

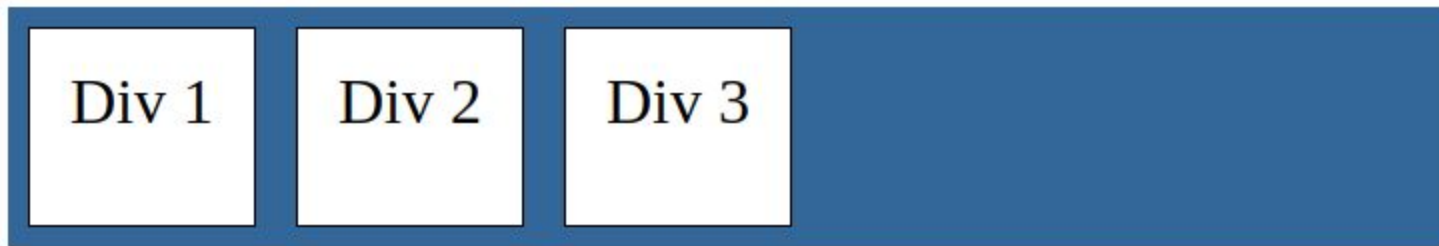
Cascading Style Sheets

Flexbox, Icon Fonts, Sass e SCSS

Flexible Box Layout Module

Flexbox - configuração do pai (flexbox1.html)

O Flexbox facilita a criação de uma estrutura de layout flexível e responsiva sem o uso das propriedades *float* ou *position*. **Regra geral:** o elemento pai controla o posicionamento dos elementos filhos.

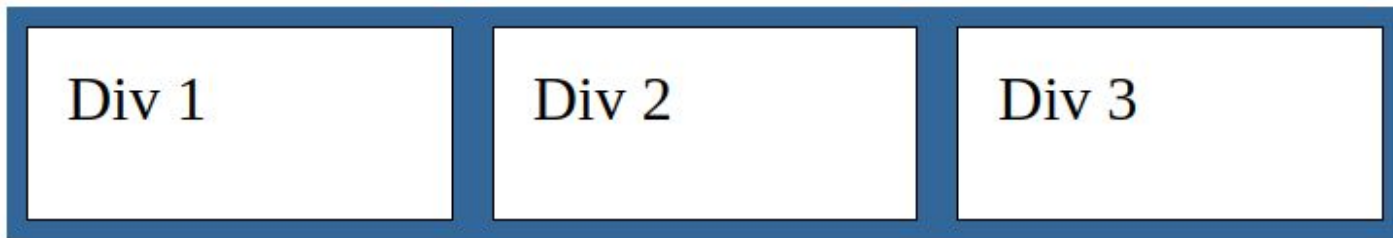


```
div.pai {  
  display: flex;  
}
```

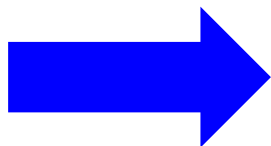
Essa simples configuração terá um efeito semelhante ao *float: left* (aplicado aos filhos).

Flexbox - configuração dos filhos (flexbox2.html)

Veja neste exemplo como faríamos atribuindo um % ao *width* dos filhos e a alternativa indicada pelo Flexbox.



```
div.pai > div {  
  width: 50%;  
}
```



```
div.pai > div {  
  flex: 1;  
}
```

Flexbox - configuração dos filhos (flexbox3.html)

flex: 0 1
50px;

flex-grow: 0;	Define quanto um filho crescerá em relação ao restante dos filhos. O valor 1 significa que ele terá uma relação de 1 para 1. O valor 2 significa que ele será 2x maior que os demais. Valor default: 0
flex-shrink: 1;	Define quanto um filho encolherá em relação ao resto dos filhos. Neste caso, o valor 1 significa que o filho vai encolher na mesma proporção que os outros filhos. Um valor 0 indica que o filho não deve encolher na mesma proporção dos demais. Valor default: 1
flex-basis: 50px;	Define o comprimento inicial de um filho.

Flexbox - configuração do pai (flexbox4.html)

A partir do elemento pai nós podemos influenciar o posicionamento dos elementos filhos. Isso é feito com a propriedade *flex-direction*.

<code>flex-direction: column;</code>	Empilha os filhos verticalmente de cima para baixo.
<code>flex-direction: column-reverse;</code>	Empilha os filhos verticalmente de baixo para cima.
<code>flex-direction: row;</code>	Posiciona os filhos horizontalmente da esquerda para a direita.
<code>flex-direction: row-reverse;</code>	Posiciona os filhos horizontalmente da direita para a esquerda.

Flexbox - configuração do pai (flexbox5.html)

Com o *flex-wrap* é possível especificar se os elementos vão quebrar (pular de linha caso estejam no limite do *width*) ou não.

<code>flex-wrap: wrap;</code>	Indica que os filhos vão pular para a próxima linha caso estejam em seus limites mínimos. Dica: ajuda a criar layouts fluídos.
<code>flex-wrap: nowrap;</code>	Indica que os filhos não vão quebrar. Neste caso o limite mínimo será reduzido. Tem um comportamento semelhante ao de uma tabela (<code>display: table-cell</code>). Este é o valor default.
<code>flex-wrap: wrap-reverse;</code>	Semelhante ao <i>wrap</i> , porém os elementos serão quebrados na ordem inversa.

Flexbox - configuração do pai (flexbox6.html)

O *justify-content* alinha os elementos filhos **horizontalmente**.

<code>justify-content: center;</code>	Alinha os filhos no centro.
<code>justify-content: flex-start;</code>	Alinha os filhos à esquerda (início da tag pai). Este é o valor default.
<code>justify-content: flex-end;</code>	Alinha os filhos à direita (fim da tag pai).
<code>justify-content: space-around;</code>	Define um espaço equivalente ao redor dos filhos, inclusive entre a borda do pai e o primeiro e último filhos.
<code>justify-content: space-between;</code>	Define um espaço equivalente entre os filhos. Neste caso o primeiro e último filhos ficam próximos da borda do pai.

Flexbox - configuração do pai (flexbox7.html)

O *align-items* alinha os filhos **verticalmente**.

<code>align-items: center;</code>	Alinha os filhos no meio do pai.
<code>align-items: flex-start;</code>	Alinha os filhos no topo do pai.
<code>align-items: flex-end;</code>	Alinha os filhos na parte de baixo do pai.
<code>align-items: stretch;</code>	Estende os filhos para preencher todo o <i>height</i> do elemento pai. Este é o valor padrão.

Exercício (exercicio1.html)

Usando Flexbox, crie uma página com um `<main>` ocupando toda viewport, estilizado com um radial-gradient (de amarelo para vermelho) e uma imagem de um sol no centro da tela. O sol deve ficar girando indefinidamente.

Icons Fonts

Icons Fonts

Até agora nós trabalhamos com ícones baixados da Internet e colocados diretamente no HTML. Uma alternativa (bem aceita no desenvolvimento de aplicativos) é o uso de fontes com ícones.

Uma fonte com ícones é equivalente a uma fonte normal, porém no lugar dos caracteres, são exibidos ícones adequados para uma aplicação web.



Font Awesome Icons (fontes1.html)

Site: <https://fontawesome.com/>

Exemplos: <https://fontawesome.com/how-to-use/svg-with-js#basic-use>

Passo 1: importe o CSS do Font Awesome

Passo 2: escolha seus ícones no site

Passo 3: utilize-os com a tag `<i>` especificada com a classe do ícone mais um prefixo *fas fa-* para os ícones normais e *fab fa-* para os ícones de marcas (facebook, amazon, etc).

```
<i class="fas fa-balance-scale"></i>
```

Download Font Awesome Icons (fontes2.html)

Outra opção é **baixar** os arquivos CSS e das fontes (EOT, SVG, TTF, WOFF e WOFF2) do site, salvá-los localmente para provê-los juntamente com seu site.

<https://fontawesome.com/get-started>

Google Icons (fontes3.html)

Site: <https://material.io/icons/>

Exemplos: <http://google.github.io/material-design-icons/>

Passo 1: importe o CSS do Google Fonts

Passo 2: escolha seus ícones no site e use a sintaxe sugerida

The screenshot shows the Google Material Icons website interface. At the top, there is a grid of icons. The 'bug report' icon, which is a grey bug, is highlighted with a red square. Below the grid, there is a blue bar containing the selected icon, its name 'bug report', a size dropdown set to '24dp', and two radio buttons for color (one selected). To the right of the bar are download buttons for 'SVG' and 'PNGS', and a button labeled '< > ICON FONT' which is also highlighted with a red square. Below the blue bar, there are two sections: 'STEP 1: Setup Icon Font' and 'STEP 2: Use Icon in Your Site'. The 'STEP 2' section contains a code snippet for embedding the icon in HTML, which is highlighted with a red box.

bug report

bug report

build

cached

camera enhance

card giftcard

card membership

card travel

change history

check circle

chrome reader mo...

class

code

compare arrows

copyright

credit card

dashboard

date range

bug report

24dp

SVG

PNGS

< > ICON FONT

STEP 1: Setup Icon Font

Follow the [instructions](#) to embed the icon font in your site and learn how to style your icons using CSS.

STEP 2: Use Icon in Your Site

```
<!-- For modern browsers. -->
<i class="material-icons">bug_report</i>
<!-- For IE9 or below. -->
```

Download do Google Icons

Como pode ser visto na seção “Downloading everything” do endereço abaixo, também é possível baixar as fontes do google. Não vamos reproduzir este exemplo aqui porque o arquivo possui 60MB.

<http://google.github.io/material-design-icons/>

Downloading everything

Grab the [latest stable zip archive](#) (~60MB) of all icons or the [bleeding-edge version from master](#).

Desvantagens de trabalhar Icons Fonts

- Os browsers desenharam as fontes de forma ligeiramente diferente. Isso pode afetar o layout das páginas.
- Ao trabalhar com icons fonts, você automaticamente obriga o usuário a baixar o CSS dos estilos e os arquivos de fontes. Se o uso de ícones for excessivo na sua página, isso é vantajoso, caso contrário, o melhor seria baixar os ícones individualmente. Veja abaixo um exemplo.

Font Awesome Icons

<input type="checkbox"/> fa-solid-900.woff2	200	font	fontes1.html	43.4 KB
<input type="checkbox"/> fa-brands-400.woff2	200	font	fontes1.html	54.2 KB

Google Icons

<input type="checkbox"/> icon?family=Materi...	200	stylesheet	fontes2.html	668 B
<input type="checkbox"/> fLUhRq6tzZclQEJ-V...	200	font	Other	48.3 KB

Outros Icons Fonts

<https://github.com/WordPress/dashicons>

<http://www.linea.io/>

<http://themify.me/themify-icons>

<https://octicons.github.com/>

<http://konpa.github.io/devicon/>

<http://streamlineicons.com/free-icons.html>

<http://genericons.com/>

<http://themes-pixeden.com/font-demos/7-stroke/index.html>

<https://freebiesbug.com/psd-freebies/80-stroke-icons-psd-ai-webfont/>

<http://vorillaz.github.io/devicons/>

<http://www.alessioatzeni.com/metrize-icons/>

<https://icons8.com/line-awesome>

Sass e SCSS

Sass e SCSS (sass/index.html)

Sass é uma linguagem de folha de estilo que oferece uma sintaxe mais enxuta e otimizada para construção do estilo das páginas. Ela possui duas sintaxes:

Sass que organiza os código apenas por indentação (sem chaves e ponto-e-vírgula) e **SCSS**, mais recente, que é mais próxima do CSS mas também oferece melhorias.



CSS	Sass	SCSS
<pre>div.css-normal { margin: 20px; width: 300px; padding: 10px; background-color: indianred; }</pre> <pre>div.css-normal span { display: block; width: 48px; margin: 0 auto; background-color: rosybrown; border: 3px solid white; }</pre>	<pre>\$tamanho-div: 300px div.css-sass margin: 20px width: \$tamanho-div padding: 10px background-color: indianred span display: block width: 48px margin: 0 auto background-color: rosybrown border: 3px solid white</pre>	<pre>\$tamanho-div: 300px; div.css-scss { margin: 20px; width: \$tamanho-div; padding: 10px; background-color: indianred; span { display: block; width: 48px; margin: 0 auto; background-color: rosybrown; border: 3px solid white; } }</pre>

A sintaxe Sass e SCSS deve ser convertida para CSS

Para utilizar a notação do Sass, você precisará de um “compilador” (ou “transpilador”) capaz de transformar o código Sass/SCSS em CSS.

Opções:

1. Existem ferramentas de desenvolvimento que são capazes de fazer essa conversão para você automaticamente.
2. Outra opção é configurar uma infraestrutura no seu projeto para sempre converter o código automaticamente na hora de gerar os fontes de distribuição.

Neste exemplo vamos utilizar a opção 2 (com auxílio do NodeJS).

Passos (no Linux)

Windows ou Mac? → <http://sass-lang.com/install>

1. **Instale o NodeJS:** <https://nodejs.org/en/>
2. **Instale o SASS:** `sudo npm install -g node-sass`
3. **Compilar os scripts:**
 - a. **Sass:** `node-sass css/estilo.sass css/estilo-sass.css`
 - b. **SCSS:** `node-sass css/estilo.scss css/estilo-scss.css`

Esses passos vão transpilar seu arquivo Sass/SCSS para um CSS. No arquivo HTML você deve referenciar o código CSS convertido.

Configurar compilação automática (no Linux)

Vamos agora configurar um script de execução no *package.json* para automatizar o passo de compilação (supondo que você tenha o *http-server* instalado).

```
"scripts": {  
  "executar": "npm run compilar-sass && npm run compilar-scss &&  
http-server",  
  "compilar-sass": "node-sass css/estilo.sass css/estilo-sass.css",  
  "compilar-scss": "node-sass css/estilo.scss css/estilo-scss.css"  
}
```

Após essa configuração, execute:

```
npm run executar
```


Obrigado!