FLEXBOX

**3 - Manipulando flex-itens**

**Objetivos da Aula**

* Mostrar as funcionalidades de organização dos itens
* Você entender melhor sobre como funciona a relação do item com o container

**Snippets**

Bom, vamos ver aqui como funciona a organização do flex-item, e para o que serve

1. Continuando com os itens que tinhamos na aula passada, só que sem wrap, sem alinhamentos, sem nada, com eles puros e apenas com o display flex.
2. .flex-container {
3. box-sizing: border-box;
4. padding: 10px;
5. background-color: black;
6. color: white;
7. min-height: 100vh;
8. display: flex;
9. gap: 5px;
10. }
11. .flex-item {
12. background-color: #ff0044;
13. }
14. Vamos então mexer com o flex-basis, que é o que mexe com a largura do item. Se eu colocar algum px em flex-basis, eu vou fazer com que a largura dos nossos itens fique com essa largura
15. .flex-item {
16. background-color: #ff0044;
17. flex-basis: 150px; /\* LINHA NOVA \*/
18. }
19. Vamos ver agora em relação ao flex-grow, o flex-grow é o aproveitamento do espaçamento extra que ficou. já que o flex-basis não utilizou 100%, o padrão dele é 0, pois ele não vai aproveitar nada desse espaçamento. Com 1 setado, ele vai fazer a divisão automática desses espaçamento, mas ele não funciona se você não tiver algum flex-basis para deixar como referência.

Agora, caso eu queira, posso fazer com que um item tenha um espaçamento maior do que os demais, apenas colocando um flex-grow específico para ele.

.flex-item {

background-color: #ff0044;

flex-basis: 150px;

flex-grow: 1;

}

1. Vamos criar um id no item 1 do container e vamos pegar o id dele para colocar um flex-grow diferente aqui no nosso css.

Mas você pode notar que se o espaçamento ficar menor, esse flex-grow em particular vai deixar de existir, por que nós temos um flex-basis, que da a base de largura para eles.

#flex-1 {

flex-grow: 3;

}

1. Vamos agora ver uma propriedade chamada flex-shrink, com ela, você vai setar a capacidade do objeto de encolher. Ou seja, se você colocar um flex-shrink em um, quando ele chegar naquele espaçamento, você vai vai fazer com que o selecionado diminua mais que os outros.
2. .flex-item {
3. background-color: #ff0044;
4. flex-basis: 250px; /\* MUDANÇA DE 150px PARA 250px \*/
5. flex-grow: 1;
6. }
7. #flex-1 {
8. flex-shrink: 3; /\* LINHA NOVA \*/
9. }
10. E nós também temos o flex. O flex nada mais é do que um atalho do que temos nos 3 acima, parecido com o flex-flow. Eles vão estar na seguinte ordem: Grow, Shrink, basis
11. .flex-item {
12. background-color: #ff0044;
13. flex: 1 1 250px;
14. }

Com isso, vimos as funcionalidades principais dos flex-itens, e da mesma forma que ficou na outra aula, pode ter ficado nessa também, sem clareza do que fazer e como usar. Mas calma, que como eu disse, quando formos para a prática você entenderá tudo da melhor forma!

**2 - Manipulando flex-container**

**Objetivos da Aula**

* Ver como usar o container do flexbox
* Ver algumas das funcionalidades do flex-container
* Ter os primeiros passos com o flexbox

**Snippets**

Artigo sobre box-sizing: <https://developer.mozilla.org/en-US/docs/Web/CSS/box-sizing>

1. Você vai criar uma pasta, dentro dela criar um index.html e um style.css e linkar um ao outro
2. <!DOCTYPE html>
3. <html lang="en">
4. <head>
5. <meta charset="UTF-8">
6. <meta http-equiv="X-UA-Compatible" content="IE=edge">
7. <meta name="viewport" content="width=device-width, initial-scale=1.0">
8. <link rel="stylesheet" href="./style.css">
9. <title>Css flex</title>
10. </head>
11. <body>
13. </body>
14. </html>
15. Agora vamos fazer o flex-container/flex-itens e também, vamos fazer a configuração base do css, para que tenhamos um ambiente pronto para nós
16. <div class="flex-container">
17. <div class="flex-item">
18. <p>FLEX-ITEM 1</p>
19. Lorem ipsum dolor sit, amet consectetur adipisicing elit. Natus cumque
20. repellat ab quas temporibus. Ipsa nobis debitis non adipisci sequi?
21. </div>
22. <div class="flex-item">
23. <p>FLEX-ITEM 2</p>
24. Lorem ipsum dolor, sit amet consectetur adipisicing elit. Quas, ex.
25. </div>
26. <div class="flex-item">
27. <p>FLEX-ITEM 3</p>
28. Lorem ipsum dolor sit amet consectetur adipisicing elit. Modi voluptates
29. veritatis asperiores nobis possimus iure! Sapiente consectetur
30. voluptatem necessitatibus aperiam.
31. </div>
32. <div class="flex-item">
33. <p>FLEX-ITEM 4</p>
34. Lorem ipsum dolor sit amet.
35. </div>
36. </div>
37. \* {
38. padding: 0;
39. margin: 0;
40. }
41. .flex-container {
42. box-sizing: border-box;
43. padding: 10px;
44. background-color: black;
45. color: white;
46. min-height: 100vh;
47. }
48. .flex-item {
49. background-color: #ff0044;
50. border: 1px solid white;
51. }
52. Agora, vamos vir aqui no css e criar o nosso flex container, dando simplesmente um “display: flex”, dentro da calss container
53. .flex-container {
54. box-sizing: border-box;
55. padding: 10px;
56. background-color: black;
57. color: white;
58. min-height: 100vh;
59. display: flex; /\*LINHA NOVA \*/
60. }
61. E agora podemos ver que todos os nossos flex-itens ficaram um ao lado do outro, vamos fazer dois movimentos importantes agora para deixar algo mais organizado:

1 - Tirar o border de flex-item

2 - Colocar um gap dentro de flex-container, que é um expaço utilizado dentro dos filhos do container

.flex-container {

box-sizing: border-box;

padding: 10px;

background-color: black;

color: white;

min-height: 100vh;

display: flex;

gap: 5px; /\* LINHA ADICIONADA \*/

}

.flex-item {

background-color: #ff0044;

/\* LINHA RETIRADA \*/

}

1. E agora nós já temos o nosso flex-container setado e configurado, vamos agora ver as propriedades de organização dele (Flex-direction, wrap e flow)

Começando pelo flex-direction. Como o próprio nome já diz, daremos direção aos nosso filhos.

flex-direction: column;

flex-direction: column-reverse;

flex-direction: row;

flex-direction: row-reverse;

1. Agora nós vamos mexer no nosso flex-wrap, que é o quebra com o flex, ou seja, vamos configurar a quebra, para ficar um abaixo do outro
2. flex-wrap: wrap;
3. flex-wrap: nowrap;
4. flex-wrap: wrap-reverse;
5. Agora vamos para o flex-flow, ele é basicamente um atalho para escrever de uma só vez os itens acima. Ou seja, você vai ter na mesma propriedade o flex-direction e o wrap.
6. flex-flow: row wrap;
7. flex-flow: row nowrap;
8. flex-flow: row wrap-reverse;
9. flex-flow: row-reverse nowrap;
10. Agora nós temos também propriedades de alinhamento, vamos colocar um height de 200px nos itens que temos e mexer com as propriedades de alinhamento
11. .flex-item {
12. background-color: #ff0044;
13. height: 200px; /\* LINHA NOVA \*/
14. }
15. .flex-container {
16. box-sizing: border-box;
17. padding: 10px;
18. background-color: black;
19. color: white;
20. min-height: 100vh;
21. display: flex;
22. gap: 5px;
23. flex-wrap: wrap;
24. justify-content: center; /\* LINHA NOVA \*/
25. align-items: center; /\* LINHA NOVA \*/
26. }

Com isso, finalizamos a nossa aula sobre as principais organizações do flex-container, possa ser que esteja um pouco confuso para você agora, vamos pode ficar tranquilo que quando formos para a prática, e você ver essas funcionalidades sendo utilizadas para o layout, ficará mais simples de se entender!

**4 - Fazendo um layout na prática com flexbox**

**Objetivos da Aula**

* Fazer um layout básico com flex-box, mostrando como usar na prática as propriedades aprendidas na aula

**Snippets**

O layout que vamos criar vai ser do 0, ou seja, vamos criar uma nova pasta e fazer a nova organização dos itens e container, vamos criar um layout baseado nesse aqui:

1. Crie uma pasta com html e css, dentro do html vamos ter 1 container e 4 flex-itens, pois eles serão os nossos elementos acima
2. <!DOCTYPE html>
3. <html lang="en">
4. <head>
5. <meta charset="UTF-8" />
6. <meta http-equiv="X-UA-Compatible" content="IE=edge" />
7. <meta name="viewport" content="width=device-width, initial-scale=1.0" />
8. <link rel="stylesheet" href="./style.css" />
9. <title>Css flexbox</title>
10. </head>
11. <body>
12. <div class="container">
13. <header>
14. <div class="logo"></div>
15. <ul class="navbar">
16. <a href="#" class="navItens">
17. <li>nav1</li>
18. </a>
19. <a href="#" class="navItens">
20. <li>nav2</li>
21. </a>
22. <a href="#" class="navItens">
23. <li>nav3</li>
24. </a>
25. </ul>
26. </header>
27. <div class="flex-container">
28. <div class="item">1</div>
29. <div class="item">2</div>
30. <div class="item">3</div>
31. <div class="item">4</div>
32. <div class="item">5</div>
33. <div class="item">6</div>
34. <div class="item">7</div>
35. <div class="item">8</div>
36. <div class="item">9</div>
37. </div>
38. <footer></footer>
39. </div>
40. </body>
41. </html>
42. Agora iremos fazer a organização de tamanhos e cores que iremos usar, mas sem mexer com o flex-box ainda
43. \* {
44. padding: 0;
45. margin: 0;
46. }
47. .container {
48. box-sizing: border-box;
49. background-color: black;
50. min-height: 100vh;
51. }
52. header {
53. height: 55px;
54. background-color: #bbb;
55. padding: 0px 20px;
56. }
57. .logo {
58. width: 45px;
59. height: 40px;
60. background-color: #ff0044;
61. border-radius: 10px;
62. }
63. .navbar {
64. list-style: none;
65. }
66. .navItens {
67. cursor: pointer;
68. text-decoration: none;
69. color: black;
70. }
71. .item {
72. max-width: 300px;
73. height: 250px;
74. background-color: #bbb;
75. }
76. footer {
77. width: 100%;
78. height: 45px;
79. background-color: #bbb;
80. }
81. Agora nós iremos fazer a organização do flex-container, para deixar fazer a organização dos flex-itens, que será pouca coisa nessa situação!
82. .flex-container {
83. box-sizing: border-box;
84. display: flex;
85. flex-wrap: wrap;
86. justify-content: center;
87. width: 900px;
88. gap: 20px;
89. margin: 20px auto;
90. }
91. Agora iremos fazer alterações nos itens, onde já declaramos eles lá em cima, só que não fizemos ainda as configurações necessárias para eles ficarem como devem
92. .item {
93. max-width: 250px;
94. height: 200px;
95. background-color: #bbb;
96. flex: 1 1 250px; /\* LINHA NOVA \*/
97. }
98. Caso você queira, podemos até centralizar os textos que estão dentro de item, colocando apenas display flex e também os alinhamentos que aprendemos, transformando os flex-itens em novos flex-container
99. .item {
100. max-width: 250px;
101. height: 200px;
102. background-color: #bbb;
103. flex: 1 1 250px;
104. display: flex; /\* LINHA NOVA \*/
105. align-items: center; /\* LINHA NOVA \*/
106. justify-content: center; /\* LINHA NOVA \*/
107. }