

# VEM SER

**HTML e CSS**

Aula 02 - CSS

# Conteúdo da aula

Introdução à CSS

Formas de estilizar

Classes vs IDs

Box Model

Display

Pseudoelementos

Pseudoclasses

Google Fonts e Font Awesome

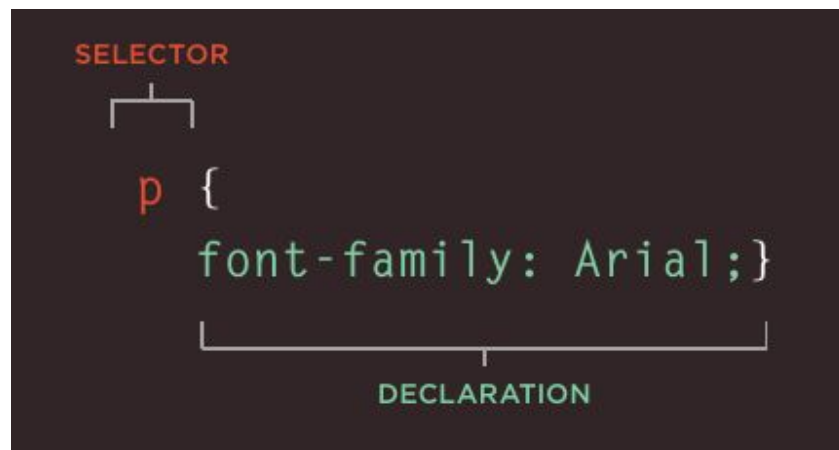
Position



# Introdução à CSS

# CSS

Conforme falamos na aula anterior, o CSS é responsável pela estilização das páginas. Isto acontece através da seleção dos elementos da página e da atribuição de declarações a esses elementos, cujos valores serão aplicados.



# Declarações: Par propriedade e valor

Cada declaração atribuída a um seletor, recebe um par de propriedade e valor. Sendo este valor, o estilo que será aplicado ao elemento.

```
h1, h2, h3 {
    font-family: Arial;
    color: yellow;}
    
```

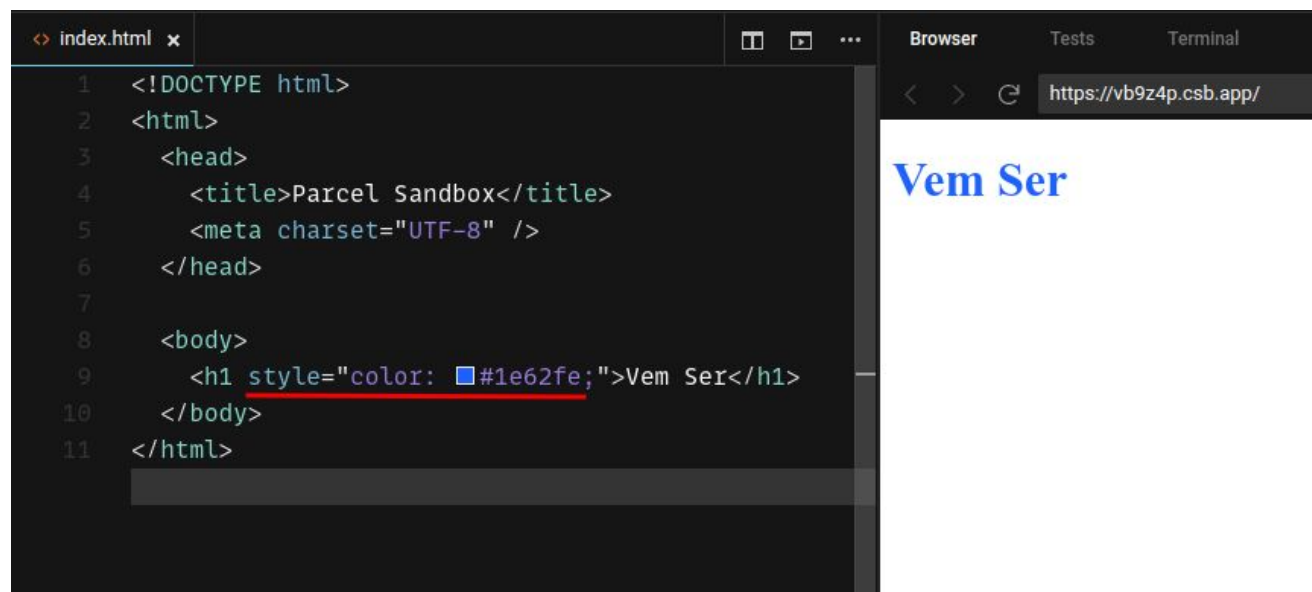




Formas de  
estilizar

# Inline

Como o próprio nome diz, trata-se de uma estilização que ocorre na mesma linha em que o elemento está declarado no HTML. Deve ser usada raramente, pois diminui a legibilidade do código.



```

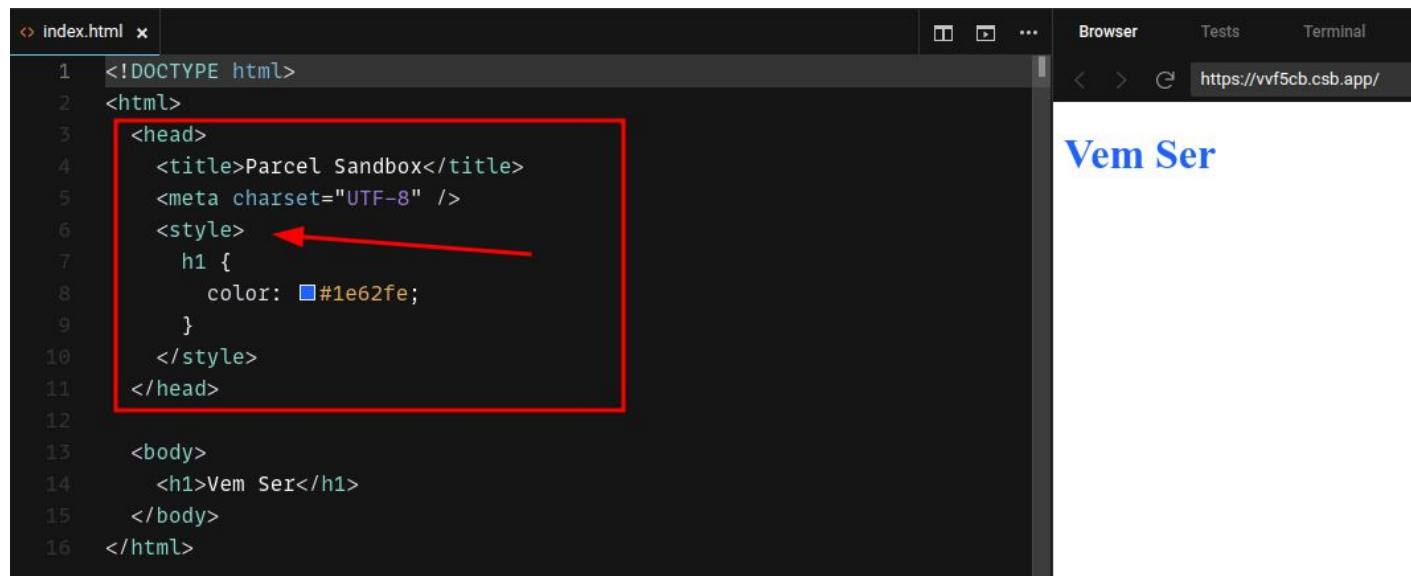
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>Parcel Sandbox</title>
5      <meta charset="UTF-8" />
6    </head>
7
8    <body>
9      <h1 style="color: #1e62fe;">Vem Ser</h1>
10     </body>
11  </html>

```

The browser preview shows the text "Vem Ser" in a blue color, matching the inline style defined in the HTML code.

# Internal / Embedded

Trata-se de uma estilização feita dentro da tag `<style></style>` e dentro do documento HTML.



```

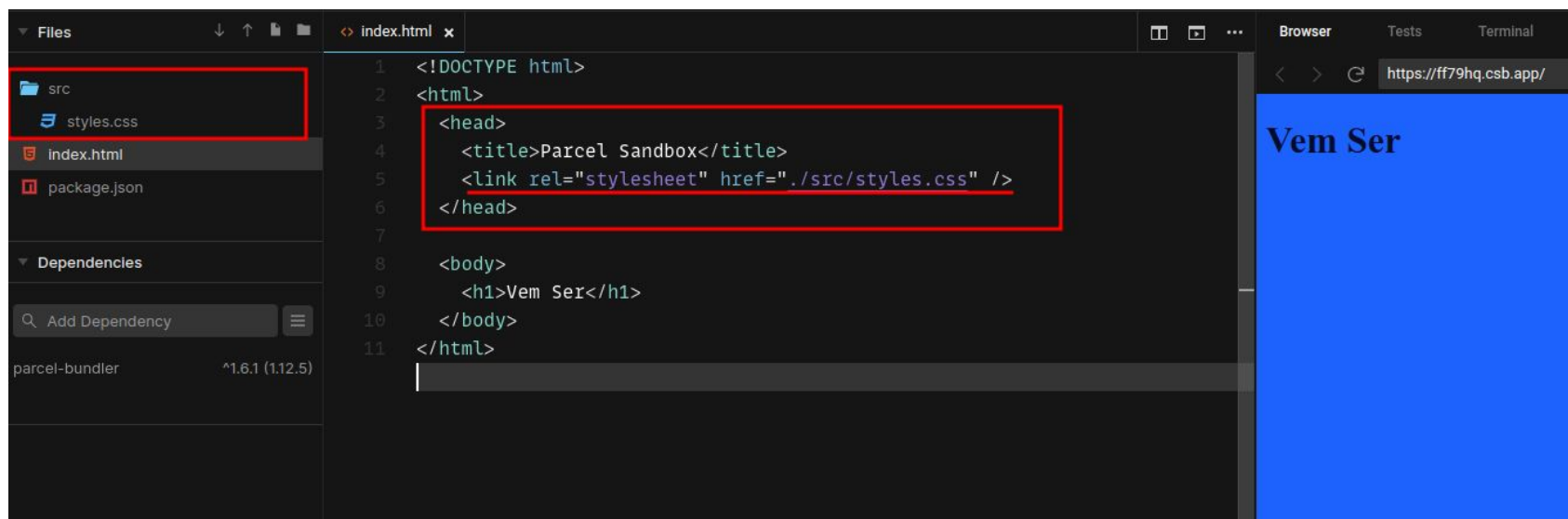
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Parcel Sandbox</title>
5     <meta charset="UTF-8" />
6     <style>
7       h1 {
8         color: #1e62fe;
9       }
10    </style>
11  </head>
12
13  <body>
14    <h1>Vem Ser</h1>
15  </body>
16 </html>
  
```

<https://codesandbox.io/s/css-internal-vvf5cb>



# External

CSS externo é a forma mais comum e mais recomendada de estilizar um site. Deve ser criado um arquivo com extensão .css e no documento HTML deve ser utilizada a tag **<link>** para incluir a estilização.



<https://codesandbox.io/s/html-css-ff79hq>



# Classes vs IDs

# Classes

Classes podem ser reutilizadas, e sua principal função é estilização.

The screenshot shows a code editor with two files: `index.html` and `styles.css`. The `index.html` file contains the following code:

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Parcel Sandbox</title>
5     <meta charset="UTF-8" />
6     <link rel="stylesheet" href="./src/styles.css" />
7   </head>
8
9   <body>
10    <div class="container">
11      <section class="inicio"></section>
12    </div>
13    <div class="container">
14      <section class="contato"></section>
15    </div>
16  </body>
17 </html>

```

The `styles.css` file contains the following code:

```

1 .container {
2   width: 1200px;
3 }
4
5 .inicio {
6   width: 100%;
7   height: 200px;
8   background-color: #1e62fe;
9 }

```

The browser window on the right shows the rendered page at `https://njbgo5.csb.app/`. It displays two solid blue rectangular sections, one above the other, which correspond to the `.inicio` and `.contato` sections defined in the CSS.

<https://codesandbox.io/s/classes-njbgo5>

# IDs

Até podem ser utilizados para estilização, **mas não é recomendado**. IDs costumam ser mais utilizados para identificar elementos e utilizá-los no código JavaScript. Eles **não** podem ser reutilizados, pois são identificadores.

Também é uma boa prática colocarmos IDs nos principais itens da nossa aplicação, para que o time de testes possa rodar os testes mais facilmente.

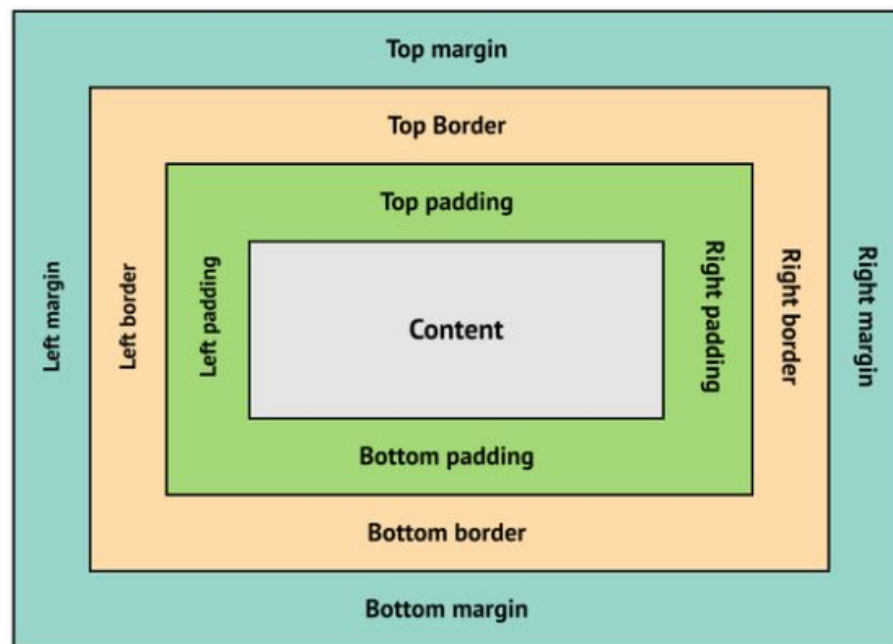


Box Model

# Box Model

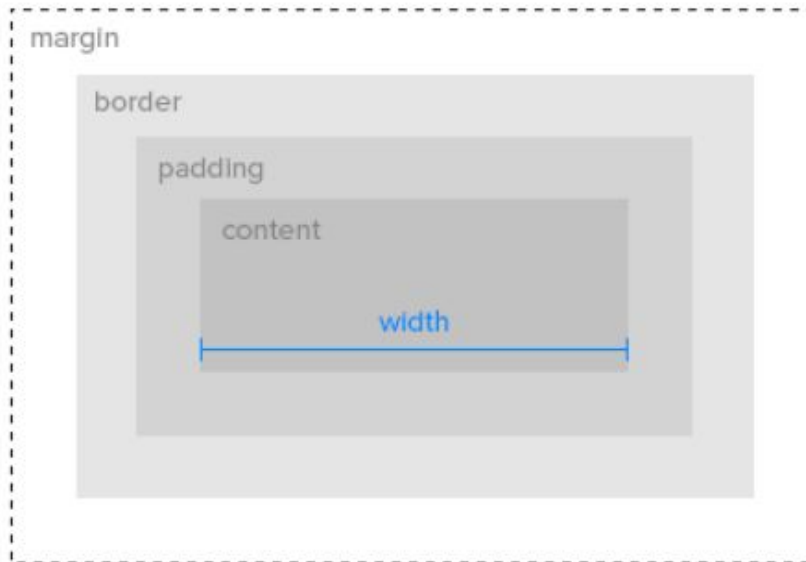
Cada elemento no CSS é uma espécie de “caixa” (*box*) e essa caixa pode ter as seguintes propriedades:

- **Conteúdo** (*content*);
- **Preenchimento** (*padding*);
- **Borda** (*border*);
- **Margem** (*margin*).

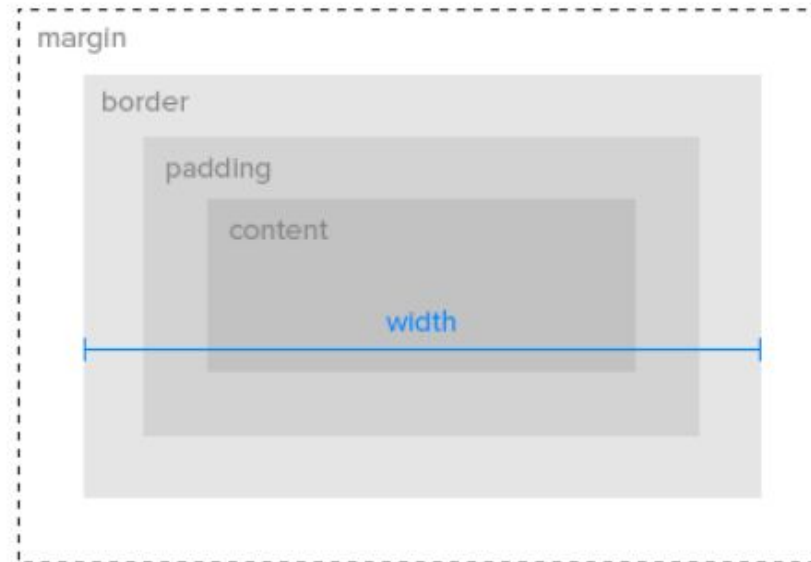


# ***box-sizing***

***content-box***



***border-box***



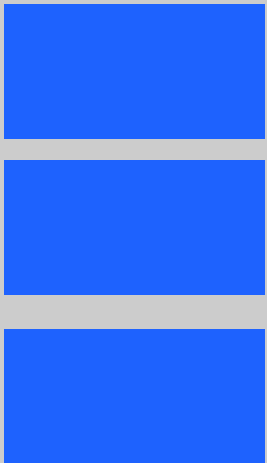
A large, dark blue speech bubble with a white interior, centered on the page. The bubble has a thick border and a small tail pointing towards the bottom left.

*Display*

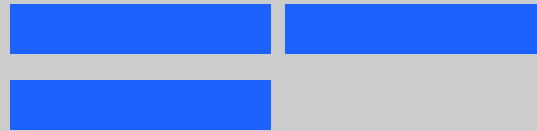


# ***Display***

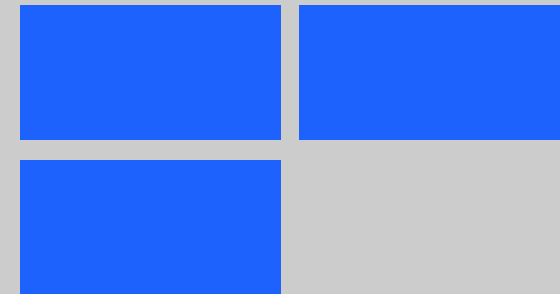
***block***



***inline***



***inline-block***





Pseudo  
elementos

# Pseudoelementos

Um “pseudoelemento” consegue se comportar **quase** que como um elemento real. A forma como adicionamos é através de uma *keyword* após o elemento que se deseja acrescentar o pseudoelemento.

Sintaxe:

```
seletor::pseudo-elemento {  
  propriedade: valor;  
}
```

# Na prática

HTML



```
1 <p>Oi</p>
```

CSS



```
1 p::before {
2   content: "*";
3   color: red;
4 }
```

Resultado

\*Oi



# Pseudoclasses

Diferentemente dos pseudoelementos, as pseudoclasses não se comportam como elementos, mas sim representam um “estado especial” dele.

# Na prática

## HTML

```
1 <button>Clique aqui</button>
```

## CSS

```
1 button {  
2   background-color: blue;  
3   color: white;  
4   font-weight: bold;  
5 }  
6  
7 button:hover {  
8   background-color: red;  
9   cursor: pointer;  
10 }
```

## Na prática

Resultado inicial

Clique aqui

Resultado após estado  
especial: ***hover***

Clique aqui

# Lista de pseudoelementos e pseudoclasses

::after	::before	::first-letter	::first-line
::placeholder	::selection	:active	:blank
:checked	:default	:dir()	:disabled
:empty	:enabled	:first	:first-child
:first-of-type	:focus	:fullscreen	:hover
:in-range	:indeterminate	:invalid	:lang()
:last-child	:last-of-type	:left	:link
:not()	:nth-child()	:nth-last-child()	:nth-last-of-type()
:nth-of-type()	:only-child	:only-of-type	:optional
:out-of-range	:read-only	:read-write	:required
:right	:root	:scope	:target
:valid	:visited	content	





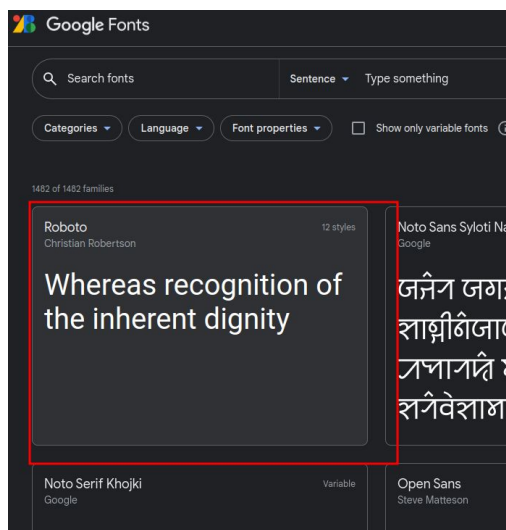
Google  
Fonts

# Google Fonts

O Google Fonts é um CDN de fontes de uso livre. Para usá-las nos nossos projetos, é bem simples.

1 - Acessar: <https://fonts.google.com/>

2 - Escolher a fonte:



# Google Fonts

3 - Selecione os tamanhos 400 - Regular, 400 - Itálico e 700 - Bold:

Regular 400

Select Regular 400 ↕

Regular 400 Italic

Select Regular 400 Italic ↕

Medium 500

Select Medium 500 ↕

Medium 500 Italic

Select Medium 500 Italic ↕

Bold 700

Select Bold 700 ↕

Bold 700 Italic

# Google Fonts

4 - Copie a tag **<link />** e cole dentro da tag **<head>** do HTML (ou copie o **@import** e cole dentro do CSS, tanto faz):

Use on the web

To embed a font, copy the code into the  
<head> of your html

☒ <link>    ☐ @import

```
<link rel="preconnect" href="http
s://fonts.googleapis.com">
<link rel="preconnect" href="http
s://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapi
s.com/css2?family=Roboto:ital,wght@
0,400;0,700;1,400&display=swap" rel
="stylesheet">
```



# Google Fonts

5 - Por último, mude a *font-family* para a seleccionada no CSS:

CSS rules to specify families

```
font-family: 'Roboto', sans-serif;
```





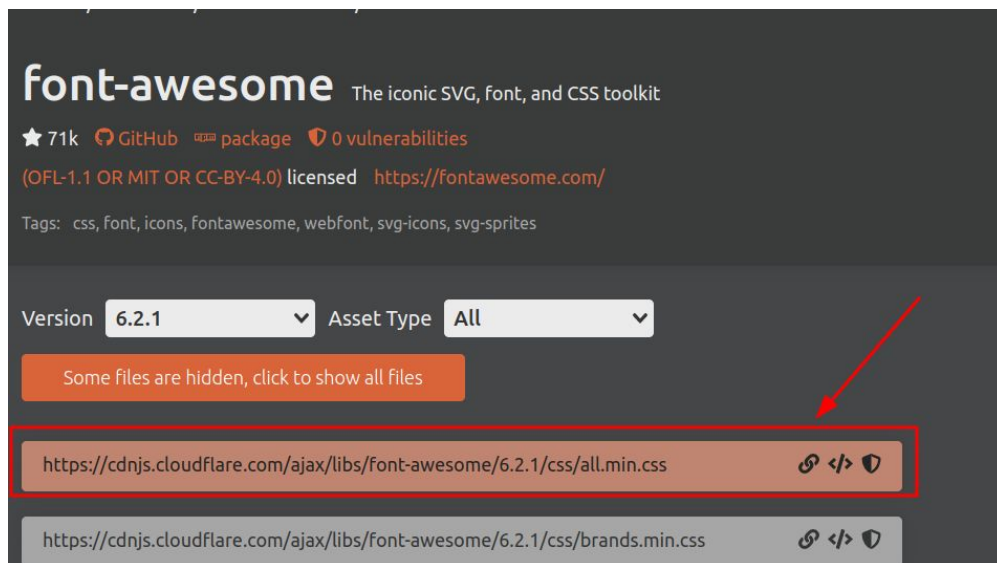
Font  
Awesome

# Font Awesome

O **Font Awesome** é um recurso que nos permite adicionar ícones ao nosso documento HTML através de uma fonte específica para isso, sem necessidade de baixar imagens. Para utilizarmos, é bem simples.

1- Vá até o CDN do Font Awesome: <https://cdnjs.com/libraries/font-awesome/>

2 - Copie o primeiro link:



# Font Awesome

3 - Cole o link como `src` de uma tag `<link />` dentro da `<head></head>`:

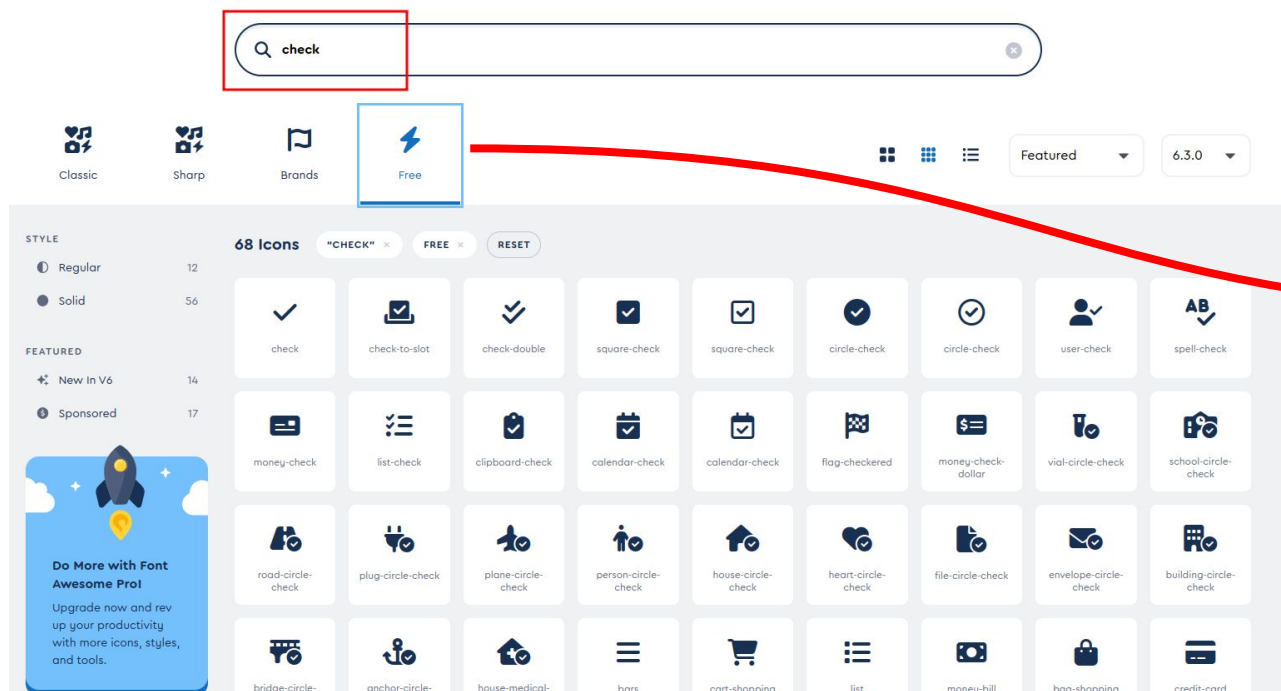


```
1 <link
2   rel="stylesheet"
3   href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.2.1/css/all.min.css"
4 />
```



# Font Awesome

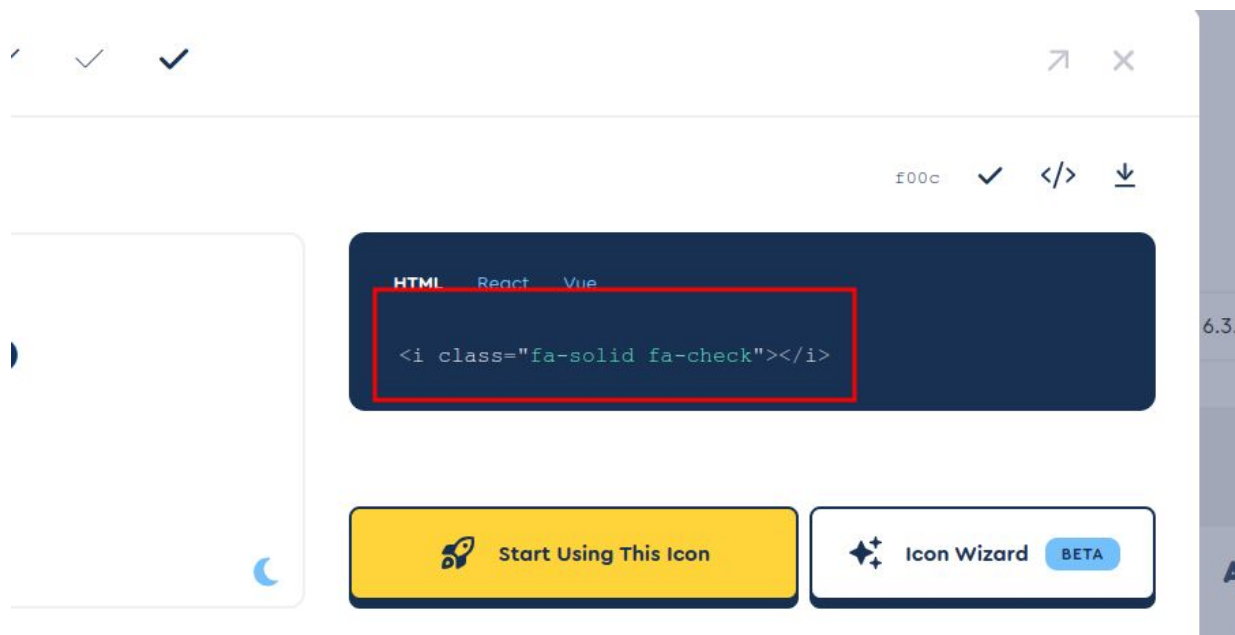
4 - Procure pelo ícone que deseja adicionar ao documento no link <https://fontawesome.com/icons>



Não esqueça de selecionar os ícones que são de uso gratuito. Caso contrário, não irá funcionar!

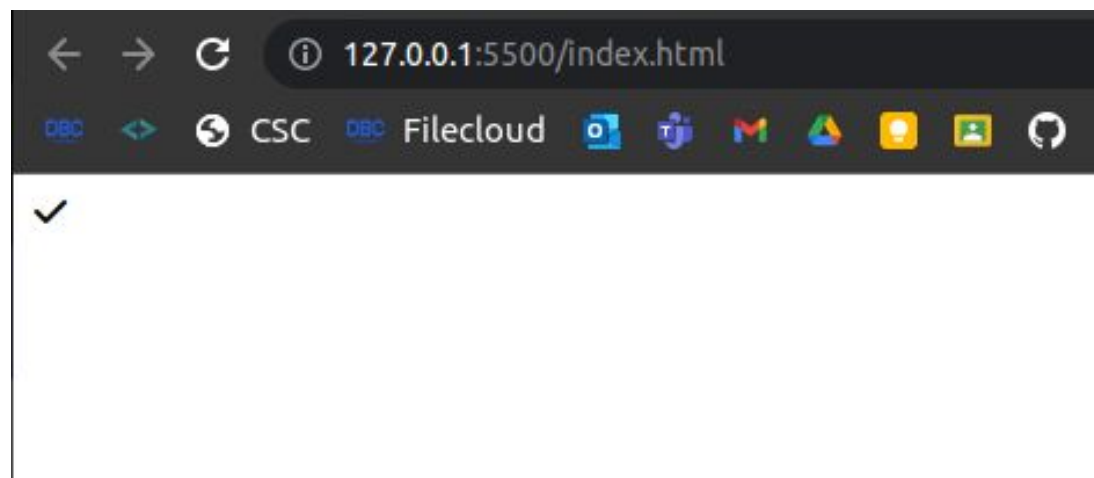
# Font Awesome

5 - Clique no ícone, selecione o código HTML dele e cole no documento HTML:



# *Font Awesome*

Resultado:





*Position*

# ***Position***

O *position* é uma propriedade através da qual definimos como um elemento será posicionado (renderizado) na página.

Tipos de *position*:

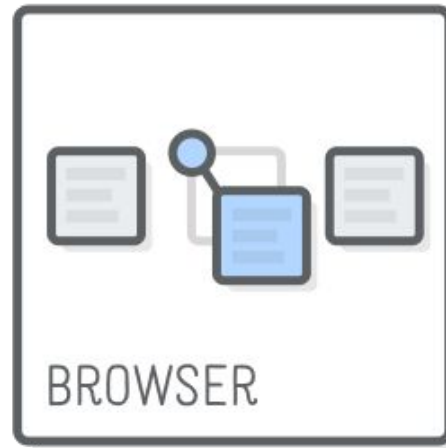
- *static;*
- *relative;*
- *fixed;*
- *absolute;*
- *sticky.*

# ***Position***

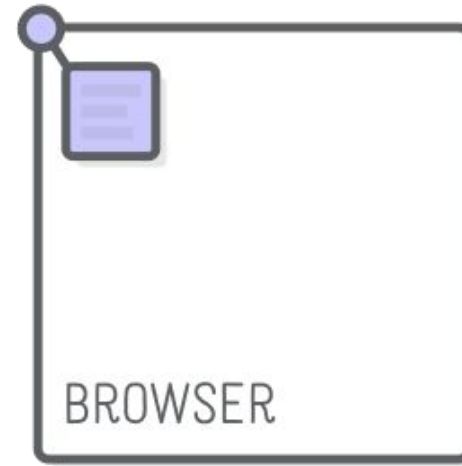
## **STATIC**



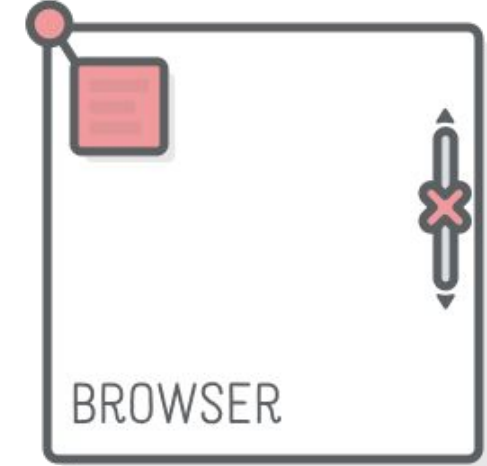
## **RELATIVE**



## **ABSOLUTE**



## **FIXED**



# Exemplo de uso de *absolute*

Ícones de notificação de mensagens não lidas



# Recomendação de leitura

Convenções de nomenclatura de classes no CSS:

<https://tableless.com.br/oocss-smacss-bem-dry-css-afinal-como-escrever-css/>





Let's *Tech Up Together*