Mobile First

Front End Web

Msc. Lucas G. F. Alves

e-mail: lucas.g.f.alves@gmail.com





Planejamento de Aula

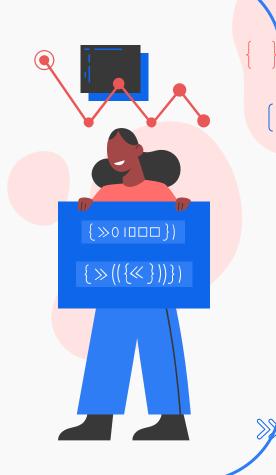
Revisão Interatividade

Git Pages e Qr Code Generator

Box-model e Box-sizing

Responsive Web Design

Design Mobile-First





Revisão





Progressive Enhancement



Muitas pessoas não gostam das versões atuais de navegadores e utilizam navegadores antigos como IE8 ou mais antigos, onde algumas propriedades novas não são aceitas.

Mas temos que implementar sempre o mais inovador ao usuário, como resolver?

Fazer funcionar em qualquer navegador sem prejudicar os antigos.

Graceful degradation foi o nome da primeira técnica a pregar isso.

Este removia funcionalidades não suportadas e degradava "graciosamente".

Hoje se utiliza Progressive enhancement com uma ideia parecida mas ao contrário, se desenvolve funcionalidades normalmente e acrescentando pequenas melhorias mesmo que só funcionem nos navegadores modernos.







Exercícios



1) Colocar alguns efeitos nos painéis com border-radius, box-shadow e text-shadow.

```
.painel { border-radius: 4px; box-shadow: 1px 1px 4px #999; } .painel h2 { text-shadow: 3px 3px 2px #FFF; }
```

2) Utilizar bordas internas aos elementos. Aplicar aos painéis de aula. Basta usar a opção inset: box-shadow: inset 1px 1px 4px #999;

3) Utilizar border-radius em algumas bordas específicas.

```
.busca { border-top-left-radius: 4px; border-top-right-radius: 4px; }
```

4) Utilizar cores com canal Alpha para translucência de 80%.

```
.painel h2 { text-shadow: 1px 1px 2px rgba(255,255,255,0.8); }
```



5) Aplicar gradientes aos paineis de aulas.

```
.aula1{background: linear-gradient(#f5dcdc, #bebef4);}
.aula2{background: linear-gradient(#dcdcf5, #f4bebe);}
```





CSS3 Transitions

Com transitions, é possível animar o processo de mudança de algum valor do CSS.

Por exemplo: temos um elemento na posição top:10px e, quando passarmos o mouse em cima (hover), queremos que o elemento mude para top:30px.

O CSS básico muda de uma vez sem transição:

```
#teste { position: relative; top: 0; }
#teste:hover { top: 30px; }
```

Com transição animando durante 2 segundos:

```
#teste:hover {transition: top 2s; }
```







CSS3 Transitions



Por padrão a transição é linear, mas tem outros tipos de transições, como:

- linear velocidade constante na animação;
- ease redução gradual na velocidade da animação;
- ease-in aumento gradual na velocidade da animação;
- ease-in-out aumento gradual, depois redução gradual na velocidade da animação;
- cubic-bezier(x1,y1,x2,y2) curva de velocidade para animação customizada

```
(avançado);
#teste:hover { transition: top 2s ease; }
#teste { position: relative; top: 0; color: white; } #teste:hover { top: 30px; color: red;
transition: top 2s, color 1s ease; } /* incluindo cores*/
#teste:hover { transition: all 2s ease; } /*o efeito para todas as propriedades*/
```







CSS3 Transforms



Nova especificação para alterar propriedades visuais dos elementos, como ângulo, mostrá-lo em uma escala maior ou menor ou alterar a posição do elemento sem sofrer interferência de sua estrutura.

Translate

```
.header { /* Move o elemento no eixo horizontal */ transform: translateX(50px); }  \# main \{ /* Move o elemento no eixo vertical */ transform: translateY(-20px); \}  footer { /* Move o elemento nos dois eixos (X, Y) */ transform: translate(40px, -20px); }
```







CSS3 Transforms



Rotate

#menu-conteudo { transform: rotate(-10deg); }

Scale

#aula1 li { /* Alterar a escala total do elemento */ transform: scale(1.2); }

#aula2 li { /* Alterar a escala vertical e horizontal do elemento */ transform: scale(1, 0.6); }







CSS3 Transforms



Skew

```
footer { /* Distorcer o elemento no eixo horizontal */ transform: skewX(10deg); } #social { /* Distorcer o elemento no eixo vertical */ transform: skewY(10deg); }
```

Mais de um transform junto:

```
html { transform: rotate(-30deg) scale(0.4); }
```







Exercícios



1) Quando passar o mouse em algum exercício dos painéis de destaques, mostrar uma sombra por trás. Utilize também transição para que essa sombra apareça suavemente:

```
.painel li:hover { box-shadow: 0 0 5px #333; transition: box-shadow 0.7s; }
```

2) Colocar agora um fundo branco no elemento. Anime esse fundo também, fazendo um efeito tipo fade. Indicar que todas as propriedades devem ser animadas.

```
.painel li:hover { background-color: rgba(255,255,255,0.8); box-shadow: 0 0 5px #333; transition: 0.7s; }
```

3) Quando passar o mouse em cima do item do painel, aumentar o elemento em 20%.

```
.painel li:hover { transform: scale(1.2); }
```

4) Alterar o comando anterior para fazer rotacionar suavemente em 5 graus.



.painel li:hover { transform: scale(1.2) rotate(-5deg); }





CSS media types



São suportes de regras de layout diferentes, que podem ser declarados ao se invocar um arquivo CSS:

```
<link rel="stylesheet" href="site.css" media="screen" />
<link rel="stylesheet" href="print.css" media="print" />
<link rel="stylesheet" href="handheld.css" media="handheld" />
@media screen { body { background: blue; color: white; } }
@media print { body { background: white; color: black; } }
```

O media type screen determina a visualização normal, no Desktop. O media type print com regras de impressão. O media type handheld, é voltado para dispositivos móveis.

Só que o handheld foi desenvolvido para celulares antigos, e os novos smartphone, tablets, televisões, já visualizam a página inteira com todos os detalhes.







CSS3 media queries



É uma nova forma de adaptar o CSS.

```
k rel="stylesheet" href="base.css" media="screen">
k rel="stylesheet" href="mobile.css" media="(max-width: 480px)">
```

Outra forma de declarar os media types é separar as regras dentro do mesmo arquivo CSS:

```
@media screen { body { font-size: 16px; } }
@media (max-width: 480px) { body { font-size: 12px; } }
```







Viewport

>>>

Por padrão, iPhones, Androids e afins costumam considerar o tamanho da tela visível, chamada de viewport como grande o suficiente para comportar os Sites Desktop normais.

A Apple criou então uma solução que depois foi copiada pelos outros smartphones, que é configurar o valor que julgamos mais adequado para o viewport:

<meta name="viewport" content="width=320">

Ou utilizar um valor device-width definido pelo fabricante.

<meta name="viewport" content="width=device-width">







Exercícios



1) Adaptar nossa home page (index.html) para mobile. Escrever nosso CSS de adaptação em um novo arquivo, chamado mobile.css. Crie esse arquivo e o importe no head do index.html.

```
<meta name="viewport" content="width=device-width">
link rel="stylesheet" href="css/mobile.css" media="(max-width: 320px)">
```

2) A página hoje tem o tamanho fixo em 940px e é centralizada (com o uso do seletor .container). No mobile.css.

```
.container { width: 96%; }
```

3) Ajustar os elementos do topo da página.

```
header h1 { text-align: center; }
header h1 img { max-width: 50%; }
.carrinho { display: none; }
.menu-conteudo { position: static; text-align: center; }
(({ >>})) « }
```





Exercícios



4) Aumentar o espaço entre os link.

```
.menu-opcoes ul li { display: inline-block; margin: 5px; }
```

5) Ajustar a seção de busca, o menu da esquerda e a imagem de destaque

```
.busca, .menu-conteudo, .destaque img { margin-right: 0; width: 100%; }
```

6) Ajustar os painéis de destaques de aulas.

```
.painel { width: auto; }
.painel li { width: 30%; }
.painel img { width: 100%; }
```

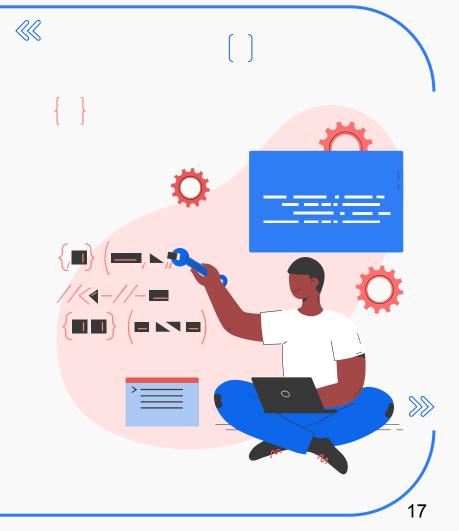
7) Ajustar a media query para aplicar o CSS de adaptação a qualquer tamanho de tela menor.

```
k rel="stylesheet" href="css/mobile.css" media="(max-width: 939px)">
```





Git Pages





Git Pages



É uma ferramenta do Github para hospedar uma página.

Com o repositório e os arquivos de sua página no repositório, importante o arquivo **index.html** deve estar na mesma hierarquia das pastas dos arquivos css, js, img. Esta hierárquia deve estar fora de pastas no github.

No Github -> Clicar em Settings -> Clicar em Pages .

Definir Source: Deploy from a branch.

Definir Branch: Selecionar a branch main.

Atualizar a página e deve aparecer.

Your site is alive at: ../nickaluno.github.io/nome_projeto









Qr Code Generator

É uma ferramenta para criação de Qr codes.

Ferramenta free e com customizações de seus Qr codes.

Criar uma conta no Qr Code Generator.

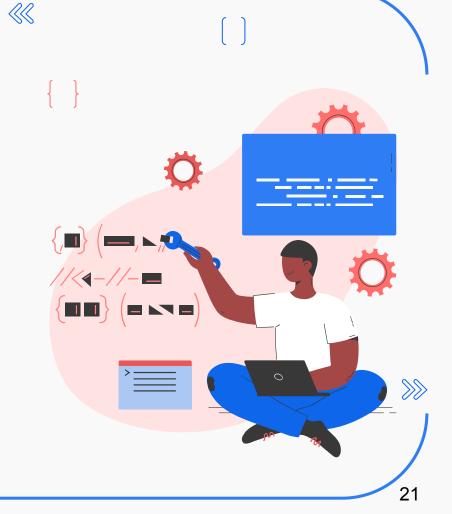
Deve ter o link do seu site hospedado.

E seguir o passo-a-passo do site para configurar o Qr code.





Box model e box-sizing



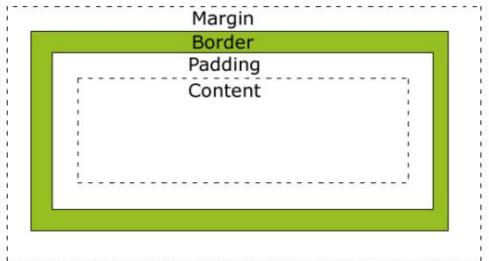


Box model e box-sizing



Box model e box-sizing

O Box model afeta outras elementos da nossa página e é constituído de quatro áreas retangulares: conteúdo (content), espaçamento (padding), bordas (borders) e margens (margin).









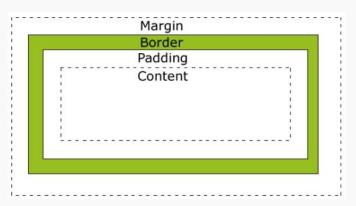
Box model e box-sizing



Box-sizing

O box-model teve muita incompatibilidade com alguns navegadores devido a quantidade de propriedades e regras para definí-lo.

Mas como a ideia era boa acabou se transformando em Box-sizing no CSS3, permitindo trocar a box-model.



Box-sizing: content-box indica que o tamanho dele é definido pelo seu conteúdo apenas. Mas podemos trocar por box-sizing: border-box que indica que o tamanho agora levará em conta até a borda





Responsive Web Design





Responsive Web Design



A página principal desenvolvida, com o mesmo HTML e pequenos ajustes de CSS, suporta diversas resoluções diferentes, desde a pequena de um celular até um Desktop.

Essa prática é o que o mercado chama de Web Design Responsivo. O termo surgiu num famoso artigo de Ethan Marcotte e diz exatamente o que acabamos de praticar. São 3 os elementos de um design responsivo:

- layout fluído usando medidas flexíveis, como porcentagens; trocar px -> %
- media queries para ajustes de design;
- uso de imagens flexíveis.

A ideia do responsivo é que a página se adapte a diferentes condições, em especial a diferentes resoluções.





Design Mobile-First





Design Mobile-First



Foi criado a Página principal do site (home page) do projeto e otimizamos nosso site para dispositivos móveis.

Aplicamos estilização e codificamos originalmente pensando nos browsers do desktop.

Posteriormente, aplicamos os conceitos de media queries e viewport para ajustar o projeto para telas menores.

É comum desenvolver para desktop primeiro, que são a maioria dos usuários, e depois ajustar o design para mobile. Mas isso não é o melhor, nem o mais fácil.

Mobile-First é fazer o design mobile primeiro e depois, ajustar para o desktop. O resultado final deve ser um site que funciona tanto no desktop quanto no mobile, como antes, só mudamos a ordem do fluxo de desenvolvimento.







Design Mobile-First



Evoluindo o design para desktop

Imagine que, em telas maiores que 600px, queremos flutuar uma imagem a esquerda:

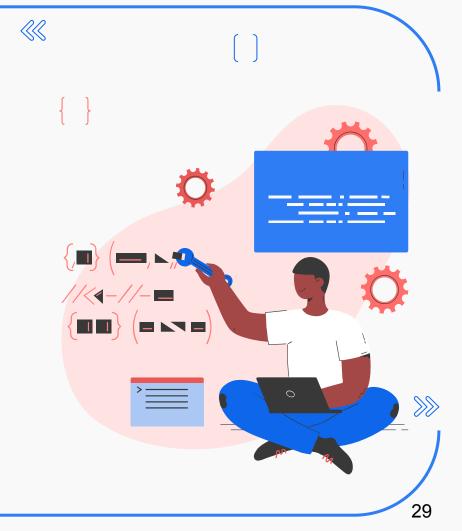
```
@media (min-width: 600px) {
    img {
        float: left;
    }
}
```

Desenvolver mobile-first, é utilizar media queries do tipo min-width para implementar as mudanças para o tablet/desktop.





Exercício





Exercícios

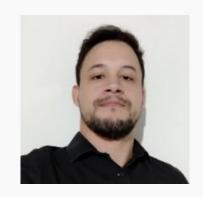


- 1) Criar um novo projeto com index.hmtl e index.css dentro da pasta aula 9.
- 2) Criar a estrutura padrão no index.html.
- 3) Definir no CSS a alteração da cor de fundo de body de acordo com o tamanho de tela. body{ background-color:#66ffcc;}
 @media screen and (min-width:480px) and (max-width:768px)
 {body{background-color:#00FF33;}}
- 4) Atribuir outras cores para tamanhos diferentes em min-width: 480px, 1024px, 1200px e max-width: 768px, 1035px, 1300px, 1360px.





Professor



Lucas G. F. Alves



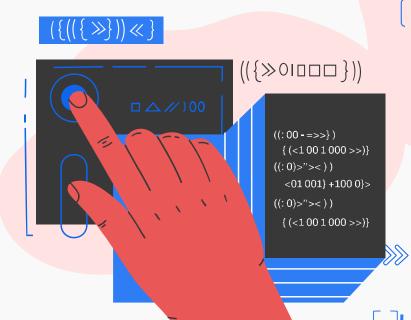






Obrigado!

E-mail:lucas.g.f.alves@gmail.com



>>>>



