

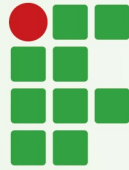
**INSTITUTO  
FEDERAL**

Santa Catarina

**Curso Superior de Tecnologia em  
Sistemas para Internet**

# **Programação para Internet I**





**INSTITUTO  
FEDERAL**  
Santa Catarina

Tópicos desta aula:

- **Introdução ao desenvolvimento de layouts para páginas WEB**
  - Normalize & Reset
  - Viewports e responsividade
  - Media queries e importação





**INSTITUTO  
FEDERAL**  
Santa Catarina

# Normalize & Reset



## – Normalize & Reset

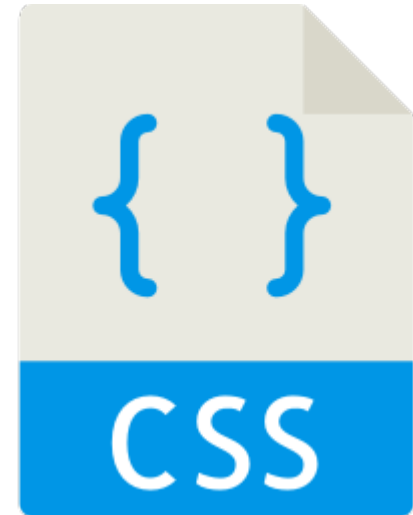
- Cada navegador possui uma forma de interpretar por padrão os elementos HTML não alterados por CSS;
- Eventualmente, uma margem de 1 pixel diferente em um determinado navegador por frustrar todo o projeto de interface;
- O objetivo do **reset** e **normalização** é fazer com que todos navegadores sejam padronizados, embora tenham diferença na abordagem



```
html, body, div, ul, ol, li, p, h1, h2, h3, h4 { style.css:1  
  margin: 0;  
  padding: 0;  
}  
  
h3 { user agent stylesheet  
  display: block;  
  font-size: 1.17em;  
  -webkit-margin-before: 1em;  
  -webkit-margin-after: 1em;  
  -webkit-margin-start: 0px;  
  -webkit-margin-end: 0px;  
  font-weight: bold;  
}
```

# Normalize

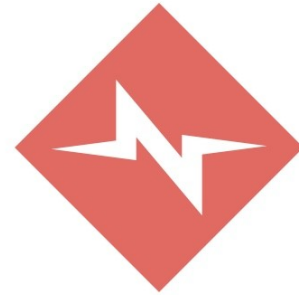
- Normalize é uma solução mais “suavizada” para este problema
- É um pequeno arquivo CSS que fornece consistência entre navegadores no estilo padrão dos elementos HTML.
- Isso significa que, se olharmos para os padrões W3C dos estilos aplicados pelos navegadores, e houver uma inconsistência em um dos navegadores, os estilos normalize.css corrigirão o estilo do navegador que tem a diferença.





# Normalize

- Então é uma boa prática incluir um normalize.css no seu projeto para evitar as divergências de renderização.
- Embora existam diversas opções de “normalize.css” disponíveis, a mais famosa é a de Nicolas Gallagher
- <https://necolas.github.io/normalize.css/>



**Normalize.css**

# Reset

- A abordagem do “reset” é mais agressiva.
- Basicamente ela diz: Não quero nenhuma formatação padrão dos navegadores.
- Você precisa (re)definir todos estilos de todos elementos.

## Título h1

### Título h2

#### Título h3

Um parágrafo qualquer

padrão

## Título h1

### Título h2

#### Título h3

Um parágrafo qualquer

Com reset

# Reset

- Se você gosta de ter controle absoluto sobre o estilo do seu site, utilize um reset.
- Também existam diversas opções de “reset.css” disponíveis, sendo uma das mais utilizadas:
- <https://meyerweb.com/eric/tools/css/reset/>

```
html, body, div, span, applet, object, iframe,  
h1, h2, h3, h4, h5, h6, p, blockquote, pre,  
a, abbr, acronym, address, big, cite, code,  
del, dfn, em, img, ins, kbd, q, s, samp,  
small, strike, strong, sub, sup, tt, var,  
b, u, i, center,  
dl, dt, dd, ol, ul, li,  
fieldset, form, label, legend,  
table, caption, tbody, tfoot, thead, tr, th, td,  
article, aside, canvas, details, embed,  
figure, figcaption, footer, header, hgroup,  
menu, nav, output, ruby, section, summary,  
time, mark, audio, video {  
margin: 0;  
padding: 0;  
border: 0;  
font-size: 100%;  
font: inherit;  
vertical-align: baseline;  
}
```

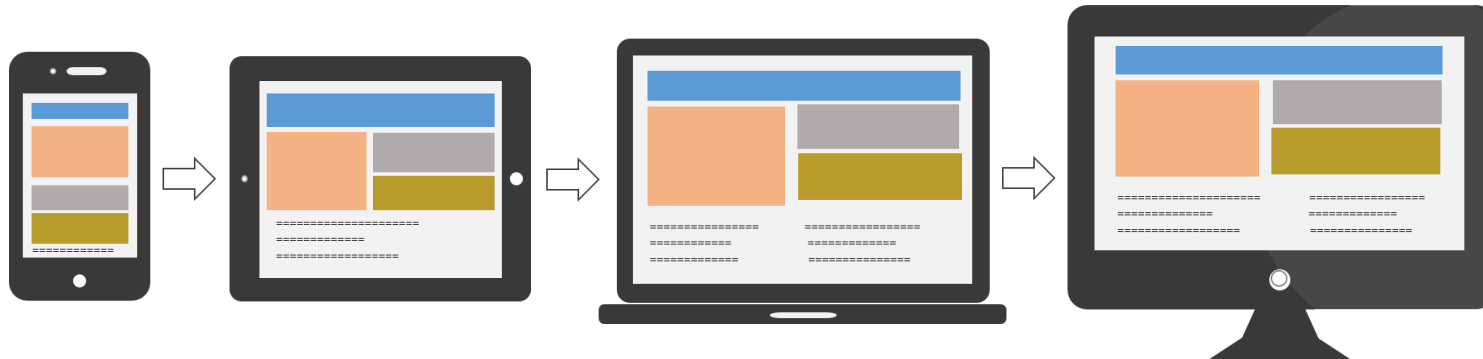


# Viewports e responsividade



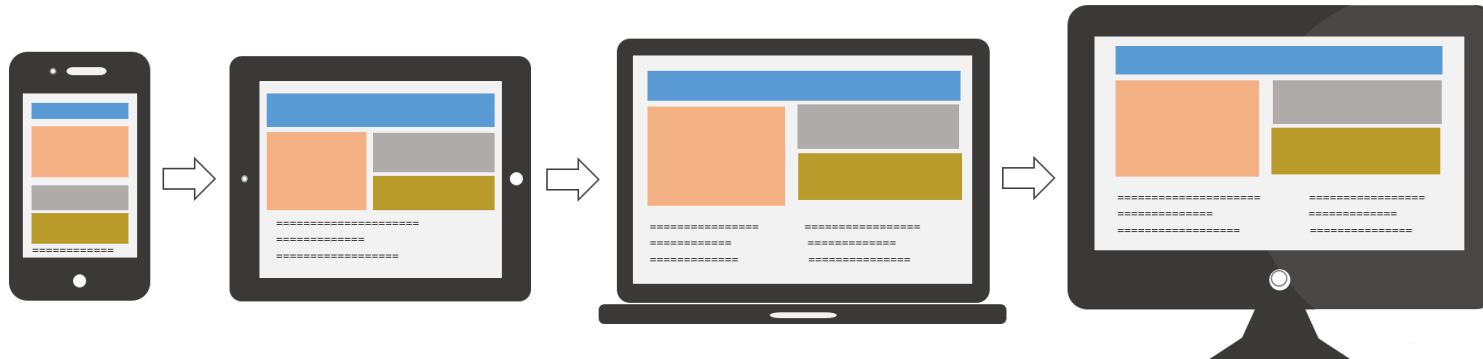
# Responsividade

- O que é um design responsivo?
- A resposta está relacionada a como a interface do site e suas informações estarão apresentadas em diferentes dispositivos
- O objetivo é evitar elementos de tamanhos desproporcionais, excesso de rolagens horizontais e verticais, textos difíceis de ler, etc.



# Responsividade

- Como fazer isto? Embora seja possível fazer uma interface (HTML&CSS) para cada dispositivo, seria algo muito complexo e a manutenção extremamente complicada;
- Então o objetivo passa a ter um projeto HTML e CSS que se consegue se adequar a tamanhos de telas diferenciados.
- Temos que considerar também que é mais fácil uma interface aumentar do que diminuir. Portanto, é comum utilizarmos a técnica *mobile first* e depois acrescentar complexidade.



# Viewport

- Viewport é a área visível de uma página para o usuário, de acordo com seu dispositivo.
- Antigamente, os sites eram feitos para monitores (desktop) apenas e possuíam medidas fixas (grandes) de largura principalmente.
- Com a chegada dos smartphones, os sites eram encolhidos para caber em pequenas telas.
- O HTML5 criou uma metatag *viewport* para termos o controle de como o navegador deve controlar a dimensão e escala da página no dispositivo

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<meta name="viewport" content="width=device-width, initial-scale=0.86, maximum-scale=5.0, minimum-scale=0.86">
```

# Breakpoints

- No desenvolvimento de layouts, precisamos escolhermos alguns breakpoints de acordo com o projeto e público alvo
- Breakpoint é o “ponto” no qual o conteúdo e o design de um site se adapta de uma determinada maneira para fornecer a melhor experiência possível ao usuário.
- Para tornar um site responsivo, os designers devem adicionar um breakpoint quando a interface parecer desalinhada ou “estranha” naquele viewport.

## Quais resoluções e dispositivos são relevantes?



Faça uma pesquisa e determine quantos e quais tamanhos de viewports (breakpoints) você utilizaria para um projeto web com público-alvo majoritariamente brasileiro.

- <https://gs.statcounter.com/screen-resolution-stats/all/brazil>
- <https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet/brazil>
- [https://www.w3schools.com/browsers/browsers\\_display.asp](https://www.w3schools.com/browsers/browsers_display.asp)

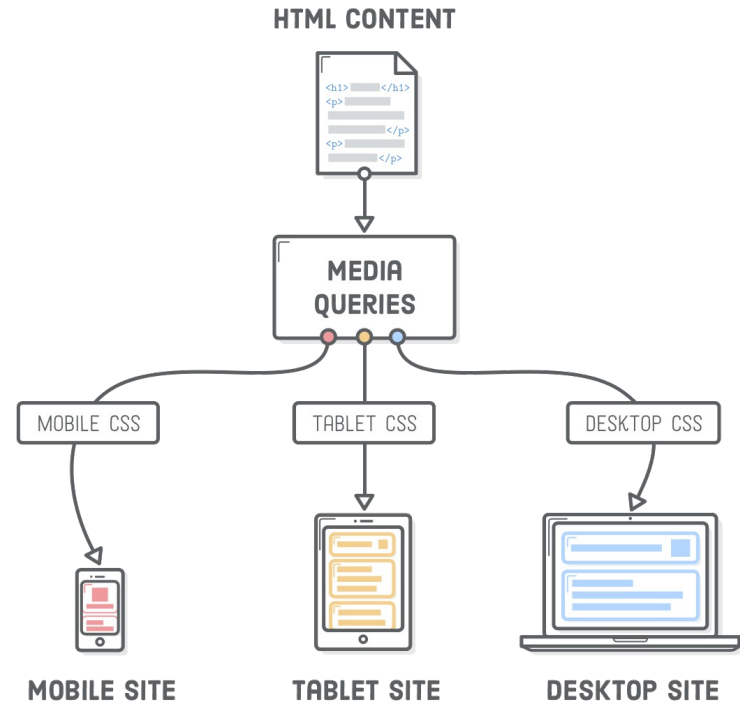


# Media queries e importação



# Layout responsivo

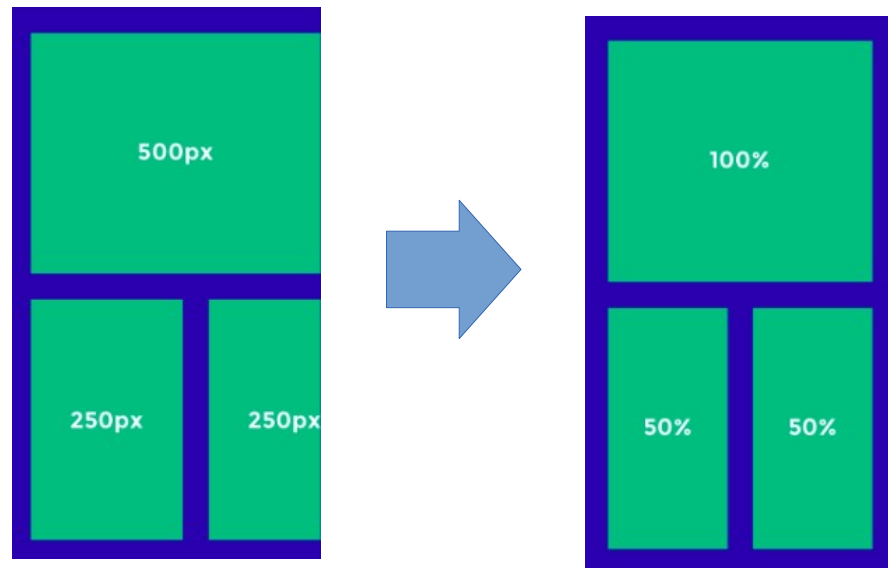
- Após a definição do projeto e dos breakpoints, o desenvolvedor consegue modificar dinamicamente a interface de acordo com os viewports utilizando media queries
- Você também pode fazer com que a interface e estilo se comportem de forma diferente em tipos de dispositivos diferentes (tela, TV, impressora, tty,...)



# Layout responsivo

- Boa parte dos problemas dos layouts responsivos podem ser contornadas através de técnicas simples de CSS baseadas em fluidez e medidas proporcionais.
- No entanto, 2 ou 3 colunas, mesmo que fluídas, podem ficar estranhas em um viewport pequeno como o de um celular.

```
/* exemplos: */  
width:90%  
max-width:1200px;  
min-width: 200px;
```



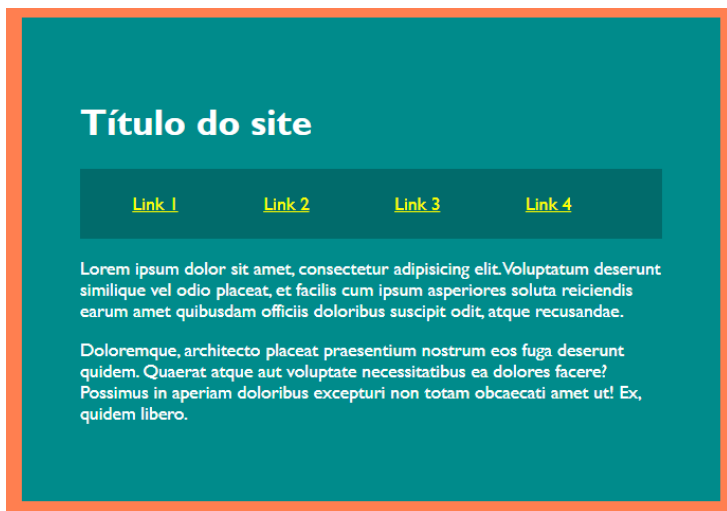
# Media queries

- No CSS, podemos fazer declarações específicas para viewports específicos utilizando media queries.
- Exemplo: Esconder um banner de propaganda em um *viewport* menor que 800px

```
@media (max-width: 600px)
{
  .banner
  {
    display: none;
  }
}
```

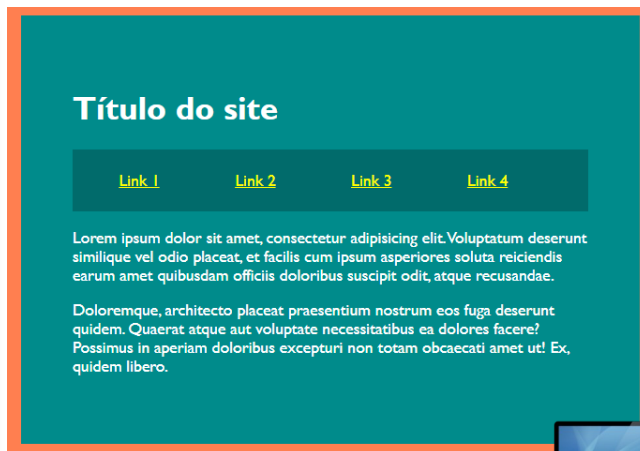
# Media queries

- Podemos também escolher o tipo de dispositivo/tela.
- Este exemplo modifica o site para impressão.



```
@media print{  
  *{  
    background-color: white;  
    color: black  
  }  
  nav{  
    display: none;  
  }  
  p{  
    text-align: justify;  
  }  
}
```

# Media queries



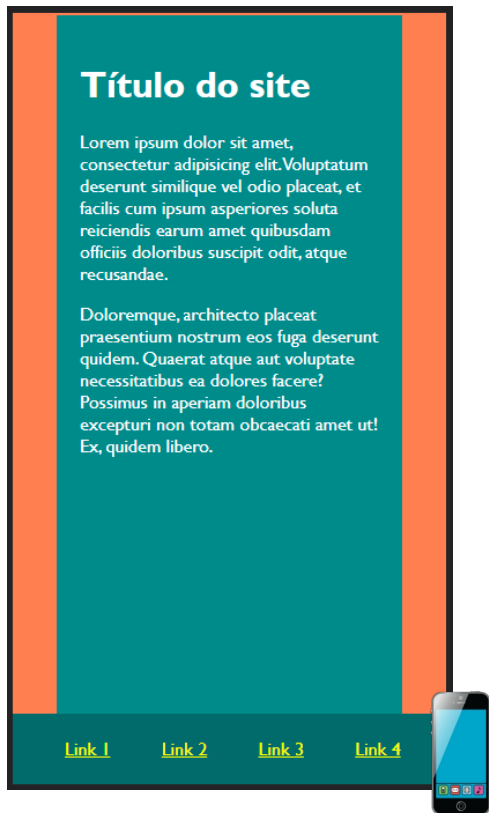
```
@media print{  
  
  *{  
  
    background-color: white;  
    color:black  
  
  }  
  
  nav{  
  
    display: none;  
  
  }  
  
  p{  
  
    text-align: justify;  
  
  }  
  
}
```



# Media queries

- Outra forma de utilizar media queries é combinando condições.
- Lembre-se que as regras fora das queries continuam valendo se não forem sobrepostas.
- Consulte as media features para compor as regras:

[https://www.w3schools.com/cssref/css3\\_pr\\_mediaquery.asp](https://www.w3schools.com/cssref/css3_pr_mediaquery.asp)



```
nav {  
  background-color: rgb(1, 107, 107);  
  padding: 5px;  
}  
@media screen and (max-width:480px) {  
  nav {  
    width: 100%;  
    position: fixed;  
    bottom: 0;  
    left: 0;  
  }  
  .container{  
    min-height: 100vh;  
    padding: 20px;  
  }  
}
```

# Importação

- Caso seu projeto seja complexo, você pode criar arquivos CSS específicos para media queries, deixando seu código mais organizado
- O importante é você se organizar na questão das sobreposições de regras, para evitar códigos repetidos desnecessários dificultando a manutenção do projeto

```
<link rel="stylesheet" href="style.css">  
<link rel="stylesheet" media="screen and (max-width: 600px)" href="smallscreen.css">  
<link rel="stylesheet" media="screen and (min-width: 900px)" href="widescreen.css">
```

# Media queries

```
<link rel="stylesheet" media="(prefers-color-scheme:dark)" href="dark.css">
```

## Título do site

[Link 1](#) [Link 2](#) [Link 3](#) [Link 4](#)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Voluptatum deserunt similique vel odio placeat, et facilis cum ipsum asperiores soluta reiciendis earum amet quibusdam officiis doloribus suscipit odit, atque recusandae.

Doloremque, architecto placeat praesentium nostrum eos fuga deserunt quidem. Queraat atque aut voluptate necessitatibus ea dolores facere? Possimus in aperiam doloribus excepturi non totam obcaecati amet ut! Ex, quidem libero.

Tema escuro

## Título do site

[Link 1](#) [Link 2](#) [Link 3](#) [Link 4](#)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Voluptatum deserunt similique vel odio placeat, et facilis cum ipsum asperiores soluta reiciendis earum amet quibusdam officiis doloribus suscipit odit, atque recusandae.

Doloremque, architecto placeat praesentium nostrum eos fuga deserunt quidem. Queraat atque aut voluptate necessitatibus ea dolores facere? Possimus in aperiam doloribus excepturi non totam obcaecati amet ut! Ex, quidem libero.

Tema claro

```
body {  
  background-color: black;  
}  
  
.container {  
  background-color: rgb(36, 61, 61);  
  color: white;  
}  
  
nav {  
  background-color: rgb(14, 22, 22);  
}
```

Arquivo ./dark.css

# Desenvolvendo...

