



**DIO**  
CEZAR  
**GO**

# Frameworks CSS



# AGENDA

---

- o que são?
- redefinição das folhas de estilo
- grids
- tipografia e ícones
- estilos pré-definidos
- componentes pré-definidos
- outros frameworks
- painéis administrativos
- bootstrap
- compatibilidade
- componentes do bootstrap
- materiais complementares

# O QUE SÃO?

- são bibliotecas pré-preparadas que se destinam a facilitar as estilizações de páginas web compatíveis com os padrões usando a linguagem CSS;
- os big players são → Bootstrap, Foundation e Materialize
- *frameworks* maiores usam intérpretes CSS como LESS ou SASS.

# O QUE SÃO?

- *frameworks* CSS podem oferecer diferentes módulos e ferramentas:
  - redefinição da folha de estilo
  - grid especialmente para o design responsivo
  - tipografia web, conjunto de ícones em sprites ou fontes de ícone
  - styling para tooltips, botões, elementos de formulários
  - partes de interfaces gráficas como o acordeão, guias, slideshow ou janelas modais

# REDEFINIÇÃO DE FOLHA DE ESTILO

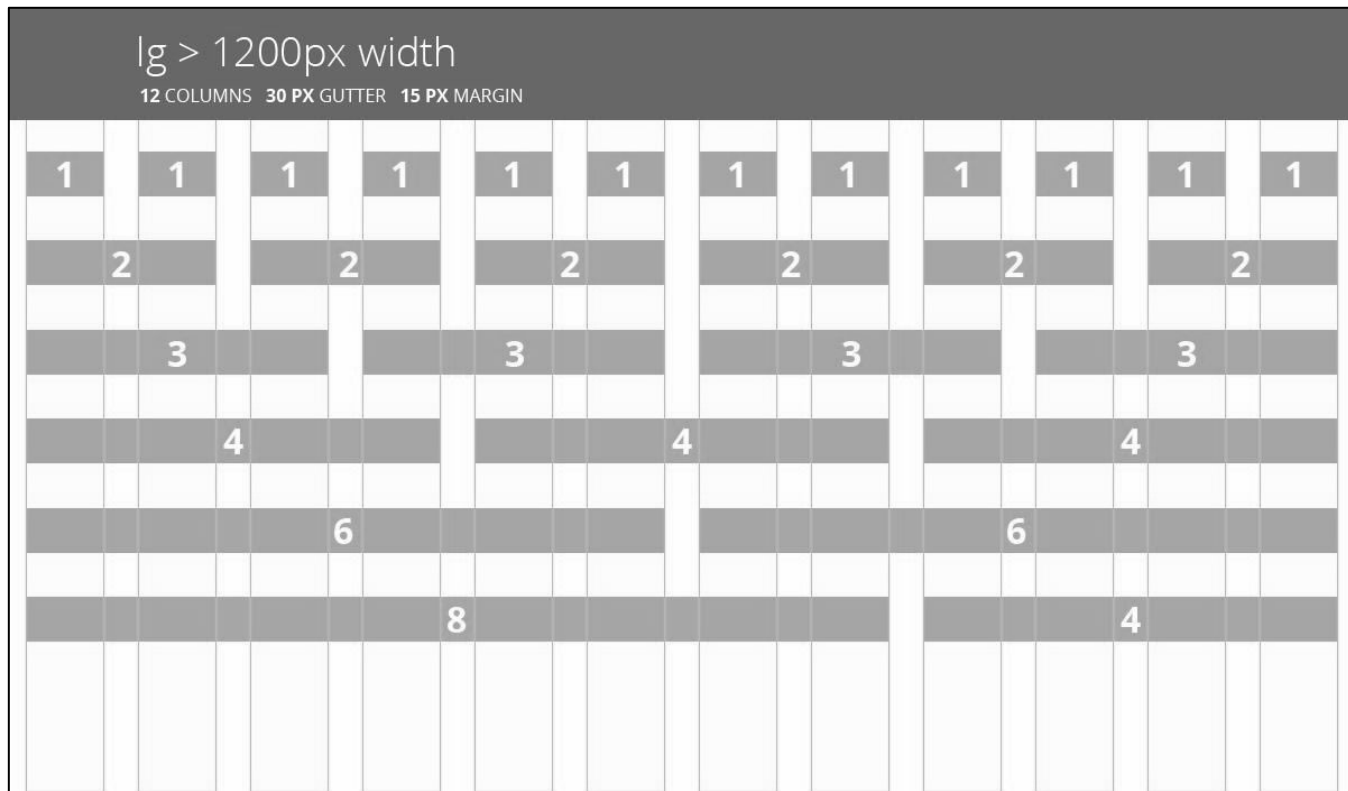
- por que isso é tão importante?
- como vimos, o css é renderizado em cascata;
- o navegador, por padrão, possui uma série de estilizações pré-definidas;
- essas estilizações podem ser incompatíveis com o layout a ser desenvolvido;
- veja mais em → <https://goo.gl/Dn6fsw>
- veja mais em → <https://goo.gl/Ph4hzN>
- um dos mais famosos é o Normalize.css
  - <https://goo.gl/FFgvvq>
- mas... nossos frameworks fazem isso por padrão;

# GRIDS

---

- as grids definem uma forma de diagramação do *layout*;
- apesar de muito material sobre a nova especificação css-grid, o conceito pode ser aplicado com outras técnicas no css;
- essencialmente uma grid é formada por um número máximo de colunas;
  - pode-se imaginar como as colunas de uma tabela;
- cada elemento poderá definir quantas colunas deverá ocupar nessa grid maior;

# GRIDS



# GRIDS

---

- cada coluna se comporta como um posicionamento relativo;
  - elas são alinhadas lado a lado enquanto “couverem” no *viewport* do navegador;
  - quando não couberem, serão colocadas umas abaixo das outras;
- cada conjunto de definição de grids deve estar agrupada por linha → *rows*;
- essencialmente, pode-se definir quantas colunas vão ser mostradas em cada um dos tamanhos pré-definidos;
- pode-se entender melhor o sistema de grids e gerar seus layout neste aplicativo web → <http://shoelace.io/>



# TIPOGRAFIA E ÍCONES

---

- a tipografia diz respeito às fontes utilizadas em seus projetos;
- já vimos como importar as fontes do Google Fonts;
- também aprendemos como importar uma fonte diretamente para nossas páginas;
- os *frameworks* apresentam quase sempre uma fonte própria que é utilizada para um conjunto de ícones utilizados no sistema;
- outros sistemas de fontes podem ser utilizados para seus ícones:
  - <http://fontawesome.io/>
  - <http://ionicons.com/>

# ESTILOS PRÉ-DEFINIDOS

---

- os *frameworks* também apresentam uma série de customizações pré-definidas;
- permite que você se preocupe com a estrutura da sua aplicação, deixando de lado estilizações complexas;
- oferecem recursos para alterações e personalizações destes elementos;
- botões, *tooltips*, formulários elementos de apresentação de conteúdos são normalmente pré-estilizados por estes *frameworks*;

# COMPONENTES PRÉ-DEFINIDOS

---

- além disso, são previstos a estilização de componentes comuns a vários websites, como por exemplo:
  - carrosséis;
  - tabs;
  - accordion;
  - janelas modais;
- cada *framework* trata seu set de componentes de uma forma diferente;
- um plugin *JavaScript* correspondente a cada *framework* deve ser incluído para o funcionamento destes componentes;

# OUTROS FRAMEWORKS

---

- existem outros frameworks:
  - <https://goo.gl/sBJFZJ>

# PAINÉIS ADMINISTRATIVOS

- mesmo com um *framework* CSS, criar um painel administrativo, pode ser uma tarefa que consumirá um bom tempo de seu desenvolvimento;
- por isso, se possível, utilize algo “pronto”
- AdminLTE → componente pronto para criação de uma painel administrativo com várias funcionalidades;
  - utiliza bootstrap;
  - utiliza sass;
  - pode ser achado em → <https://adminlte.io/>

# BOOTSTRAP

---

- instalação → `npm install bootstrap@4.0.0-beta`
  - ou baixe os arquivos no site oficial;
- as novidades em relação à versão anterior podem ser vistas em: <https://goo.gl/XW95x9>
- um *overview* dos *layouts* que podem ser construídos com bootstrap podem ser vistos em: <https://goo.gl/EiJGGy>
- o sistema de grids pode ser visto em: <https://goo.gl/m8AQmL>

# COMPATIBILIDADE

---

- uma lista de compatibilidades pode ser vista aqui → <https://goo.gl/oPAg5V>
- se algo não lhe agradar, pode-se utilizar o bootstrap na versão 3.0;
- mas... os mesmos conceitos se aplicam para as duas versões, com pequenas modificações de sintaxe e recursos que foram descontinuados ou renomeados;

# COMPONENTES DO BOOTSTRAP

---

- uma série de componentes são definidos para a utilização com o Bootstrap;
- alerts → mensagens de feedback para o usuário → <https://goo.gl/5qhTn1>
- badge → pequenas tags para colocar em seus elementos → <https://goo.gl/RcMLqS>
- breadcrumb → navegação do site, onde você está → <https://goo.gl/vwGL42>
- buttons → botões do seu site → <https://goo.gl/2rG45w>
- button group → botões agrupados → <https://goo.gl/2lBeYi>
- card → container de conteúdo com várias opções → <https://goo.gl/wiEqDr>
- carousel → slide de conteúdos → <https://goo.gl/RKU8Zr>



# COMPONENTES DO BOOTSTRAP

---

- collapse → mostra ou oculta conteúdos → <https://goo.gl/7p1ntA>
- dropdowns → mostra uma lista dropdown → <https://goo.gl/svMeu8>
- forms → toda estrutura de formulários → <https://goo.gl/ACvgTN>
- input-group → detalhes visuais para forms → <https://goo.gl/nqn6Hs>
- modal → janelas que se abrem sobre o seu conteúdo → <https://goo.gl/CynCHL>
- navs → criação de menus responsivos → <https://goo.gl/D27EfX>
- navbar → criação de menus responsivos para cabeçalhos → <https://goo.gl/D78mc2>
- e outros...

# ESTUDANDO BOOTSTRAP

---

- estudando um exemplo com a implementação simples de alguns componentes e conceitos do bootstrap;
  - <https://github.com/diogocezar/dctb-utfpr/tree/master/2017/programacao-web-2/ex08-bootstrap>
- estudando um template feito com bootstrap;
  - <https://github.com/diogocezar/dctb-utfpr/tree/master/2017/programacao-web-2/ex08-bootstrap/template-bootstrap-4>

# ATIVIDADE

---

- utilizando bootstrap, crie um *layout* com 3 seções:
  - menu → utilize o componente *nav* para a criação de um menu responsivo no header fixo, com os seguintes itens: home, produtos, contato;
  - produtos → crie uma grid com 4 colunas e 2 linhas; cada produtos deverá ter um título de uma descrição;
    - nos dispositivos abaixo de sm deve possuir 12 colunas;
  - contato → crie um form de contato em 2 colunas de tamanhos:
    - 8 e 4 para lg;
    - 7 e 5 para md;
    - 6 e 6 para sm;
    - 12 para xs;
    - a coluna primeira deverá conter: nome e email. A segunda deverá conter: assunto e mensagem;

# MATERIAIS COMPLEMENTARES

---

- <https://goo.gl/4siQ6X>
- <https://goo.gl/Ft4zgc>
- <https://goo.gl/KCWbV4>
- uma opinião contrária? → <https://goo.gl/FfzCGD>