

4.3 Media Queries e Type

00:15:14 - 00:42:28

Continuando nosso assunto sobre a responsividade, vocês lembram do projeto do site da Ada que vocês realizaram lá no nosso curso do FrontEnd #1? Lembram também dos principais desafios que vocês encontravam durante a construção, quando vocês tentavam diminuir a janela do navegador para tentar ver como ficava o site em telas menores e esses cenários aconteciam? Então, por exemplo, o menu não ficava da forma como deveria ser.

00:42:29 - 01:04:27

O texto ficava por cima, o formulário ficava um do lado do outro, mas ficava pequenininho demais, a tela não fluía e tal. Como a gente falou lá na nossa aula anterior, um dos motivos era porque não utilizávamos a tag “viewport”, que é aquela janela de exibição que permite com que a gente consiga transformar nosso site em responsivo.

01:05:12 - 01:36:23

Porém, no CCS existem regras nas quais a gente pode utilizar dentro desse processo de construção de sites responsivos, que permitem que a gente adicione mais mudanças de comportamentos quando a tela vai se diminuindo. Assim, evita de menu quebrar, do formulário ficar pequeno demais, o texto ficar desproporcional ao tamanho da tela. A gente tem propriedades no CCS que permitem que a gente consiga manipular e trabalhar isso da melhor forma.

01:37:02 - 02:18:18

Só que antes de a gente falar mais a fundo sobre essas regras, tem um assunto que é muito importante também da gente entender que é sobre pontos de quebra. Voltando ao site de novo da Ada, por que é que ele quebrou? Porque não havia pontos de quebra, que a gente chama de “breakpoints”, que significa que não houve regras ali de quebra de pontos específicos que o site, quando chegasse num determinado tamanho, deveria ter um comportamento diferente para que o layout não quebrasse como quebrou, quando vocês fizeram no curso 1. Aqui nesse gif eu explico de uma forma mais interativa como é que funciona.

02:18:18 - 02:49:18

Quando nós temos um site com “breakpoints”, com esses pontos de quebra, vocês podem perceber na imagem à esquerda que quando vocês estão visualizando em telas menores, o site fica proporcional: ele vai ficar um debaixo do outro, quebradinho em blocos, tudo certinho, sem problema nenhum de layout. Quando o site não tem esses pontos de quebra, o conteúdo simplesmente fica todo condensado, fica desproporcional, há a quebra de layout, como foi o que aconteceu no site da Ada e não fica da forma como deveria ser.

02:49:25 - 03:19:07

Um dos motivos, pelos quais eu comentei, é que não tem o “viewport”, que é a jornada de exibição que permite com que a gente trabalhe com responsividade e o outro tem a ver com essas regras do CSS nos quais a gente precisa trabalhar, que é a estrutura “media” ou “media queries”. As “medias queries” são propriedades que permitem que a gente adicione regras no CSS para que a gente consiga determinar o que nós vamos fazer ali.

03:19:26 - 03:49:06

Então, por exemplo, você quer fazer com que o seu site quando chegue em um determinado tamanho ou tiver acessando pelo celular, por exemplo, você quer que o título da sua página fique da cor vermelha. Se você escrevesse o CSS normal, ele não vai ficar na cor vermelha, porque não há uma regra. Então, quando você escreve uma folha de estilo sem o “media”, você está determinando que toda regra ali ela vai ser global, ou seja, ela vai valer pra todo mundo.

03:49:13 - 04:18:03

Só que no caso de um site responsivo, a gente precisa trabalhar com regras específicas até para o nosso site ficar fluído de uma forma coesa e não haver quebra de layout. Mas voltando aqui sobre o “media”, a gente tem aqui um modelo de estrutura, de como a gente tem que trabalhar com ele. Então, o “@media”, como eu falei, é uma regra que a gente vai determinar dentro da nossa folha de estilo. A gente tem o “media screen”, que é o tipo de regra, ou o “media type” como a gente chama.

04:18:10 - 04:40:14

Então, qual o tipo de regra que eu estou adicionando? Quando a gente deixa explícito o “media screen”, a gente está determinando para o navegador que aquela regra só vai ser adaptada para telas. Porque existem outros tipos... Por exemplo, a gente tem o “media print”, que é para quando houver casos de impressão de documento. Por exemplo, às vezes você precisa imprimir um documento da web:

04:40:24 - 05:02:18

Você tem uma regra de “media print” para quando você precisar imprimir um arquivo ali ou um documento da sua página do site, tenha um visual diferente, tenha uma exibição diferente. Hoje, na nossa aula, nós vamos falar só sobre o “media type screen”, que é o da tela. Mas tem uma outra forma também de fazer que eu vou explicar um pouquinho mais à frente.

05:02:28 - 05:29:03

Mas prosseguindo aqui, depois de a gente declarar o nosso tipo, a gente tem o operador, que costuma ser o “and”. Para essa aula, a gente vai trabalhar com esse. Então a gente está determinando aqui a “media” que a gente vai utilizar, que é apenas para as telas “e”, que em português “and” significa “e”, a gente tem a “feature”. Qual é a principal funcionalidade que a gente vai colocar dentro dessa regra?

05:29:13 - 05:51:28

A gente aqui, no caso, pode estar determinando uma largura máxima ou largura mínima, ou os dois juntos. Então a gente pode falar que o “max width” e o “min width” vão ter uma largura mínima e máxima dentro de uma única regra ou a gente pode trabalhar apenas com a largura máxima ou a largura mínima. E dentro dessas chaves a gente coloca as regras do CSS.

05:51:28 - 06:18:05

As classes, com as devidas customizações, de acordo com a necessidade. Pode estar aparecendo muito confuso aqui agora, mas eu te garanto que depois que a gente mostrar o exemplo através do código, vocês vão entender como é que vai funcionar o “media”. Lembra quando eu falei que existia uma outra forma de escrever o “media”? Pois bem, existe esse formato aqui, não necessariamente precisando especificar que é um “media” do tipo “screen”.

06:18:09 - 06:40:07

Por que eu digo isso? Porque, por padrão, a gente pode utilizar dessa forma, o que significa que a gente está abrangendo para qualquer tipo de “media” o comportamento dessas regras. A gente deixa, por exemplo, um “media screen” ou qualquer outro tipo quando é algo muito específico e a gente quer aquela experiência seja. É errado escrever “media screen”? Não, não é.

06:40:07 - 07:10:23

Você pode trabalhar com “media screen”, se você quiser. Mas nós vamos seguir mais ou menos o padrão que a gente trabalha geralmente, que é apenas o “@ media” e a gente determina largura máxima ou largura mínima e dentro dela a gente atribui as regras. Ok? Então a gente vai fazer agora um exemplo prático de como é a aplicação do uso “media” num projeto web para vocês entenderem os comportamentos e as transformações que acontecem a partir do momento que a gente adiciona uma regra e também um ponto de quebra, que são os famosos “breakpoints”.

07:10:28 - 07:12:22

Então vamos aqui, eu já estou com meu VS Code aberto.

07:13:22 - 07:41:18

Estou aqui com a minha pasta PrograMaria, com meu arquivo “index”. Já deixei meu “style” CSS já criado também dentro da pasta do projeto PrograMaria, pra gente otimizar o processo do desenvolvimento. Lembram daquela dica do “emmet”? Que a gente coloca lá a exclamaçãozinha, aperta tab ou enter e ele já cria nosso HTML. E se vocês perceberem, ele também já está criando aqui o nosso “viewport”, nossa janela de exibição.

07:41:18 - 08:10:20

Então não vou nem me preocupar com isso. O que eu vou fazer agora é adicionar a tag “link”. Outra dica também muito boa do “emmet” aqui com o VS Code é que se você digita “link”, olha as abreviações que aparecem: todos os formatos de link pra gente poder fazer importações. Então, como a gente vai usar o CSS, a gente vem aqui, vem com a setinha do teclado, escolhe a opção do CSS, aperta enter e olha que ele já faz...

08:10:29 - 08:31:08

Ele já determina qual é o tipo de relação desse “link” que está sendo vinculado, que é o tipo “folha de estilo”, como a gente viu na nossa aula 1, e a referência, que é o “style.css”. Por padrão, ele escreve o “style.css” e depois a gente customiza com outro nome. Deixei como “style.css” em nosso arquivo pra ficar mais fácil a interpretação.

08:31:16 - 09:08:04

Então eu vou colocar um título chamado “media queries” e ok. Vou criar o nosso documento, então eu vou criar uma “div”. Ah, uma outra dica também sobre o VS Code com o “emmet” é que a gente coloca aqui “div*3”, olha o que ele faz: ele vai criar três “divs” pra gente. É muito prático a gente trabalhar com isso, porque às vezes a gente precisa criar várias, uma lista de informações e aí tem que ficar escrevendo o “ul, li, li...”

09:08:04 - 09:31:25

E aqui, no caso, vou até fazer de novo para vocês verem, a gente coloca “div”, o nome da tag pode ser “link”, pode ser h1, pode ser h2, pode ser parágrafo, pode ser qualquer tag html. A gente coloca o asterisco, que é multiplicação - a gente vai multiplicar aqui a criação de tags - e o número. Então a gente quer que multiplique a criação de três divs aqui.

09:31:25 - 10:16:23

Então já fiz aqui a nossa criação e aí eu vou colocar um h2... Pera aí... Vou colocar a nossa tag h2, que é o nosso título, vou criar também um parágrafo aqui só de marcação: texto ocorrido. Vou copiar e colar. Então eu vou selecionar aqui a informação para não ter que ficar toda hora digitando, vamos vir aqui. Copio e colo, copio e colo e aí eu salvo aqui o nosso código para que a gente consiga visualizar.

10:17:03 - 10:44:24

Ok, criei o nosso arquivo aqui HTML, tudo certinho, ok. Vamos ver como é que ficou lá no navegador. Vamos aqui ver... Ó: texto corrido, texto corrido, texto corrido. Ok, um debaixo do outro. Então vamos agora criar customização. A gente vem aqui na “div”... Eu não estou trabalhando com classe, nada do tipo, pra ficar bem prático, bem simples pra vocês entenderem a questão de como funcionam as “media queries”.

10:44:26 - 11:12:26

Ok? Então a gente vai criar uma “div” e aí eu vou botar uma borda: “1px solid”. Relembrando aqui o CSS do que vocês viram lá no curso 1, “border” é a propriedade que permite a gente criar bordas, dos nossos blocos de elemento, de qualquer outro elemento, um botão, “input”, por exemplo. Então o “border” aqui é um atalho para a gente determinar a largura, por isso que a gente coloca “1px”.

11:13:13 - 11:40:25

O tipo dela: se a gente quer sólido, que é aquela linha reta, se a gente quer com bolinhas, que é o “dotted”, ou você quer com aquele tracinho. Então a gente determina aqui qual é o estilo dessa borda e a cor. Então eu vou trabalhar aqui com hexadecimal e colocar aqui, que é a cor preta. Então vou criar uma borda com uma largura de 1 pixel do tipo sólido e da cor preta.

11:41:17 - 12:07:29

Vou colocar também uma margem. Como vocês aprenderam na revisão, a margem é para a gente trabalhar espaçamentos externos de cada um dos blocos. Então eu vou determinar que essa margem vai ter 10 pixels de altura, tanto do “top” quanto do “bottom”, e vai ter um espaçamento de 20 pixels tanto do lado esquerdo e lado direito de cada bloco.

12:08:18 - 12:40:29

Ok? E aí nós temos o nosso “padding” aqui e nós vamos colocar o “padding” novamente, que é o espaçamento interno. Vamos colocar um valor de 20 pixels. Vamos ver como é que está ficando? Vai dar aquela atualizada. Já ficou aqui os bloquinhos das informações, um de baixo do outro, certo? Mas a gente quer que esse bloco fique um do lado do outro, porque na tela maior esse bloco tem que ficar do lado do outro.

12:40:29 - 13:06:27

Para isso, a gente usa o display “inline-block”. Lembra quando eu falei do “inline-block”, que permite que a gente consiga trabalhar com a ideia dos blocos, igual o caso do “display block”, porém utilizando a propriedade do “inline”, que é deixar um do lado do outro, e também com o “block” de a gente conseguir trabalhar mantendo o espaçamento, largura e altura de cada um desses blocos.

13:06:27 - 13:28:22

Então o “inline-block” aqui é o mais indicado. Então vamos ver como é que ficou? Vai dar uma atualização... Como vocês podem ver, temos três blocos aqui de informações, um do lado do outro. Agora, como é que a gente usa as “medias queries?” Então a gente criou aqui a “div” e aí a gente vai criar as “medias”. A gente tem aqui “@media”.

13:29:04 - 13:54:16

A gente vai trabalhar com o “max-width”. Então a gente vai determinar que nosso “max-width” vai ser de 768 pixels. O que eu estou querendo dizer aqui? Que eu estou determinando um ponto de quebra dentro dessa regra. Então eu estou usando o “media” e determinando uma regra que: qualquer coisa que tiver até a largura máxima de 768... Por que 768? Porque costuma ser o tamanho padrão de tablet no formato retrato, no formato vertical.

13:54:16 - 14:16:20

E acaba sendo um dos pontos de quebra, sendo bastante utilizado. Então a gente vai usar esse tamanho padrão aqui, que é bastante comum. Então a gente vai determinar para o “media” que quando ele chegar na largura máxima de 768, qualquer coisa que tiver até a largura máxima de 768, ele vai ter um comportamento diferente.

14:16:22 - 14:40:14

E qual é esse comportamento que nós vamos determinar? Aqui eu só vou usar o “border-color”, ok? Porque a gente só vai alterar a cor. Eu vou dizer que a minha borda é da cor verde, vou escrever assim para facilitar. Também vamos dizer que o nosso “display” agora vai ser “block”, ou seja, a gente vai botar as nossas caixinhas ali que a gente criou uma debaixo da outra.

14:40:15 - 15:10:25

Ok? Vamos também determinar que o nosso “padding” vai ter agora 10 pixels, vai ser um pouquinho menor. E a nossa margem... Eu vou botar aqui que a nossa margem vai ser de 5 pixels e 10 pixels. Então vamos ver como é que ficou agora? Aqui beleza, continua a mesma coisa no desktop, mas vamos ver no que vai ficar na tela menor?

15:11:14 - 15:54:08

Uma dica: para vocês testarem sites responsivos, se vocês estiverem usando o Google Chrome, cliquem sempre aqui com o mouse no lado direito e venha aqui em inspecionar. Está vendo? A gente foi no inspecionar e aí a gente viu aqui. Vamos dar só um “refresh”, salvei aqui... Como vocês podem ver, a borda verde aqui ficou bem fraca, pera aí que eu vou botar uma borda mais chamativa. Vamos ver aqui como ficou.

15:54:29 - 16:24:14

Como vocês já podem ver, coloquei a borda vermelha aqui e o site já ficou com um bloco debaixo do outro, que foi a regra que a gente determinou lá através das “medias queries”. Vou estender aqui para vocês verem. Passou dos 768, indo para um tamanho maior, aqui a gente já volta com os nossos blocos um do lado do outro.

16:24:25 - 16:52:24

Já passando dos 768 para diminuir, indo pra telas menores, a gente já tem esse novo comportamento, que é a borda da cor vermelha, as margens com valores menores, o “padding” também com valor menor. Então você pode ver que o nosso site já está devidamente responsivo, porque num celular a gente quer que nossos blocos tenham tamanhos menores, que fiquem um debaixo do outro e possam ficar fluídas sem ficar totalmente desproporcional ou algo do tipo.

16:53:03 - 17:16:06

E aí, quando tiver em telas maiores, a gente quer que o nosso bloco fique um do lado do outro. Ok? Vamos fazer agora a customização dos nossos textos para vocês também verem o impacto também da responsividade nesse cenário. Então a gente volta para o nosso VS Code, a gente vem aqui... Já vamos fazer aqui direto dentro do “media” para vocês verem a diferença.

17:16:16 - 17:43:02

A partir do momento que a gente adiciona esse ponto de quebra também para os nossos textos. No h2 eu vou botar uma fonte bem grande, vou botar um “font-size” de 36 pixels. Vou determinar também que a cor do nosso título é azul “blue”, e também vou botar o “text-decoration”. O “text-decoration” serve pra gente determinar como é que vai ser aquele estilo de linha marcando nosso texto.

17:43:18 - 18:11:13

Então por padrão, eu vou marcar o “underline”. Ah, isso aqui é uma dica também do VS Code: quando você digita, ele sempre tem um “autocomplete” aqui, que é algo completo que a gente está falando. A gente está escrevendo, no caso... Aí a gente escreve “un”, então já aparece o “underline” aqui como alternativa. Também vamos determinar que lá na nossa tag “p”, a gente vai ter um “color”: “red”.

18:11:25 - 18:45:16

Vamos colocar assim só para a gente ver impacto das fontes. Então vamos aqui atualizar. Pronto, já ficou agora do jeitinho, esse layout superbonito. Eu fiz dessa forma, gente, para facilitar o aprendizado e vocês entenderem também de uma forma bem brusca, bem explícita, como é que funciona as “media queries” e também os pontos de quebra. Então a gente já viu aqui, já está no 663 aqui, numa tela bem menor, numa largura de tela bem menor que o 768.

18:45:17 - 19:09:07

E vocês já viram um impacto aqui. Então a gente tem o título aqui de uma cor diferente, de tamanho diferente, com a decoração de texto, com isso aqui grifado aqui embaixo, o “underline”. A gente tem o nosso texto corrido já da cor vermelha e se a gente for arrastar para botar a tela maior, fica daquele padrão que a gente desenvolveu de um do lado do outro.

19:09:20 - 19:32:13

Aqui sem estilização no título, todo mundo bonitinho, todo mundo certinho, como tem que ser. Então a gente não tem quebras de layout aqui, porque as regras que a gente determinou no “media” é exatamente o que está sendo executado. Então, quando chegar no 768 para baixo, a gente está determinando que o nosso site vai ter esse comportamento.

19:32:16 - 20:03:25

Se a gente tiver acima de 768, a partir de 769, o nosso site já vai ter um outro comportamento, que são esses blocos um do lado do outro. Então, recapitulando o “media”: “o “media” é quando a gente precisa determinar regras dentro do nosso CSS. E com essas regras, a gente atribui os pontos de quebra. Os pontos de quebras podem ser tanto uma largura mínima quanto uma largura máxima. Aqui a gente está trabalhando com a largura máxima como exemplo.

20:04:17 - 20:32:22

Mas vocês podem botar uma largura mínima, por exemplo, também, porque vai funcionar da mesma forma. “Ah, Aline, eu posso usar o ‘media-screen’?” Pode usar o “media-screen”, porque quando a gente usa o “media-screen”, a gente está especificando que é apenas para a tela. Então, quando a gente especifica o tipo dessa “media”, é quando a gente quer algo muito específico, como eu falei lá anteriormente sobre o “media print”, que é quando a gente precisa para impressão de documento na tela do nosso navegador.

20:33:11 - 21:01:27

Mas como a gente não vai utilizar isso aqui, pelo menos nas nossas aulas e também no nosso projeto final, a gente vai utilizar apenas o “@media” que acaba sendo mais comum também no dia a dia de trabalho. Então vocês vão ver muito mais essa escrita do que as outras, “media screen” e tal. Pode usar, pode usar sem problema nenhum, mas a gente vai acabar utilizando mais do que a padrão de mercado, até para facilitar também o processo de aprendizado de vocês e também de aprofundamento dos estudos.

21:02:12 - 21:35:25

Então, novamente, “media” é quando a gente determina uma regra, é uma propriedade do CSS que nos permite criar regras de estilo: como o nosso site vai se comportar visualmente falando em determinados dispositivos, seja no celular, no tablet, computador. E os “break points”, que são os pontos de quebra, é onde a gente vai determinar, isso no nosso CSS, a partir de qual tamanho, a partir de qual cenário a gente quer que essas mudanças apareçam.

21:36:07 - 22:00:19

Bom, pessoal, então é isso. Hoje foi a nossa última aula de CCS aqui no nosso curso FrontEnd #2 do EuProgrAmo. Estou muito feliz em ver vocês até aqui, na metade do nosso caminho. Próximo módulo agora é de JavaScript que também tem muita novidade, muita coisa para vocês aprenderem. Espero que vocês tenham se divertido, aprendido muita coisa sobre responsividade, sobre CSS.

22:01:01 - 22:33:09

Eu espero que essa jornada esteja sendo muito tranquila para todos. E eu estou muito feliz mesmo, de verdade, de ver vocês por aqui se aprofundando cada vez mais, entendendo e aceitando também os desafios, porque eu sei que não deve estar fácil aí do outro lado. É muita informação, é normal. Tira um tempo também para absorver, vai fazendo os exercícios, vai procurando ler os materiais de apoio que a gente está sempre oferecendo. E assim, contem com a PrograMaria quando vocês precisarem, com as pessoas tutoras, se vocês tiverem dúvidas.

22:33:09 - 22:44:14

E eu vejo vocês agora no nosso próximo módulo, que é o de JavaScript. E é isso, gente, até lá! Tchau, tchau