Mobile First

Front End Web

Msc. Lucas G. F. Alves

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





Planejamento de Aula

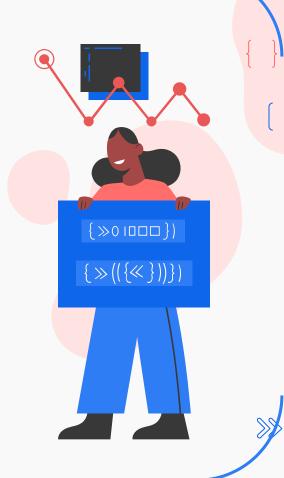
Revisão Mobile First

Responsividade

Estrutura Flexível

Novos componentes do HTML5

Exercícios





Revisão





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.





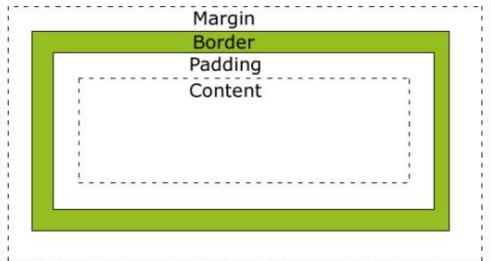


Box model e box-sizing



Box model e box-sizing

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



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



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.







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.











Conceitos:

Fluído

Seu foco é na adaptação da estrutura.

Em outras palavras, uso de porcentagem nas medidas.

Adaptativo

Seu foco é específico na experiência do usuário.

Exemplo: Desenvolver dois sites dedicados, sendo um para cada aparelho

Responsivo

É a combinação entre sites fluído + adaptativo.

Onde, existe somente uma única estrutura HTML (com navegação) seja onde ele estiver sendo visualizado (telas grandes ou pequenas).









Exemplo de site responsivo:

























Media Queries:

```
<link rel="stylesheet" href="small.css" />
```

k rel="stylesheet" href="medium.css" media="only screen and (min-width:640px) and
(max-width:1024px)" />

k rel="stylesheet" href="large.css" media="only screen and (min-width:1024px)" />

Este código define os Breakpoints usados no site.

- MediaQueries checam qual breakpoint executar.
- Os CSS são carregados de acordo com o breakpoint
- É só CSS. Não tem nada de JavaScript aqui.

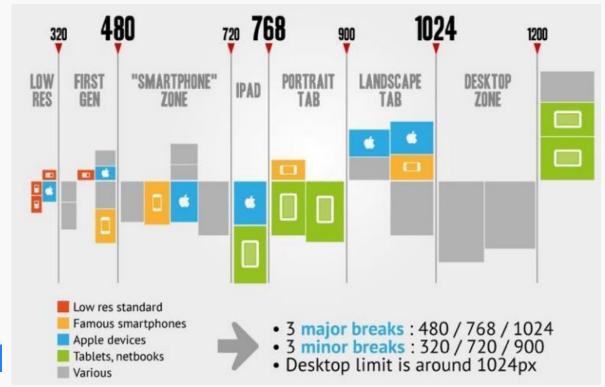






>>

Media Queries:









Carregamento condicional

Site responsivo != (desktop + mobile)

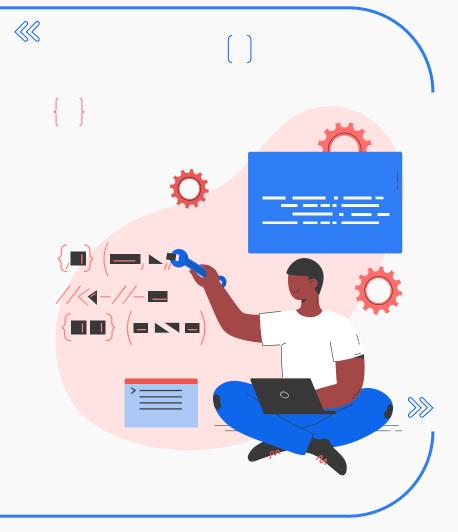
Evita carregamento extra dos recursos (imagens, CSS, JS)







Estrutura Flexível



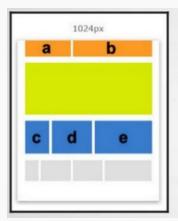


Estrutura Flexível



Porcentagem

Fórmula mágica: elemento alvo / contexto = dimensão



exemplos:

a: 340px 340/1024= 33,203125%

b: 684px 684/1024= 66,796875%

c: 200px 200/1024= 19,53125%

d: 400px 400/1024= 39,0625%

e: 424px 424/1024= 41,40625%









Estrutura Flexível



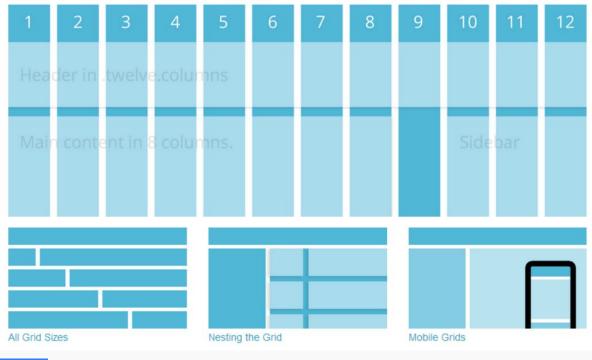






Imagem e Video Responsivo





Imagem e Vídeo Responsivo



Imagens Flexíveis

Redimensionamento proporcional (automático)

Sugerido somente dentro dos limites do mesmo breakpoint img {max-width:100%; height:auto;}











Imagem e Vídeo Responsivo



Vídeos Flexíveis

Utilizando a tag vídeo dentro de uma Figure podemos alterar seu tamanho.





Exercícios





Exercícios



- 1) Nos arquivos CSS das páginas, modificar as imagens que estão definidas em pixels para porcentagem.
- 2) Definir as imagens das páginas dentro das tags figures.
- 3) Colocar um vídeo responsivo na página de contato.
- 4) Nas páginas Blog e Contato, definir os media queries no CSS aplicando ajustes para tablet e desktop.





HTML5 Input Range





HTML5 Input Range



Input Range

O <input type="range"> é um componente novo do HTML5 com bom suporte já nos navegadores que representa um slider numérico.

Ele recebe atributos min e max com o intervalo numérico possível. Opcionalmente, há o atributo step que indica de quanto em quanto o número deve pular.

<input type="range" min="36" max="46" value="42" step="2" name="tamanho">

Ele já vêm com estilo padrão do navegador em questão o que é bem interessante. A interface padrão é familiar para o usuário.













O uso de tabelas era muito comum há alguns anos para a definição de áreas. Seu uso para essa finalidade acabou se tornando prejudicial pela complexidade da marcação, o que dificulta bastante a manutenção das páginas.

Além disso havia uma implicação direta na definição de relevância do conteúdo das tabelas para os indexadores de conteúdo por mecanismos de busca. Ainda assim, hoje, quando queremos exibir uma série de dados tabulares, é indicado o uso da tag de tabela .

```
table>

Título da primeira coluna
Título da segunda coluna

Linha 1, coluna 1.
```







>>

Na primeira linha da tabela, as células são indicadas com a tag , que serve para diferenciar seu conteúdo das células de dados.

Existem diversas maneiras de se alterar uma estrutura de uma tabela, como por exemplo: Indicar que uma célula ou ocupa mais de uma linha de altura por meio do atributo rowspan, ou então que ela ocupa mais de uma coluna de largura com o uso do atributo colspan.

Pode-se adicionar um título à tabela com a tag <aption>.

Ainda existem as tags <thead>, <tfoot> e , que servem para agrupar linhas da tabela. Dentro do grupo <thead> devemos ter apenas linhas contendo a tag como célula.

Outra tag de agrupamento e que permite que sejam definidas as colunas, é a tag <colgroup>. Dentro dessa tag definida uma tag <col> para cada coluna, e dessa maneira se adiciona alguns atributos que $(\{((\{ \gg \})) \ll \})$ influenciarão todas as células daquela coluna.





```
Exemplo:
              <caption>Produtos.</caption>
                   <colgroup>
                        <col width="80%">
                        <col width="20%">
                   </colgroup>
                   <thead>
                        Quantidade de Produtos
                             Preço
                        </thead>
                   Produto1
                        <tfoot>
                        Valor total: $45.00
                        </tfoot>
```







Email e URL

Os inputs do tipo email e url permitem que os dispositivos móveis, principalmente, exibam um teclado adaptado para facilitar esse tipo de entrada. Por exemplo, o iPhone exibe um teclado com o caractere @ e com as opções de domínio .com.

```
<input type="email" name="email">
```

<input type="url" name="endereco">











Number

O input do tipo number, além de exibir um teclado numérico em dispositivos móveis, nos navegadores modernos exibe um controle que permite incrementar ou decrementar o valor do campo clicando em uma seta para cima ou para baixo.

Além dessa diferença visual, é possível determinar valores mínimos, máximos e se há uma escala de valores válidos. No exemplo, o elemento deve aceitar números múltiplos de 5 com o limite do valor "100"

<input type="number" max="100" step="5">









Date, month, week, time, datetime e datetime-local

<input type="date" name="validade">

Os controles de "date picker" são feitos para coletar uma informação de data ou hora. São várias as possibilidades de formato de data ou hora necessárias. No navegador Opera, quando utilizado esse tipo de controle, o usuário pode selecionar uma data a partir de um calendário. É possível especificar datas mínima e máxima.

Em geral, os navegadores devem oferecer alguma funcionalidade de escolha de datas para o

usuário, como o Chrome:









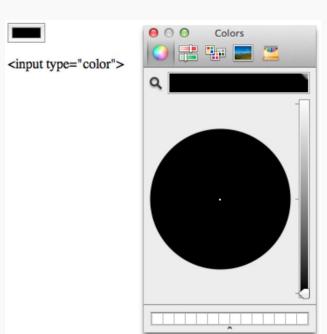




Color

<input type="color" name="cor_olhos">

O elemento input do tipo color permite que seja exibido um "color picker" para o preenchimento do seu valor. O Chrome no Mac, por exemplo, exibe o color picker padrão do sistema:











Search

<input type="search" results="10">

O input do tipo search exibe um campo específico para busca. O atributo "results" determina quantas últimas buscas serão armazenadas e lembradas, além de exibir uma lupa dentro do campo (Safari e Chrome).

Tel

<input type="tel" name="telefone">

O input do tipo tel foi especificado para coletar um número de telefone.

Em dispositivos com teclados virtuais como smartphones e tablets, é comum o teclado ser adaptado para exibir apenas opções relevantes à entrada de números telefônicos.

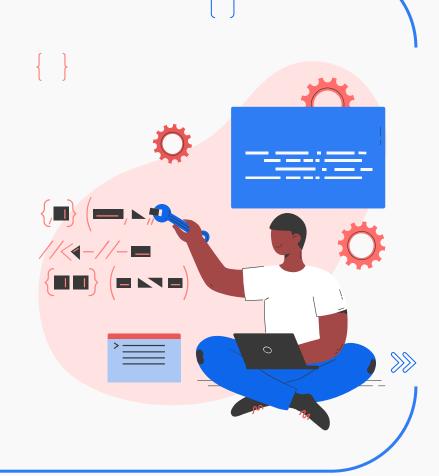




X

+ * #

Novos atributos para formulário





Novos atributos para formulário



No HTML5 estão definidos atributos para os elementos de formulario, visando implementar algumas necessidades comuns que antes não eram possíveis de serem atendidas puramente com a marcação do formulário.

autofocus

Sua presença indica que aquele campo deve iniciar com foco quando a página for carregada.

O usuário já pode começar a digitar algo sem nenhum clique.

<input name="nome" autofocus>

placeholder

<input type="text" name="nome" placeholder="Insira seu nome">

O atributo placeholder exibe o texto contido em seu valor dentro do elemento do formulário caso o seu valor seja vazio.







Novos atributos para formulário



autocomplete, list e datalist

É possível implementar uma funcionalidade de sugestão de valores com mais facilidade.

```
<input type="text" list="cidades" autocomplete="on">
    <datalist id="cidades">
        <option value="Brasília">
        <option value="Rio de Janeiro">
        <option value="São Paulo">
        </datalist>
```





Exercícios





Exercícios

1) Implemente a funcionalidade de escolher o volume de audio usando um input range do HTML5, colocando o código a seguir:

```
HTML -> <audio id="audio" src="arquivo.mp3"></audio> <input type="range" min="0" max="1" step="0.01" id="input-range" />
```

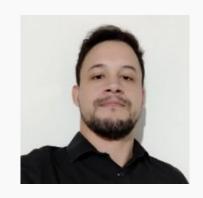
```
JS-> const Audio= document.getElementById("audio");
    const InputRange= document.getElementById("input-range");
    volumeSlider.addEventListener("input", () => { Audio.volume = InputRange.value;});
```

2) Dentro da Página Conteúdo crie uma tabela para relacionar as aulas e exercícios feitos. Criem as tags semanticas header, body e footer e atribuam uma classe container a elas definindo um tamanho de largura. Escrevam sobre as aulas e os exercícios com detalhes. Apliquem estilização na página conteúdo.





Professor



Lucas G. F. Alves





Obrigado!

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



>>>>



