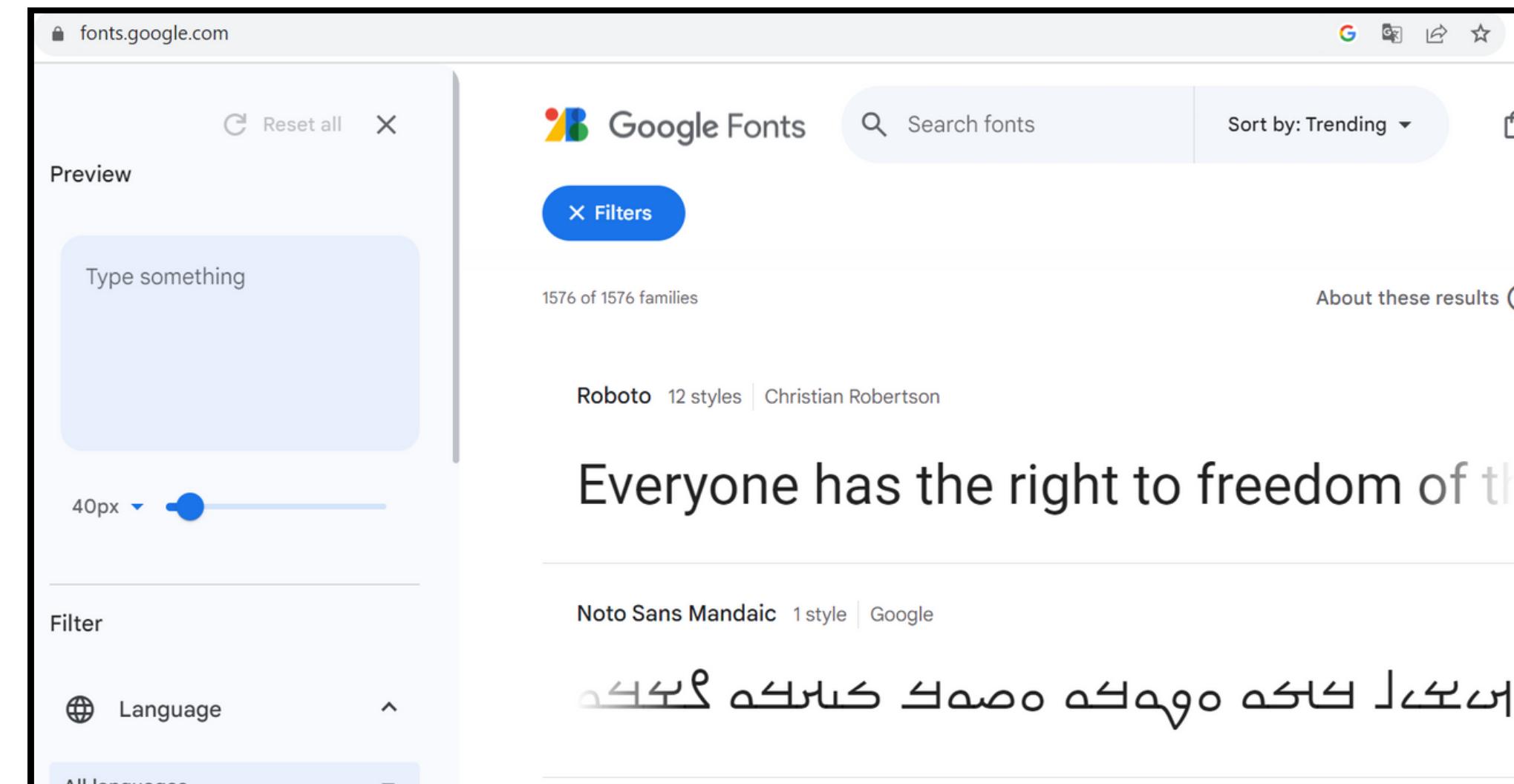


Google Fonts



Fontes

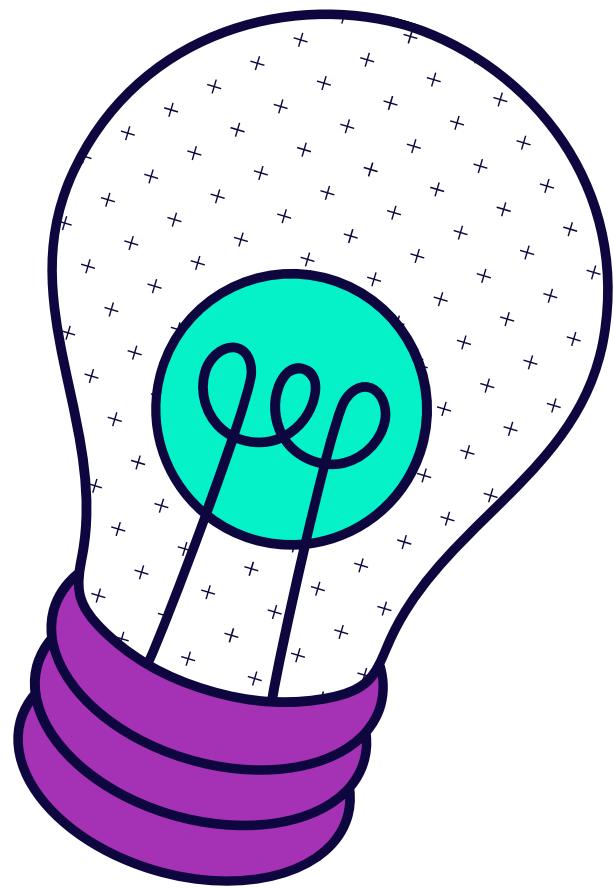
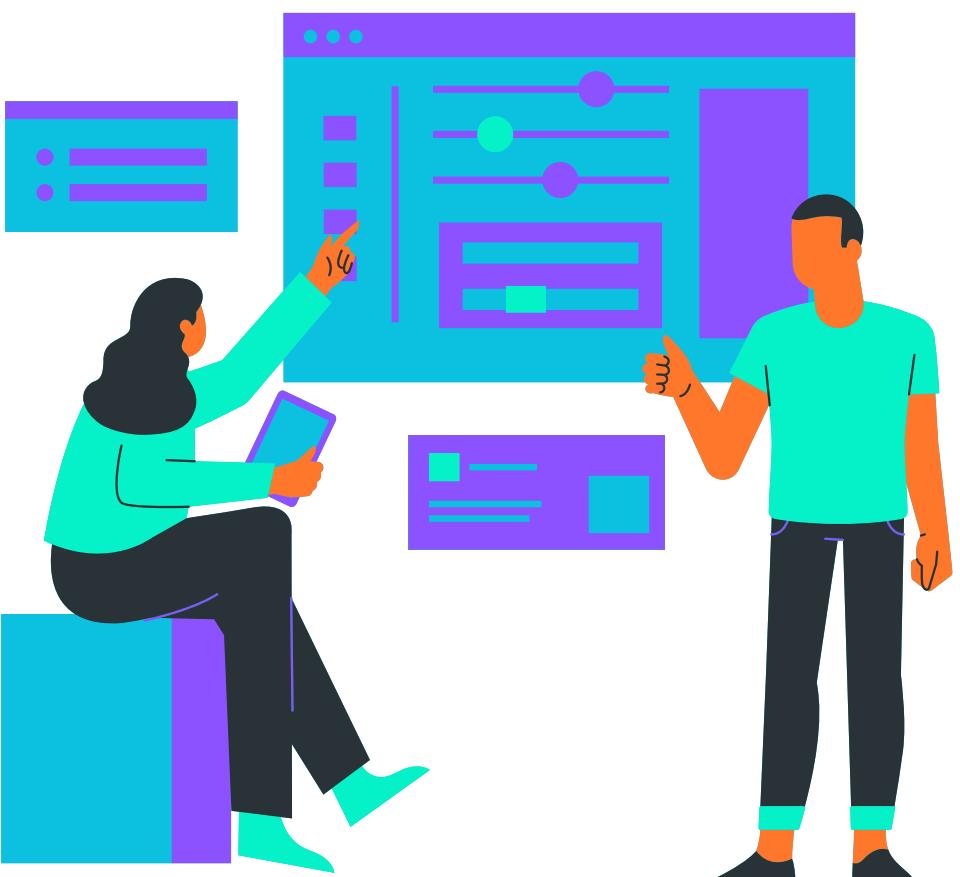
Como importar fontes do **Google Fonts**.



<https://fonts.googleapis.com/>

search

QUIZ



Pseudo-Classes

Pseudo-classe em CSS é um seletor que especifica um estado especial de um elemento.

Elas são utilizadas para selecionar elementos com base em características como interações do utilizador, estados de foco, visitas e relações com outros elementos.

São representadas por dois pontos (:) antes do nome da pseudo-classe.

Pseudo-Classes

```
/* Seleciona links quando o mouse está sobre eles */
a:hover {
    color: blue;
}

/* Seleciona um elemento quando está sendo clicado */
button:active {
    background-color: #e0e0e0;
}

/* Seleciona um input quando está em foco */
input:focus {
    border: 2px solid #ff0000;
}

/* Seleciona o primeiro filho de um elemento */
li:first-child {
    font-weight: bold;
}
```

Box Model

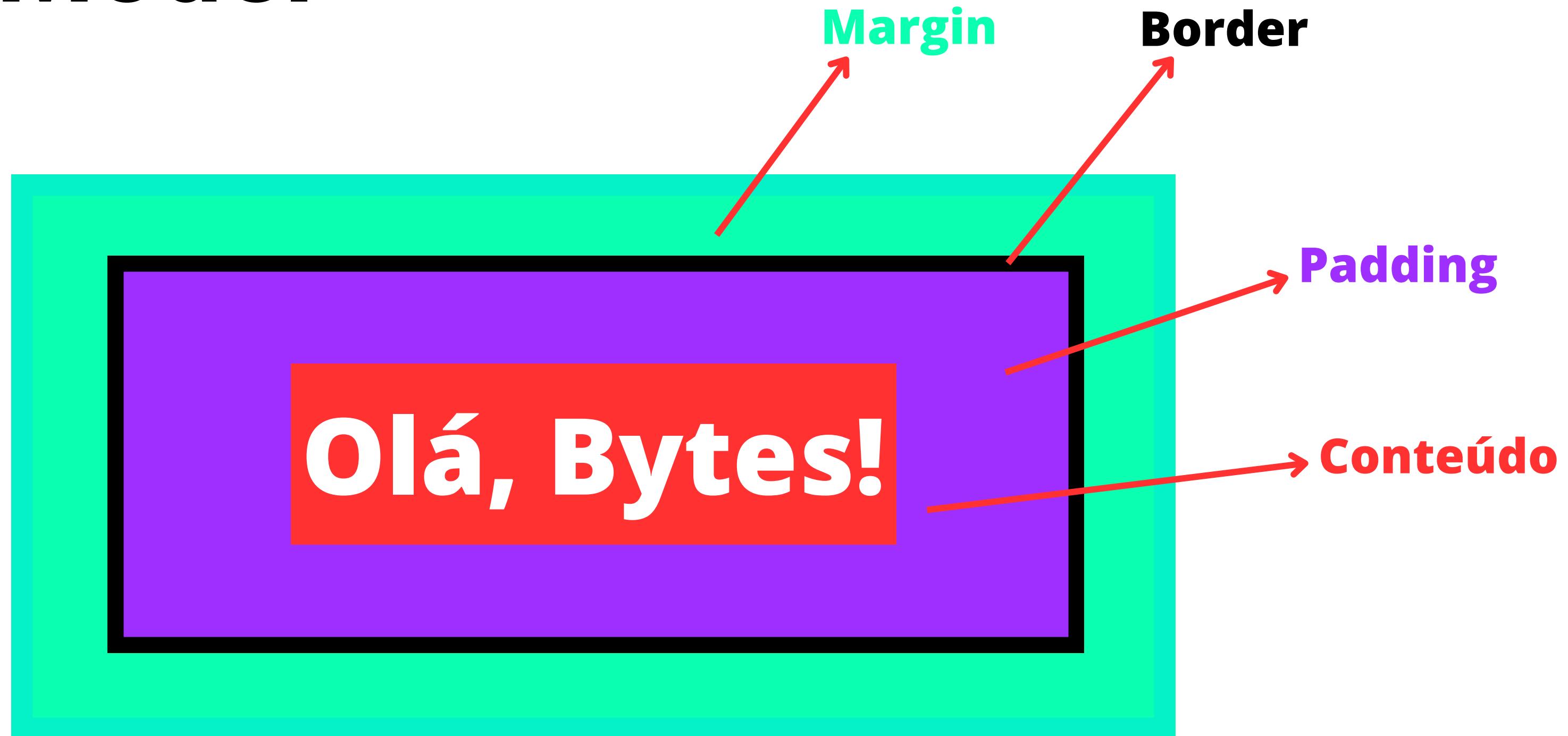
Quando falamos do **CSS Box Model**, estamos a referir-nos à maneira de como o browser vê os elementos de HTML, como boxes/caixas.

Vamos visualizar:



}

Box Model

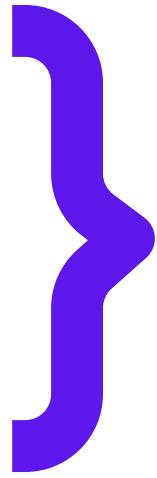


Box Model

A **margin**, **border**, e **padding** são propriedades. Ou seja, são utilizadas para estilizar elementos de HTML.

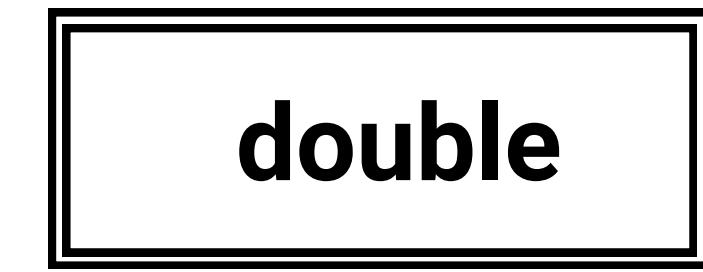
Neste caso, a **margin** refere-se ao espaço entre dois elementos.

A **border** refere-se a uma borda à volta desse mesmo elemento, e por fim, o **padding** refere-se ao espaço entre o conteúdo de um elemento e as suas bordas.



Border

Podem ser definidos utilizando atributos, começando pelo **border-style**:



Ainda é possível também aplicar estilos mistos, sempre em sentido horário.

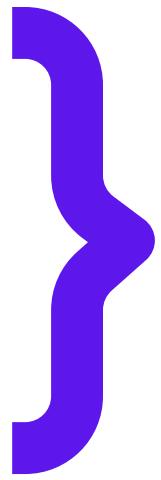


border-style: dotted dashed solid double

Border

A propriedade “**border-radius**” permite associar um valor em px para arredondar os cantos dos limites.

solid



Margin

- **Margin** permite definir o espaço que vai desde o limite do elemento até ao elemento imediatamente colado (pode ser a própria página).

Padding

- **Padding** aplica uma margem interna ao elemento, ou seja desde o limite até por exemplo ao texto que contém.

Margin e Padding

```
.margin{          .padding{  
    background-color: blue;    background-color: blue;  
    border:2px solid red;    border:2px solid red;  
    color: white;    color: white;  
    margin: 20px;    padding: 20px;  
}
```

Exemplo

Exemplo

Vamos
praticar



Exercício:

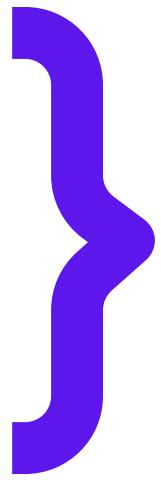
- Utiliza a página “**Formulário de Login**” criada na aula anterior e aplica os elementos de background, margin, padding, border, color.
- Na mesma página, aplica tipos de letras variados.



css

Cascading Style Sheets

BYTES4FUTURE



Flexbox

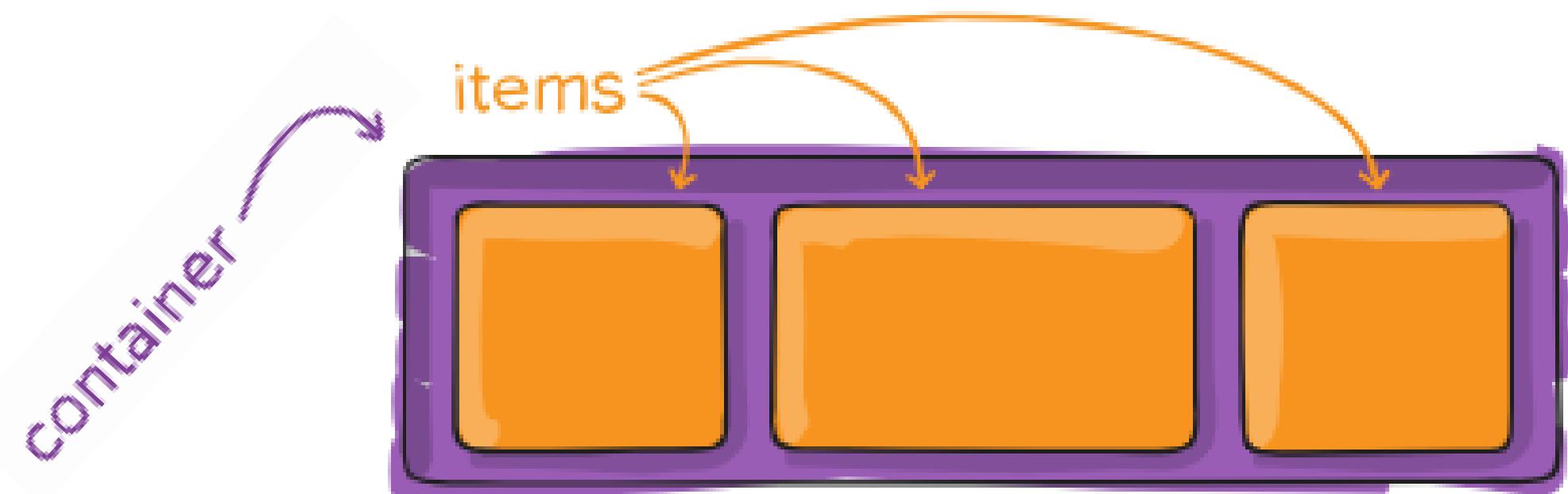
- é um **modelo de layout** em CSS projetado para facilitar o design de layouts complexos e a distribuição de espaçamento e alinhamento entre os itens de uma interface, mesmo quando o tamanho dos elementos é desconhecido ou dinâmico.

Utilidade do Flexbox

- Simplifica na distribuição de espaço ao longo de um único eixo ou ambos.
- Facilita a criação de layouts responsivos.
- Alinhamento vertical e horizontal dos itens torna-se mais simples.
- Permite reorganizar visualmente os itens sem modificar a ordem no código HTML.
- Dimensionamento Dinâmico.
- Facilidade de Uso: Isso torna o desenvolvimento mais eficiente.

Flexbox Model

Para começar a usar o modelo Flexbox, primeiro você precisa definir um flex container.



O elemento a cima representa um flex container(a área em roxo) com três flex items.

Flexbox Model

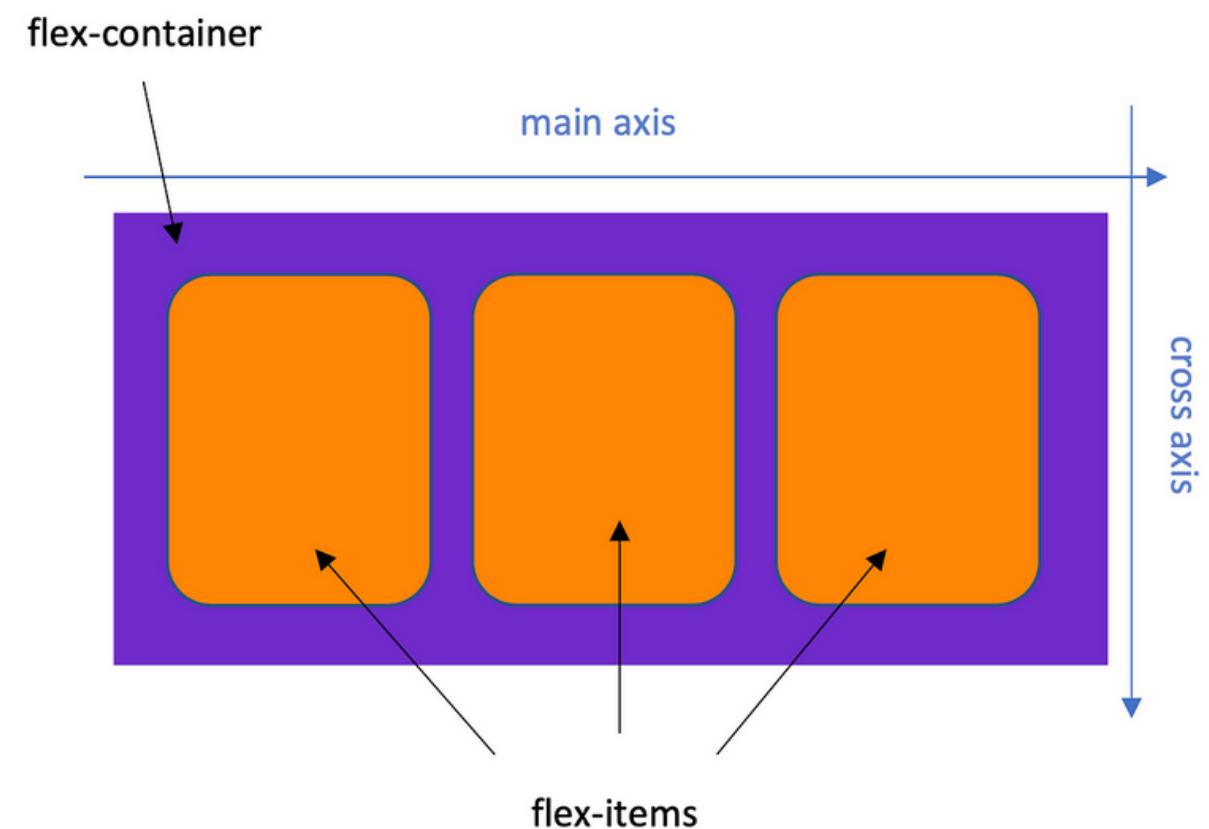
Primeiro precisamos de um flex container, ou seja, precisamos de definir a área em que queremos trabalhar.

Fazemos isso, definindo a propriedade display para flex **display: flex**.

O flex container tem dois eixos: o eixo principal e eixo secundário.

Para colocar os items de acordo o main axis (eixo principal), utilizamos a propriedade justify-content.

Para colocar os items ao longo do cross axis (eixo secundário), utilizamos a propriedade align-items.

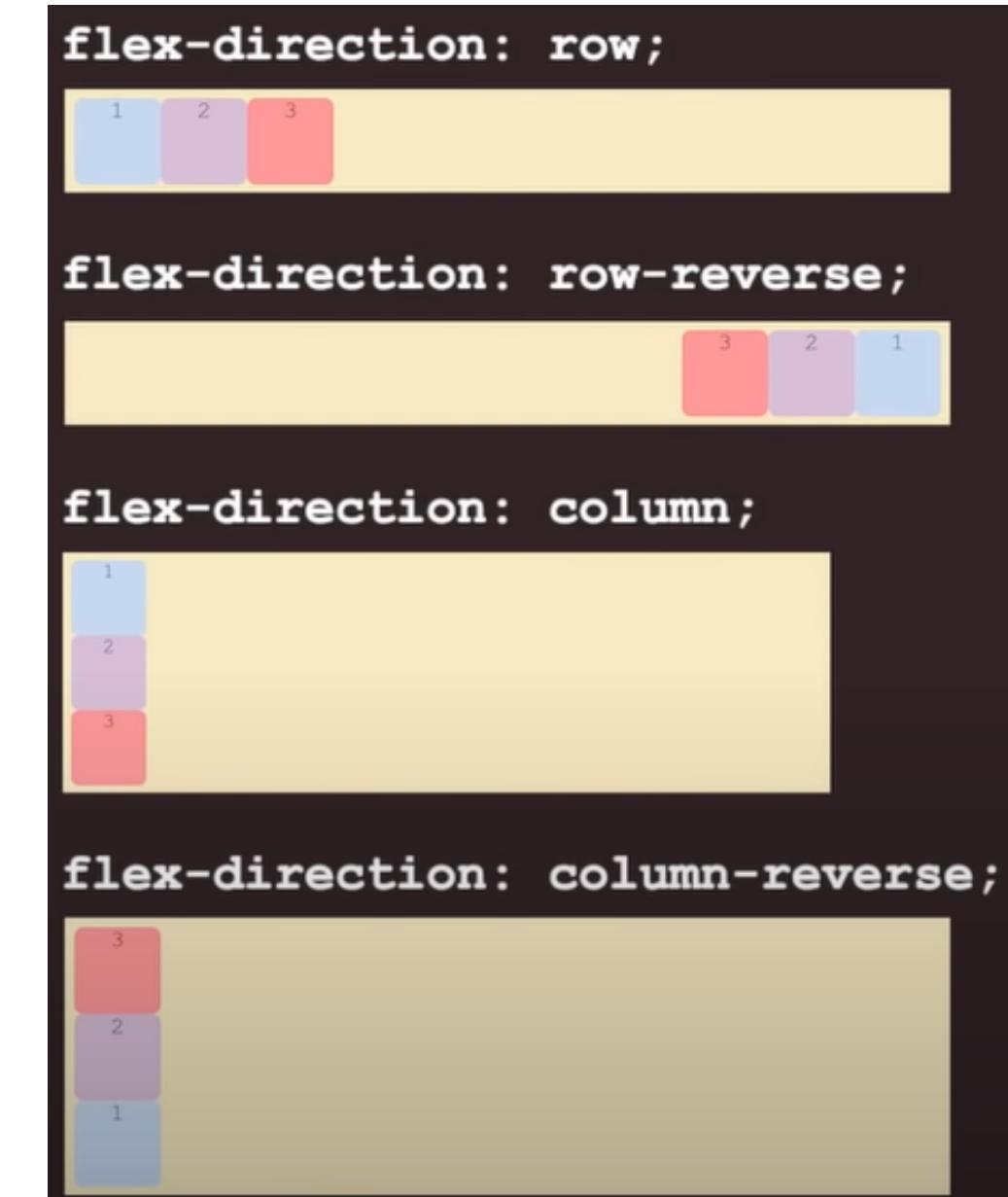




flex-direction:

Controla a direção principal dos itens no container flexível.

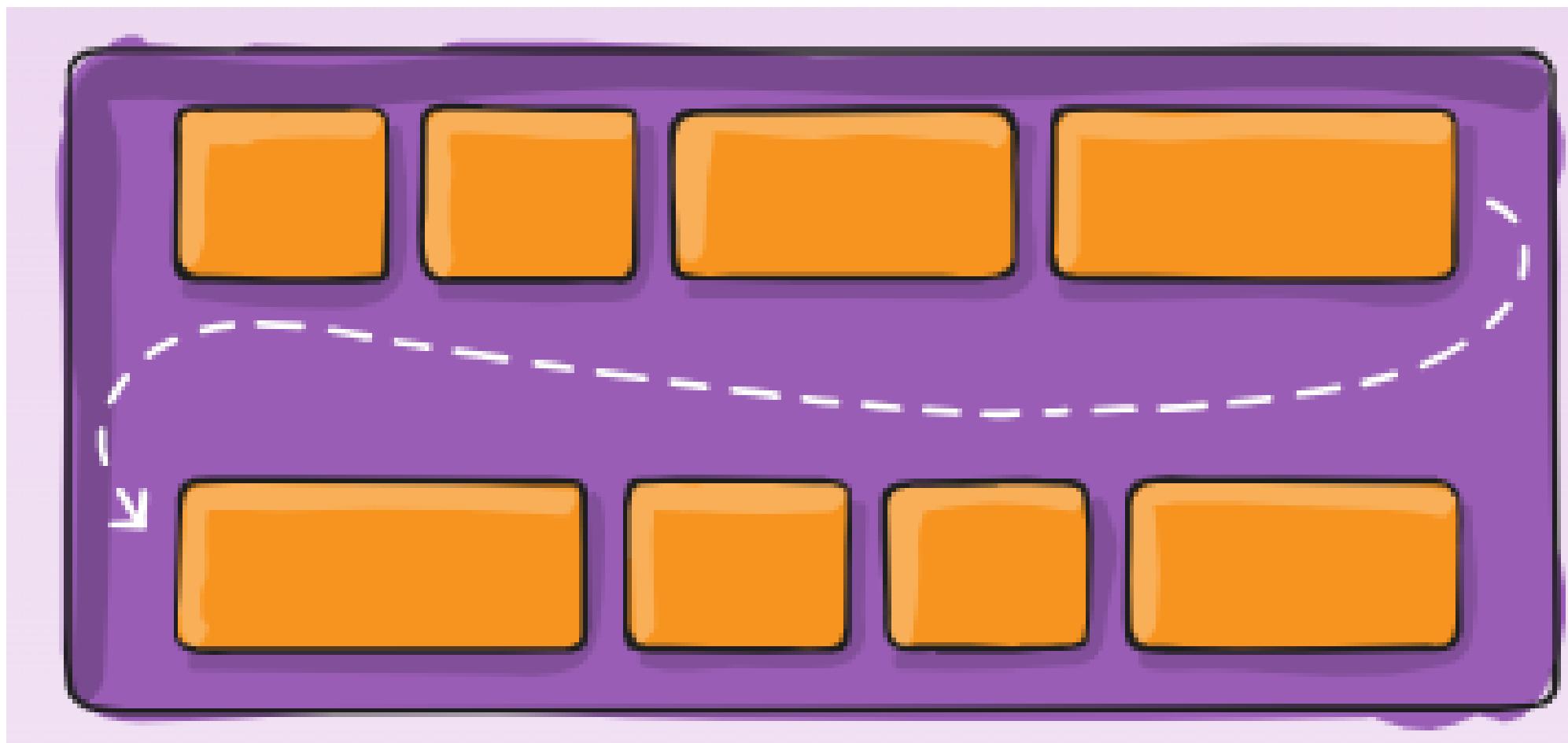
- **row (default)**: linha, no canto superior esquierdo
- **row-reverse**: linha reversa, no canto superior direito
- **column**: em coluna, na parte superior
- **column-reverse**: em coluna reversa, na parte inferior.



{

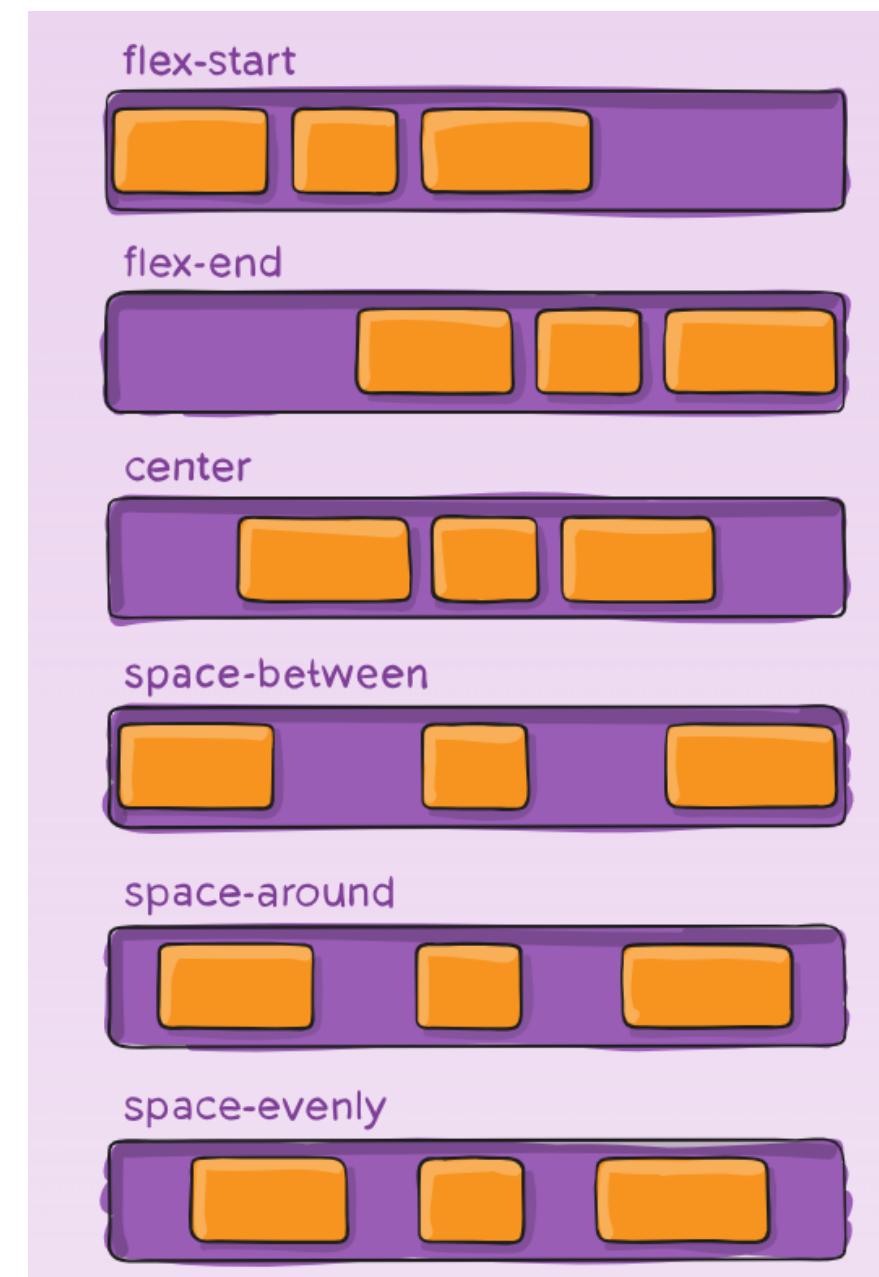
flex-wrap:

- Define se os itens devem ser ajustados em uma única linha ou várias linhas no container flexível.



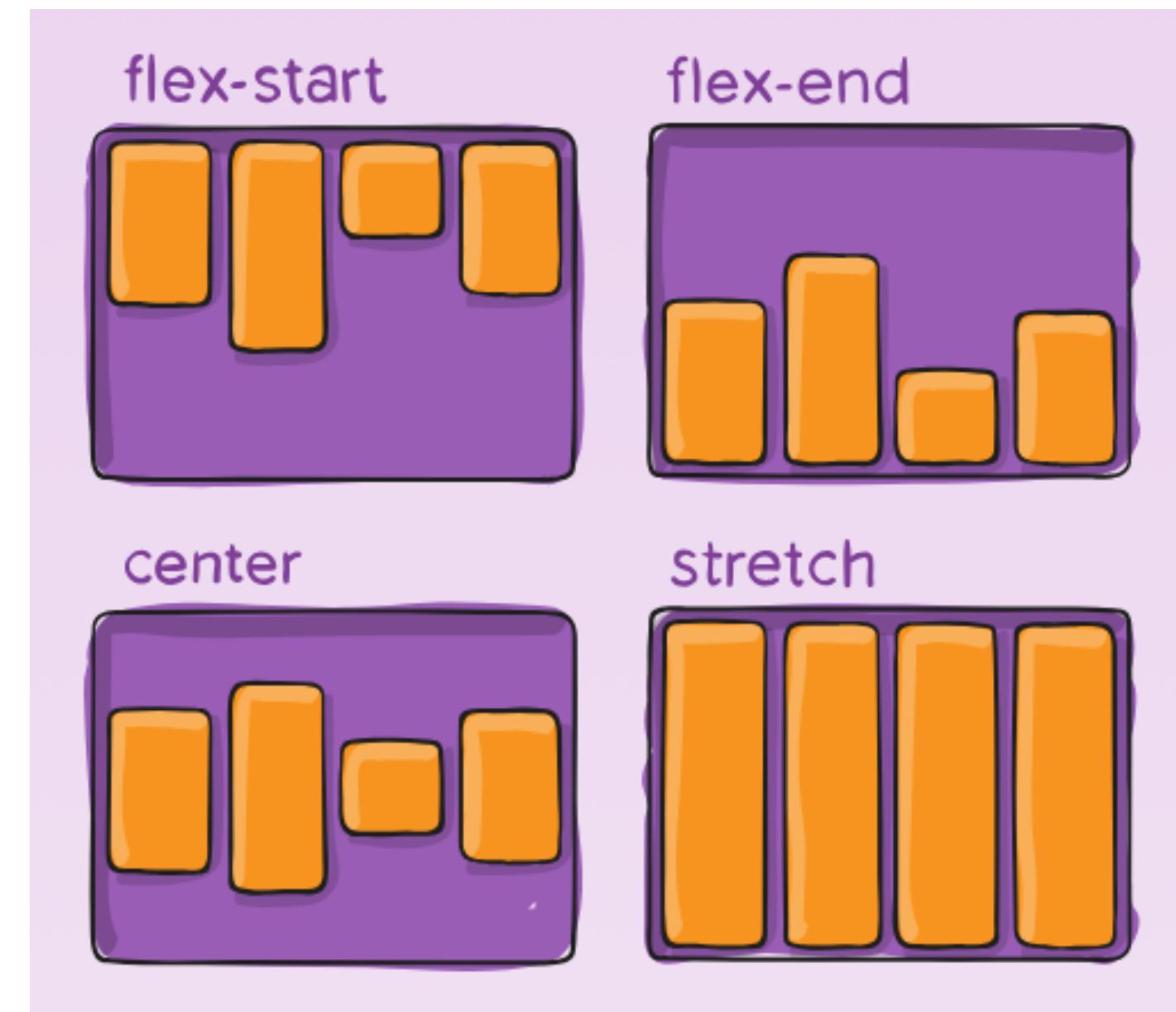
justify-content:

- Alinha os itens ao longo do eixo principal (horizontal) do container flexível.



} align-itens:

- Alinha os itens ao longo do eixo cruzado (vertical) do container flexível.

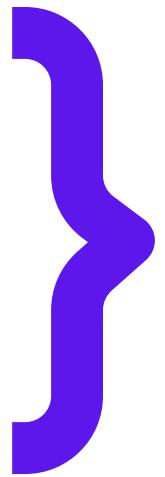


{

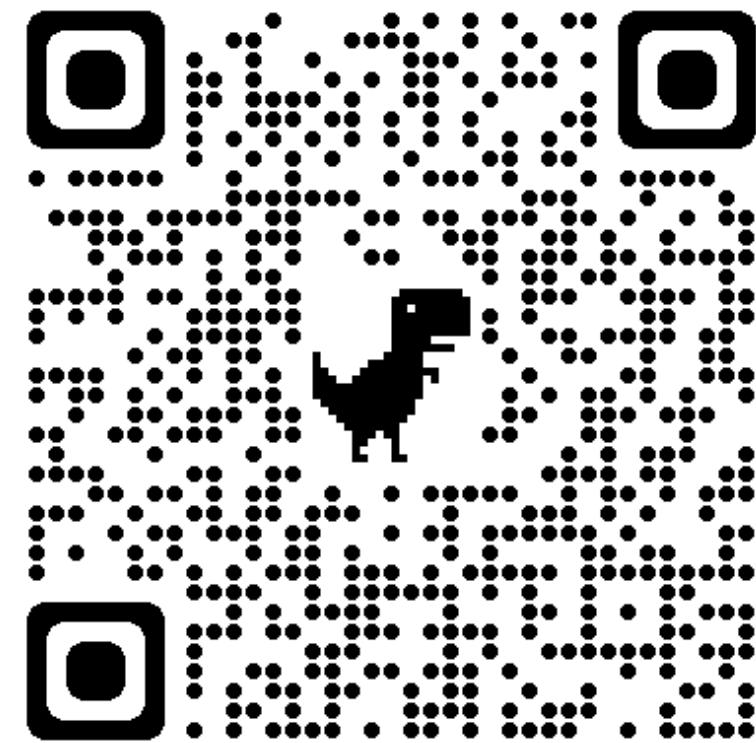
order:

- Controla a ordem de exibição dos itens no container flexível.
- A propriedade aceita valores inteiros, positivos, negativos ou zero. O valor padrão é 0.

```
.container {  
  display: flex;  
}  
  
.item1 {  
  order: 2;  
}  
  
.item2 {  
  order: 1;  
}  
  
.item3 {  
  /* order não definido, então é 0 por padrão */  
}
```



Flex Box



Complete Guide to
Flexbox

Vamos
praticar



FLEXBOX FROGGY

Level 1 of 24

Welcome to Flexbox Froggy, a game where you help Froggy and friends by writing CSS code! Guide this frog to the lily pad on the right by using the **justify-content** property, which aligns items horizontally and accepts the following values:

- **flex-start**: Items align to the left side of the container.
- **flex-end**: Items align to the right side of the container.
- **center**: Items align at the center of the container.
- **space-between**: Items display with equal spacing between them.
- **space-around**: Items display with equal spacing around them.

For example, **justify-content: flex-end;** will move the frog to the right.

```
1 #pond {  
2   display: flex;  
3   [REDACTED]  
4 }  
5  
6  
7  
8  
9  
10
```

Next



Welcome to Flexbox Froggy, a game where you help Froggy and friends by writing CSS code! Guide this frog to the lily pad on the right by using the **justify-content** property, which aligns items horizontally and accepts the following values:

- **flex-start**: Items align to the left side of the container.
- **flex-end**: Items align to the right side of the container.
- **center**: Items align at the center of the container.
- **space-between**: Items display with equal spacing between them.
- **space-around**: Items display with equal spacing around them.

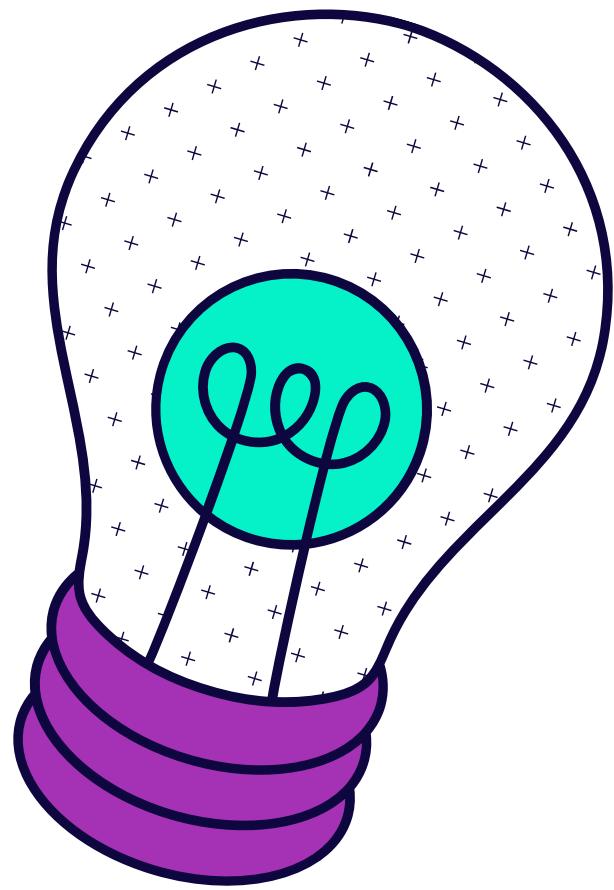
For example, **justify-content: flex-end;** will move the frog to the right.

```
1 #pond {  
2   display: flex;  
3 }  
4  
5  
6  
7  
8  
9  
10
```

Next

search

QUIZ



Flexbox Defense



Wave 1 of 12 ▾



Use the `justify-content` ▾ property to move these two towers into position. Click the ? button in the stylesheet for a reminder on how the property works.

```
1 .tower-group-1 {  
2   display: flex;  
3   |  
4 }  
  
 hide tower inputs
```



Start Wave

Made by Channing Allen



.tower-group-1



Flexbox Defense



Wave 1 of 12 ▾



Use the `justify-content` ▾ property to move these two towers into position. Click the ? button in the stylesheet for a reminder on how the property works.

```
1 .tower-group-1 {  
2   display: flex;  
3   |  
4 }
```



Start Wave

hide tower inputs

Made by Channing Allen



http://www.flexboxdefense.com

.tower-group-1