



# SHIFT



FIAP





# FRONT-END JOURNEY

DESENVOLVIMENTO WEB COM ANGULAR & REACT



# FRONT: **PROGRAMAÇÃO**

HTML - CSS - BOOTSTRAP - SASS





## ISRAEL MARQUES JÚNIOR

PROFESSOR

---

- Israel é pós-graduado em Engenharia Web e trabalha com educação há 27 anos. Trabalhou no desenvolvimento de sistemas para desktop e, em seguida, migrou para a criação de aplicações para a Internet.
- Na FIAP, é professor nos cursos de: Sistemas de Informação, Análise e Desenvolvimento de Sistemas, Sistemas para Internet e Jogos Digitais para as disciplinas focadas em front end.

✉ [profisrael.copi@fiap.com.br](mailto:profisrael.copi@fiap.com.br)



# AGENDA

---

1

## AULA 1

CONTEÚDO – Baixando o Editor + Introdução HTML + CSS

2

## AULA 2

CONTEÚDO – Listas + Links + Imagens + Divs + Posicionamento

3

## AULA 3

CONTEÚDO – Semântica – Background – Flexbox

4

## AULA 4

CONTEÚDO – Tabelas – Formulários

5

## AULA 5

CONTEÚDO – Design Responsivo + Media Queries + Mobile First

# AGENDA

---

6

## AULA 6

CONTEÚDO – Bootstrap: Introdução + Containers + Utilitários

7

## AULA 7

CONTEÚDO – Bootstrap: Grid + NavBar + Cards + Carousel

8

## AULA 8

CONTEÚDO – Bootstrap: Accordion + Modal + Formulários

9

## AULA 9

CONTEÚDO – Sass: Introdução + Instalação + Conceitos Iniciais

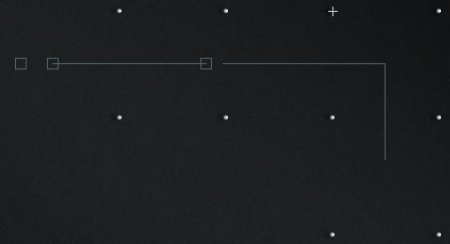
10

## AULA 10

CONTEÚDO – Sass: Estilização + Variáveis + Funcionalidades

## AULA 3

# SEMÂNTICA - FLEXBOX









# BACKGROUNDS





















# FLEXBOX







# Display flex

Para usarmos o Flexbox, precisamos transformar o elemento pai em um flex container, para isso aplicaremos nele a propriedade **display** com o valor **flex**.

Por padrão, os filhos desse container ficarão alinhados um ao lado do outro.

Os elementos filhos serão definidos como flex-item.

```
div {  
  
    display: flex;  
  
}
```

# Flex-direction

Define a direção que os flex itens devem seguir no container pai. Possui os seguintes valores válidos:

**Row:** os flex-items ficarão um ao lado do outro.

+ **Column:** os flex-items ficarão um abaixo do outro.

**Row-reverse:** os flex-items ficarão alinhados a partir da direita do container pai.

**Column-reverse:** os flex-items ficarão um abaixo do outro, formando uma única coluna, mas em ordem inversa.

```
.box{
```

```
flex-direction: row;
```

```
}
```

## Flex-wrap

A propriedade **flex-wrap** controla a quebra de linhas quando necessário. Possui as seguintes opções:

**Nowrap:** não permitirá a quebra de linhas, esse é o valor padrão e não precisará ser declarado.

**Wrap:** permite a quebra de linhas quando os flex itens não couberem na mesma linha do container.

**Wrap-reverse:** permite a quebra de linha iniciando o posicionamento dos elementos da parte inferior para a parte superior do container.

```
.box{  
  
    flex-wrap: wrap;  
  
}
```



## Align-items

Faz o alinhamento vertical dos flex-items no container pai. Possui as seguintes opções:

**Flex-start:** alinhará os containers verticalmente no início do elemento pai.

+ **Flex-end:** alinhará os containers verticalmente no fim do elemento pai.

**Center:** alinhará os flex itens verticalmente.

**Stretch:** deixará os flex itens com a mesma altura do elemento pai.

**Baseline:** a altura dos flex itens será definida pela quantidade de conteúdo que eles receberem.

```
.box{  
  
    align-items: center;  
  
}
```



## Flex-grow

Essa propriedade deverá ser aplicada apenas aos flex-items permitindo que eles tenham larguras diferentes. O valor padrão é zero, eles ocuparão o tamanho máximo baseado no conteúdo ou na largura definida na regra CSS.

Se você defini-los com valor 1, eles tentarão ter a mesma largura. Caso algum deles tiver um conteúdo muito grande, a largura do elemento será preservada.

```
.container {  
  
    display: flex;  
  
    align-items: baseline; }  
  
.box-um { flex-grow:2; }  
  
.box-dois { flex-grow:1; }  
  
.box-tres { flex-grow:1; }  
  
.box-quatro { flex-grow:2; }
```

# Flex-basis

Essa propriedade também só deve ser aplicada aos flex-items, ela define um valor fixo para a largura de um container, isso acontecerá caso exista espaço suficiente. Os demais elementos terão a sua largura calculada pelo espaço restante do container pai. Podemos usar em conjunto com a propriedade flex-grow.

```
.container {  
  
    display: flex;  
  
    align-items: baseline; }  
  
.box-um { flex-basis:90px; }  
  
.box-dois { flex-grow:1; }  
  
.box-tres { flex-basis:550px; }  
  
.box-quatro { flex-grow:2; }
```

## Order

Também aplicado apenas aos flex-items, ela define a ordem como os elementos devem aparecer na tela. É uma opção legal para reorganizar conteúdo sem ter de alterar o código HTML.

```
.container {  
  
    display: flex;  
  
    align-items: baseline; }  
  
.box-um { order:3; }  
  
.box-dois{ order:4; }  
  
.box-tres{ order:1; }  
  
.box-quatro{ order:2; }
```

# CONTAINERS SEMÂNTICOS

## Containers Semânticos

Você deve ter percebido que uma das tags que mais utilizamos em nossos códigos é a **<div>**, é bem simples explicar o porquê, ela é uma tag de container onde é possível armazenar conteúdos diversos da nossa página.

Tudo bem usar a `<div>` como container, mas as vezes temos tantas divs em nosso código que na hora de fazer uma pequena alteração o desenvolvedor precisará de um tempo para entender como o HTML foi estruturado, praticamente tudo fica dentro de `<div>` e, muitas vezes, divs dentro de outras divs.



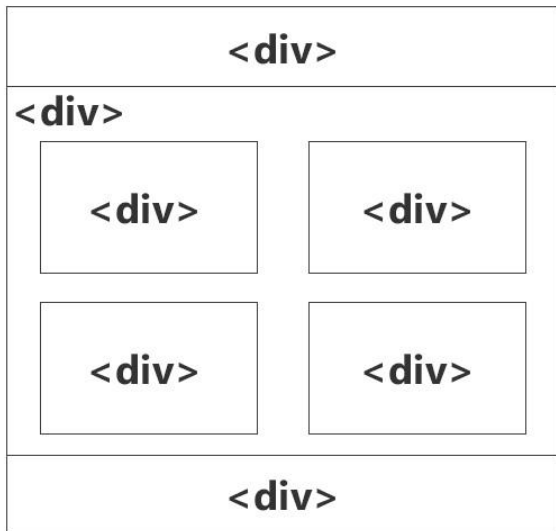
## Containers Semânticos

Mas como podemos melhorar o código HTML para que ele possa ter um pouco mais de sentido, ficando assim mais fácil de ser entendido? A resposta para essa pergunta é a semântica.

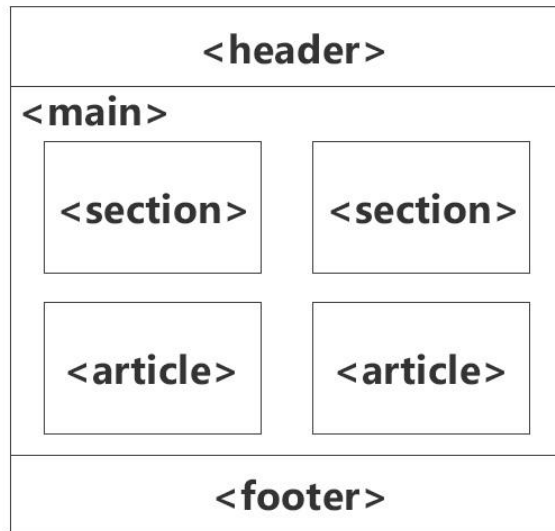
A ideia é usar tags semânticas, elas são containers com as mesmas características das **divs**, só que agora elas possuem uma relevância que não existia antes, trazendo assim grandes benefícios.

# Containers Semânticos

## Estrutura com divs



## Estrutura semântica



# Header

Define o cabeçalho para a página ou para outros containers, indicando assim o início de um elemento ou seção. É muito usado para a inserção de logotipos, áreas de navegação e campos para busca.

Em um mesmo documento podemos ter vários elementos `<header>`.

```
<header>
```

```
<h1> Lorem Ipsum </h1>
```

```

```

```
</header>
```

Main

Define o container que receberá o conteúdo principal da sua página. Essa tag deverá ser única, ou seja, não deveremos ter mais de um elemento <main> no mesmo documento HTML.

```
<main>
```

```
<h1> Lorem Ipsum </h1>
```

```

```

```
</main>
```

## Section

Define uma ou mais seções em nossa página. Uma seção deve abordar algum determinado conteúdo ou assunto, e sempre possuir um título.

Em um mesmo documento podemos ter vários elementos `<section>`, mas tome cuidado para não usar esse elemento como um container mais genérico, para isso usamos a tag `<div>`.

```
<section>
```

```
<h2> Lorem Ipsum </h2>
```

```
<p> Lorem Ipsum... </p>
```

```
</section>
```



## Article

Define um container de conteúdo mais específico,  
independente e que pode ser reutilizável em outras  
páginas sem perder o seu contexto.

Em um mesmo documento podemos ter vários  
elementos `<article>`.

```
<article>
```

```
<h2> Lorem Ipsum </h2>
```

```
<p> Lorem Ipsum... </p>
```

```
</article>
```

## Aside

Define um container de conteúdo separado, mas vinculado a um determinado assunto ou conteúdo. Na maioria das vezes, esse elemento é posicionado ao lado do conteúdo que ele foi associado, como uma barra lateral.

Em um mesmo documento podemos ter vários elementos `<aside>`.

```
<aside>
```

```
<a href="#"> Link </a>
```

```
<a href="#"> Link </a>
```

```
<a href="#"> Link </a>
```

```
</aside>
```

## Nav

Define um container que deve agrupar os links criados pela tag <a>. Não é necessário que todos os links estejam dentro de um container <nav>, apenas aqueles que formarem o sistema de navegação de sua aplicação.

A tag <nav> é comumente encontrada em elementos <header>, mas também podem ser inseridos em outros containers existentes na página.

Em um mesmo documento podemos ter vários elementos <nav>.

```
<nav>
```

```
<ul>
```

```
<li><a href="#">Link</a></li>
```

```
<li><a href="#">Link</a></li>
```

```
<li><a href="#">Link</a></li>
```

```
</ul>
```

```
</nav>
```

## Figure - Figcaption

Define um container que deve agrupar imagens existentes na página.

Podemos utilizar a tag `<figcaption>` que definirá uma legenda a ser usada por um grupo de imagens.

Também é possível utilizar em cada imagem um `<figcaption>` diferente.

```
<figure>
```

```

```

```

```

```

```

```
<figcaption> Fotos da página </figcaption>
```

```
</figure>
```

## Footer

| Define o rodapé para a página ou para outros containers,

+ indicando assim o final de um elemento ou seção.

Em um mesmo documento podemos ter vários elementos `<footer>`.

```
<footer>
```

```
<p>Todos os direitos reservados</p>
```

```
</footer>
```





# Font Awesome



## Font Awesome

Podemos fazer o uso de ícones em nossas páginas. Para isso, usaremos o <https://www.fontawesome.com>, um conjunto de ícones que podem ser adicionados a sua aplicação.



**Font Awesome**

Get vector icons and social logos on your website with Font Awesome, the web's most popular icon set and toolkit.



**Start for Free**

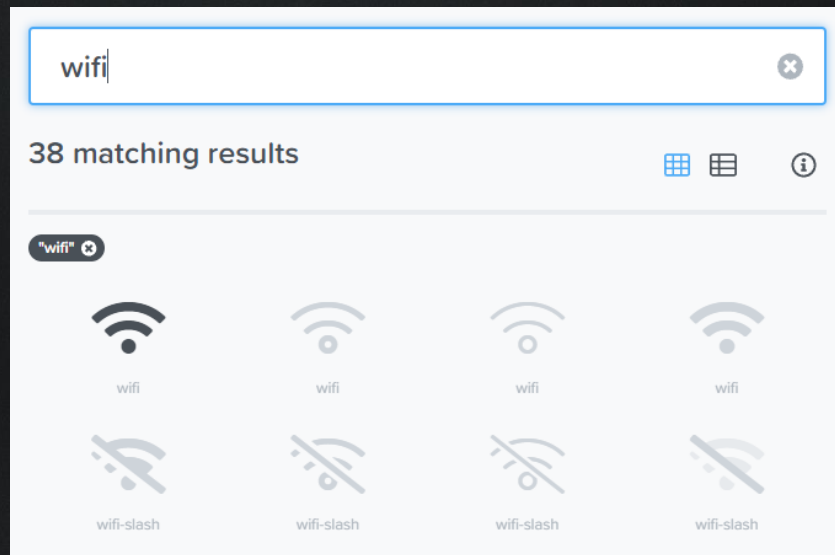


**Get More with Pro**

## Font Awesome

Para buscar algum ícone basta fazer a pesquisa usando a barra posicionada na parte esquerda do menu principal.

Vamos fazer uma busca usando a palavra **wifi**, ela retornará os ícones encontrados, basta clicar na imagem desejada para podermos ter acesso ao seu link.

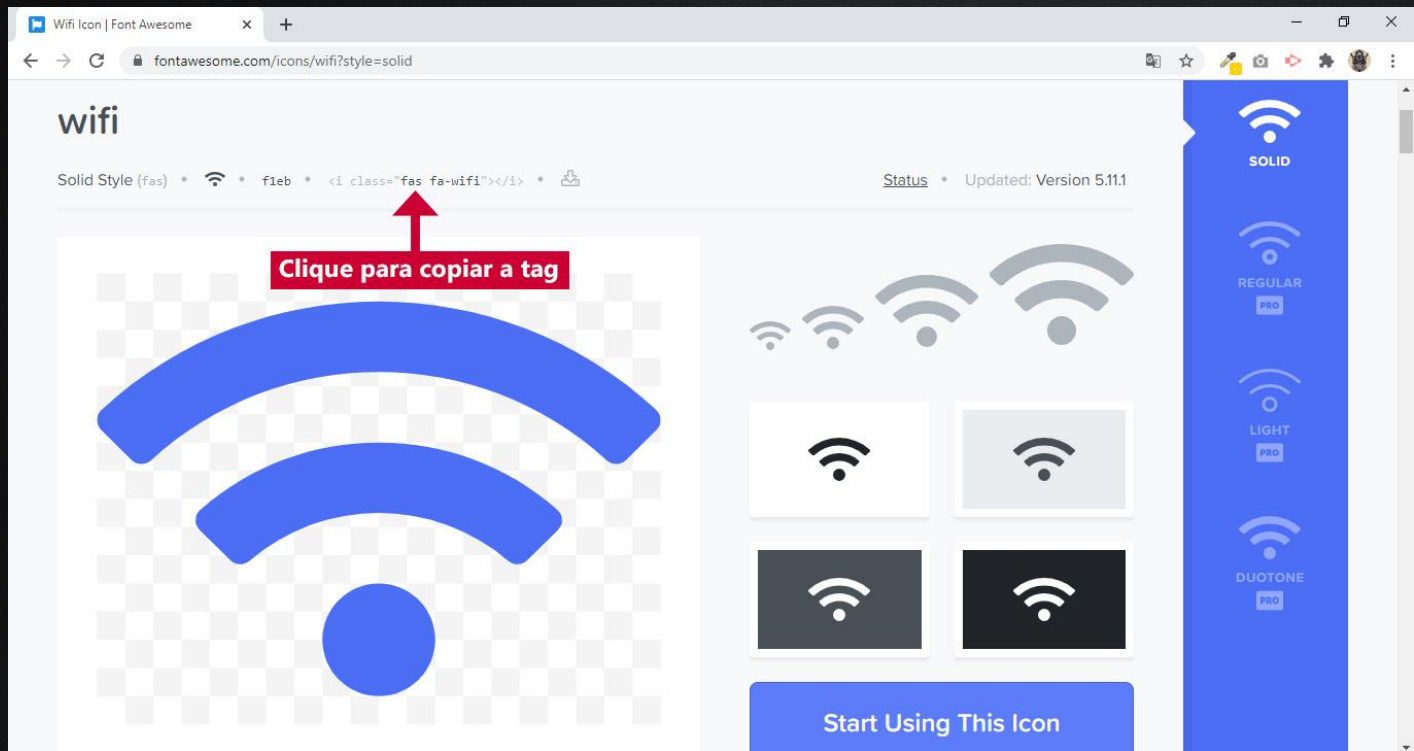


## Font Awesome

Quando você clicar no ícone escolhido, será redirecionado para uma tela onde ele aparecerá bem grande. Na parte superior teremos a tag html `<i>` contendo as classes que formam o ícone, basta clicar nessa tag e ela será automaticamente copiada para a área de transferência da memória do seu computador.

Vá até seu código HTML e cole o conteúdo na área que deseja inserir os ícones.

# Font Awesome





## Font Awesome

A tag `<i>` permite a um texto ficar no estilo itálico, as letras ficam inclinadas. Era uma tag que usávamos nos primórdios da Internet, antes do surgimento das boas práticas quando era comum misturar conteúdo e formatação.

No caso da Font Awesome, eles usam essa tag como um container para as classes que irão exibir os ícones. Não insira conteúdo nesse elemento.

## Font Awesome

```
<i class="fas fa-wifi" > </i>
```



classes para inserção do ícone

## Font Awesome

Para que os ícones sejam visíveis na página, precisamos de alguém que entenda essas classes que estão na tag `<i>`. No próprio site da Font Awesome, você pode fazer seu cadastro, ele enviará um link com o endereço de algum servidor que disponibiliza esse serviço.

Caso você não queira se cadastrar, pode utilizar <https://cdnjs.com/libraries/font-awesome>, ele também disponibilizará o endereço de algum servidor com esse serviço.

# Font Awesome

The screenshot shows the cdnjs.com website for the font-awesome library. The page includes a search bar, navigation links, and a list of available CSS files. A red annotation box with the text "clique para copiar o link" and a red arrow points to the first link: `https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.14.0/css/all.min.css`. The second link is `https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.14.0/css/brands.min.css`. The page also displays the library's version (5.14.0), asset type (All), and a note that some files are hidden.

font-awesome - Libraries - cdnjs

Search from 3.921 libraries on cdnjs...

Home / Libraries / font-awesome

## font-awesome

The iconic SVG, font, and CSS toolkit

★ 64k GitHub package 0 vulnerabilities

(OFL-1.1 OR MIT OR CC-BY-4.0) licensed <https://fontawesome.com/>

Tags: css, font, icons, fontawesome, webfont, svg-icons, svg-sprites

Version 5.14.0 Asset Type All

Some files are hidden, click to show all files

<https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.14.0/css/all.min.css>

<https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.14.0/css/brands.min.css>

clique para copiar o link





# OBRIGADO



/icajai

## FIAP

Copyright © 2018 | Professor Israel Marques Cajai Júnior

Todos os direitos reservados. A reprodução ou divulgação total ou parcial deste documento é expressamente proibida sem o consentimento formal, por escrito, do professor/autor.

