

Capítulo 17 - Aula 02 - Responsividade

terça-feira, 11 de julho de 2023

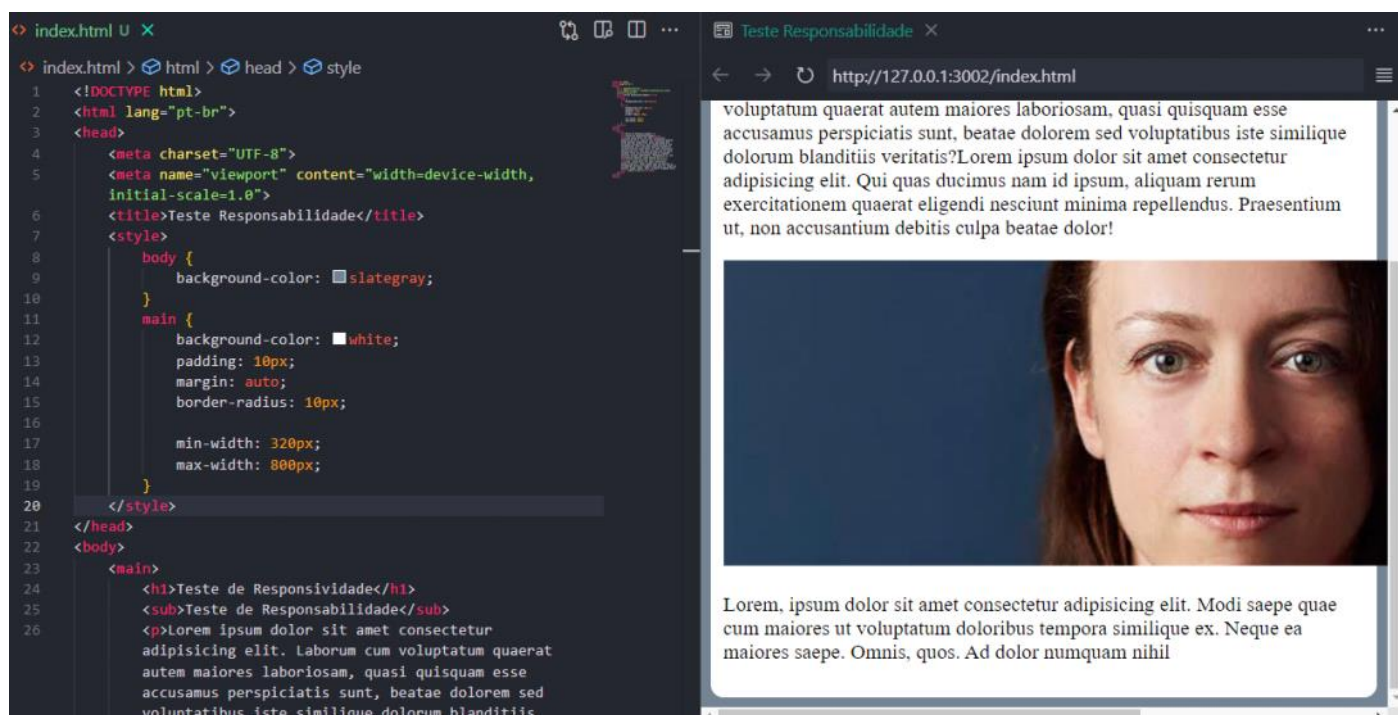
18:34

Telas e seus Múltiplos Tamanhos

O recomendado é manter a largura dos elementos em um campo de até 1200px (ou algo próximo a isso), principalmente para facilitar a leitura de textos. Isto porque quando o site é colocado em telas muito grandes, como a de uma TV, por exemplo, seria praticamente cansativo movimentar a cabeça para ler as informações da tela.

Determinando Medidas Mínimas e Máximas

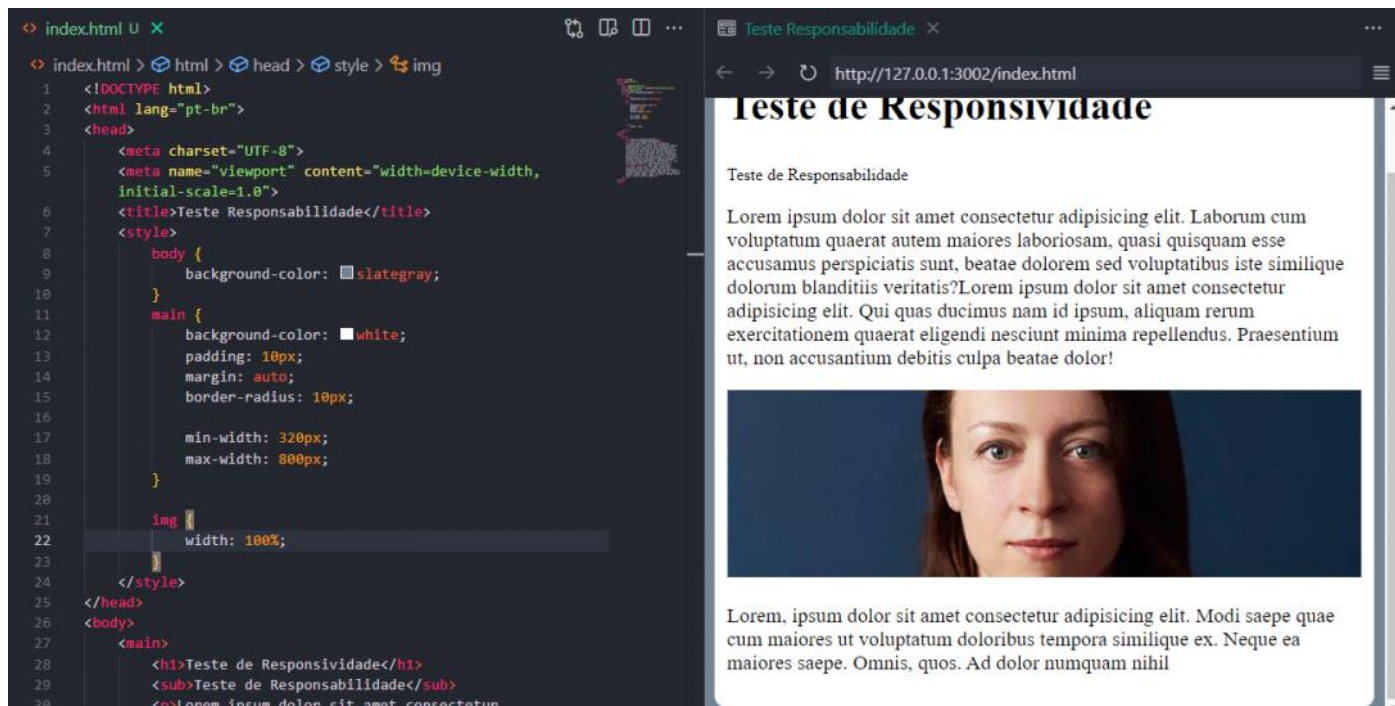
Algo que ajuda na hora de dimensionar os elementos de um site é estabelecer um comprimento mínimo (min-width) e máximo (max-width).



Neste exemplo, quando a tela atingisse um tamanho menor que o mínimo, a caixa "main" não se adaptaria mais e permaneceria com comprimento de 320px. A partir do momento que o comprimento da tela aumenta, a caixa aumenta até que o valor máximo de 800px seja atingido, quando já não aumenta mais seu tamanho.

A medida de 320px não veio do absoluto nada. Na verdade, alguns celulares mais antigos possuem esta medida. Então, por se tratarem de aparelhos que entrarão em desuso tão logo, podemos usá-los como modelo para a nossa responsividade. Uma dica é testar se a caixa fica à mostra da maneira correta. Caso contrário, o tamanho de 320px pode ser mudado.

Algo que pode ajudar a redimensionar também a imagem é colocar seu comprimento para 100%.



Assim, a imagem se adapta ao tamanho da tela. Algumas imagens pequenas ficam muito grandes quando usamos uma largura de 100%. Então, nestes casos, podemos colocá-la para 50% ou alguma porcentagem menor que 100% para ajustar o tamanho.

Se ainda assim perceber que o tamanho da imagem ficou estranho após os ajustes, experimente usar o `max-width` ou `min-width`. No caso da imagem do bugdroid (desafio 10), utilizamos um `max-width` para não aumentar demais a imagem e mantê-la no mesmo tamanho quando a tela diminuir.

Dica para centralizar imagens ou qualquer coisa que não se centralizam com "margin: auto;"

O elemento basicamente não está sendo compreendido como um bloco. Então, a solução é adicionar `"display: block"`.

```
main img {  
  width: 100%;  
  display: block;  
  margin: auto;  
}
```

Espaço entre linhas e seleção de tags dentro de outras tags

A propriedade `line-height` define o espaço entre linha de um texto.

Para escolher uma tag que está dentro de outra, em qualquer nível, não precisamos usar `">"` para selecioná-la, pois isto significa dizer que estou selecionando a tag um nível abaixo da "parent".

```
main p {
  text-align: justify;
  text-indent: 20px;
  line-height: 2em;
  margin: 20px 0px;
}

main img {
  width: 100%;
  display: block;
  margin: auto;
}
```

Dicas diversas

```
aside > ul {
  list-style-type: '\2705\00A0\00A0';
  list-style-position: inside;
  columns: 2;
}
```

- **List-style-position:** muda a posição das marcações de listas para dentro de uma caixa;
- **List-style-type:** permite mudar as marcações - usei cada "\" para gerar um emoji (\2705) e dois espaços (\00A0).
- **Columns:** cria colunas para as listas

✓ 1.5 - Cupcake	✓ 4.1 - Jelly Bean
✓ 1.6 - Donut	✓ 4.4 - KitKat
✓ 2.0 - Eclair	✓ 5.0 - Lollipop
✓ 2.2 - Froyo	✓ 6.0 - Marshmallow
✓ 2.3 - Gingerbread	✓ 7.0 - Nougat
✓ 3.0 - Honeycomb	✓ 8.0 - Oreo
✓ 4.0 - Ice Cream Sandwich	✓ 9.0 - Pie

```
main h2 {
  font-family: var(--fonte-android);
  font-size: 1.2em;
  color: var(--cor4);
  background-image: linear-gradient(to right, var(
    --cor1), transparent);
  padding: 3px;
}
```

Uso do transparent para adicionar uma cor transparente.

SURGE UM NOVO MASCOTE


```
aside > h3 {  
  color: white;  
  background-color: var(--cor4);  
  padding: 10px;  
  margin: -10px -10px 0px -10px;  
  border-radius: 10px 10px 0px 0px;  
}
```

Uso de margin negativa em cima, na direita e na esquerda para crescer uma caixa por dentro.

Quer aprender mais?

```
a.externo::after {  
  content: '\1F517';  
}
```

Adição de um emoji após um link da classe externo

Dan Morrill 

Responsividade de um vídeo incorporado

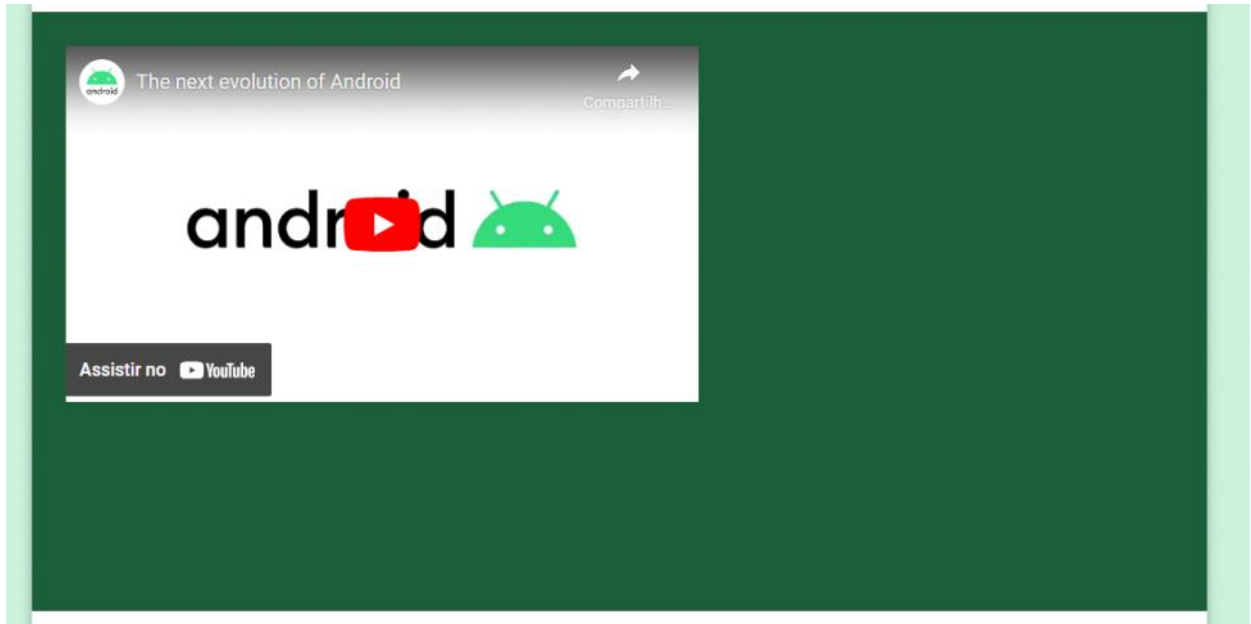
Como centralizar um vídeo?

Podemos fazer o seguinte: envelopar o iframe dentro de uma div. A partir daí, configuramos separadamente tanto a posição da div, com "position: relative", quanto a do iframe, com "position: absolute".

```
div.video {  
  background-color: var(--cor4);  
  padding: 30px;  
  margin: 0px -20px 30px -20px;  
  
  position: relative;  
}  
  
div.video > iframe {  
  position: absolute;  
}
```

A nossa div com o vídeo fica toda estranha, com tamanhos nitidamente incompatíveis. Para ajustar isto, colocamos um padding-bottom para 50%.

```
padding-bottom: 50%;
```



Isto só fica correto dependendo do tamanho da tela. Em telas menores, por enquanto, não funciona.

Com a "position: absolute", consigo configurar as propriedades "top" (distância entre a ponta superior esquerda do vídeo e o topo da div) e left (distância entre o lado esquerdo do vídeo e o lado esquerdo da div).

```
div.video > iframe {  
  position: absolute;  
  top: 5%;  
  left: 5%;  
  width: 90%;  
  height: 90%;  
}
```

E foi assim que o vídeo ficou centralizado. Com top e left em 5% e width e height a 90%.