**Flexbox:** foi projetado como um modelo de layout unidimensional e como um método que pode oferecer distribuição de espaço entre itens em uma interface e recursos de alinhamento, os navegadores que tem suporte ao flexbox são chrome, internet explorer, mozilla, opera e safari.

**Flex Container:** é a tag que envolve os itens, é a tag que possui itens filhos, vai ser nela que iremos aplicar a propriedade **“display:flex”**, vai transformar todos os seus itens filhos em flex itens. Essa propriedade de display de container, pode ser feito em qualquer tipo de tag, seja ela div, span, h1, h2 ou até mesmo a tag de ancora/link <a>, a partir do momento que essa tag possui itens filhos, ela é passiva de ser aplicada a propriedade de display:flex. O display:flex possui propriedades relacionadas que são **display** que é o inicializador do container, o **flex-direction** que vai fazer o direcionamento dos itens, seja em linha, coluna, **flex-wrap** que se aplica em quebra de linha ou não, o **flex-flow** que é uma abreviação para o flex-direction e flex-wrap, o **justify-content** que vai alinhar os itens do container de acordo com sua direção, o **align-items** que vai alinhar os itens de acordo com seu eixo do container e **align-content** que vai alinhar as linhas desse container.

**Flex item:** são os elementos filhos diretos do flex container e esses elementos filhos também podem se tornar flex container, podendo reaproveitar esse layout. O flex item também possuem propriedades relacionadas que são **flex-grow** que vai definir o fator de crescimento, o **flex-basis** que vai definir o tamanho inicial do item antes da distribuição do espaço restante do container, o **flex-shirk** que define a capacidade de redução, o **flex** que é uma abreviação para as propriedades flex-grow basis e shirk, o **order** que é relacionado a ordem de distribuição e listagem destes itens e o **algin-self** que vai definir o alinhamento de um item específico do container.

**Estrutura básica do display:flex:** deve utilizar duas extensões no visual studeo code que é o HTML Snippets que vai tratar do auto complete, a outra extensão é o Live HTML Previewer que vai fazer a renderização dos elementos HTML em tela, não precisa abrir um arquivo HTML no navegador.

**Display: flex;** torna a tag um elemento do tipo flex container, e assim automaticamente todos os seus filhos diretos desta tag, tornam-se em flex items, a gente consegue aplicar a propriedade display: flex; para qualquer tipo de tag HTML, assim como conseguimos fazer com que os flex-itens também sejam flex container.

**<style>**

**.flex {** /\*é classe criada para o container\*/

**max-width: 300px;**

**padding: 10px;**

**border: 2px solid black;**

**display: flex**; /\*torna os elementos filhos/itens em caixas flexíveis\*/ **}**

**.item {** /\*essa class item vai ser aplicada aos filhos do container .flex criado acima\*/

**background-color: aqua;**

**margin: 5px;**

**}**

**</style>**

**</head>**

**<body>**

**<div class="flex">**

**<div class="item">item 1</div>** <!--escrever div.item\*3 irá criar essas 3 divs-->

**<div class="item">item 2</div>**

**<div class="item">item 3</div>**

**</div>**

**</body>**

**Estrutura básica do flex-direction:** é a propriedade que estabelece o eixo principal do container, definindo assim a direção que os flex items são colocados no flex container, são 2 eixos principais o eixo horizontal e o eixo vertical. Os eixos são **row** que é o eixo padrão à que é a direção do texto da esquerda para direita, o **row-reverse** é o sentido oposto à direção do texto traz ordenação da direita para esquerda, outro eixo é o eixo vertical chamado de **column** ordenação de cima para baixo em coluna única, o **column-reverse** a ordenação é inversa de baixo para cima. É muito comum que a gente precisa mudar a orientação dos nossos elementos em tela, quando estamos trabalhando com telas de tamanhos diferentes, então a cada dimensão de tela posso modificar as posições dos elementos de linha para coluna, facilitando assim nossa vida quando estiver-mos trabalhando com responsividade, o flex-container surge para resolver esse problema.

**<style>**

**.flex-container {**

**margin: 0;**

**padding: 0;**

**display: flex;** /\*torna os elementos filhos/itens em caixas flexíveis\*/

**list-style: none;** /\*ao criar uma lista, irá tirar os pontinhos ou os números na frente da lista\*/

**}**

**.flex-item {**

**background:** blue; /\*os flex-itens ficarão setados de um modo geral/genérico como azul, podendo ser setado a cor nas classes abaixo neutralizando a cor azul padrão\*/

**height: 50px;** /\*altura\*/

**width: 50px;** /\*largura\*/

**line-height: 50px**; /\*altura da linha\*/

**font-size: 20px;**

**color: white;**

**text-align: center**; /\*alinhamento do texto ao centro\*/

**margin: 5px;** /\*os itens que estão dentro dentro da flex-container, esse código é para separar os itens, sendo possível ver a separação ou os espaços entres os itens dentro da flex-container\*/

**}**

**.row {**

**flex-direction: row**; /\*irá organizar os itens da esquerda para direita em uma linha\*/

**}**

**.row-reverse {**

**flex-direction: row-reverse;** /\*os itens ficam alinhados ao lado oposto, da ordem da direita para esquerda, para delimitar essa área pode usar o código no .flex-item {max-width: 500px} que ficará mais ou menos no centro da tela\*/

**}**

**.row-reverse li {**

**background-color: red**; /\*aqui vou colorir o fundo dos itens\*/

**}**

**.column {**

**float: left;** /\*a coluna ficará na parte esquerda da página\*/

**flex-direction: column;** /\*os itens ficarão em uma coluna, de cima para baixo\*/

**}**

**.column li {**

**background-color: yellow;** /\*aqui vou colorir o fundo dos itens\*/

**}**

**.column-reverse {**

**float: right;** /\*a coluna ficará na direita da página\*/

**flex-direction: column-reverse;** /\*esse código fará com que a coluna fique na ordem de baixo para cima\*/

**}**

**.column-reverse li {**

**background-color: purple**; /\*aqui vou colorir o fundo dos itens\*/

**}**

**</style>**

**</head>**

**<body>**

**<ul class="flex-container row">**

**<li class="flex-item">1</li>** <!--li\*5 vai criar os 5 li-->

**<li class="flex-item">2</li>**

**<li class="flex-item">3</li>**

**<li class="flex-item">4</li>**

**<li class="flex-item">5</li>**

**</ul>**

**<ul class="flex-container row-reverse">**

**<li class="flex-item">1</li>** <!--li\*5 vai criar os 5 li-->

**<li class="flex-item">2</li>**

**<li class="flex-item">3</li>**

**<li class="flex-item">4</li>**

**<li class="flex-item">5</li>**

**</ul>**

**<ul class="flex-container column">**

**<li class="flex-item">1</li>** <!--li\*5 vai criar os 5 li-->

**<li class="flex-item">2</li>**

**<li class="flex-item">3</li>**

**<li class="flex-item">4</li>**

**<li class="flex-item">5</li>**

**</ul>**

**<ul class="flex-container column-reverse">**

**<li class="flex-item">1</li>** <!--li\*5 vai criar os 5 li-->

**<li class="flex-item">2</li>**

**<li class="flex-item">3</li>**

**<li class="flex-item">4</li>**

**<li class="flex-item">5</li>**

**</ul>**

**</body>**

**Estrutura básica de flex-wrap:** é a propriedade que define se os itens devem ou não quebrar a linha, por padrão eles não quebram linhas, isso faz com que os flex itens sejam compactados além do limite do conteúdo, então wrap vai cuidar desse comportamento, se os itens devem ou não passar para a próxima linha, por padrão os itens não são quebrados, vão fazer com que sejam compactados no limite de seu conteúdo e também no limite do container. A partir do momento que o container não suporte mais esses itens, esses itens vão começar a vazar para fora do container. **Nowrap** é o padrão, não permite a quebra de linha, elementos vazando do container, começa a ter problemas de interface, tem situações de deve delimitar o container e que os itens fiquem dentro dele, ai que surge o wrap. **Wrap** permite a quebra de linha assim que um dos flex itens não puder mais ser compactados, o elemento que não consegue ficar dentro do container, que não possa mais ser compactado, esse item vai para a linha de baixo, não tendo a problemática do item vazando do container, porém terá problemas com espaçamento. **Wrap-reverse** permite a quebra de linha assim que um dos flex itens não puder mais ser compactado, porém na direção contrária da linha, da linha de baixo para cima.

**<style>**

**.flex-container {**

**display: flex;** /\*torna os elementos filhos/itens em caixas flexíveis\*/

**height: 150px;** /\*container com 150px de altura\*/

**border: 2px solid black;** /\*adiciona a borda no container\*/

**max-width: 300px;** /\*largura máxima do container é de 300px\*/

**}**

/\*

**.flex-container {**

**display: flex;** /\*torna os elementos filhos/itens em caixas flexíveis\*/

**border: 2px solid black;**

**max-width: 300px;**

**flex-direction: column;** /\*o container fica em forma de coluna, os itens ficam um embaixo do outro\*/

.item { /\*elemento filho\*/

**font-size: 24px;**

**height: 50%;**

**text-align: center;**

**min-width: 100px;** /\*largura nesse caso dos itens, como tem 3 itens setado na div abaixo, irá ficar no limite do container que foi setado 300px de laargura, se acrescentar mais 1 item, mais uma div, esse item ficará fora do container, porque precisaria que o container tivesse 400px de largura, o mesmo acontece com a altura\*/

**line-height: 50px**; /\*o conteúdo de dentro da flex vai ter um espaaço de 50px em cima dando uma centralizada na vertical\*/

**}**

**.nowrap {**

**flex-wrap: nowrap;** /\*código para fazer a quebra ou não de linha, nowrap é para não fazer quebra de linha\*/

**}**

**.wrap {**

**flex-wrap: wrap;** /\*código para fazer a quebra de linha, wrap é para fazer a quebra de linha\*/

**}**

**.wrap-reverse {**

**flex-wrap: wrap-reverse;** /\*faz a quebra de linha de baixo para cima\*/

**}**

**.blue {**

**background-color: blue;**

**}**

**.green {**

**background-color: lightgreen;**

**}**

**.orange {**

**background-color: orange;**

**}**

**</style>**

**</head>**

**<body>**

**<h1>no wrap</h1>**

**<div class="flex-container">**

**<div class="item orange">1</div>** <!--div.item\*3 já vai criar 3 divis com a class="item" automaticamente-->

**<div class="item green">2</div>**

**<div class="item blue">3</div>**

**<div class="item blue">3</div>**

**</div>**

**<h1>Wrap</h1>**

**<div class="flex-container wrap">** <!--nesse caso terá a quebra de linha, porque foi setador a class wrap, a largura máxima de flex-container é de 300px, cada item ta setado com 100 de largura, como tem 3 itens o quarto item/div descerá para linha de baixo-->

**<div class="item green">1</div>**

**<div class="item blue">2</div>**

**<div class="item orange">3</div>**

**<div class="item blue">3</div>**

**</div>**

**<h1>nowrap</h1>**

**<div class="flex-container wrap-reverse">**

**<div class="item green">1</div>**

**<div class="item blue">2</div>**

**<div class="item orange">3</div>**

**<div class="item orange">4</div>**

**</div>**

**</body>**

**Flex-flow:** é um atalho para as propriedades **flex-direction** e **flex-wrap,** flex-flow vai comprimir 2 escritas, porém seu uso não é tão comum, visto que, quando mudamos o flex-direction para column, mantemos o padrão do flex-wrap que é nowrap, ou seja, mantém o comportamento padrão esperado.

**<style>**

**.flex-container {**

**display: flex;** /\*torna os elementos filhos/itens em caixas flexíveis\*/

**flex-flow: row nowrap;** /\*mantem o elemento um do lado do outro sem quebra de linha\*/

**border: 1px solid black;**

**max-width: 500px;**

**margin-bottom: 10px;**

**}**

**.default {**

**flex-flow: row nowrap;** /\*mantem o elemento um do lado do outro sem quebra de linha\*/

**}**

**.wrap {**

**flex-flow: row wrap;** /\*faz a quebra do elemento para a próxima linha\*/

**}**

**.wrap-reverse {**

**flex-flow: row wrap-reverse**; /\*faz a quebra de linha do elemento de baixo para cima\*/

**}**

**.r-nowrap {**

**flex-flow: row-reverse nowrap**; /\*inverte a sequencia dos itens, começa da direita para esquerda, porém sem quebra de linha\*/

**}**

**.r-wrap {**

**flex-flow: row-reverse wrap;** /\*inverte a sequencia dos itens, começa da direita para a esquerda, porém tem quebra de linha\*/

**}**

**.rr-wrap {**

**flex-flow: row-reverse wrap-reverse;** /\*itens inicializam da direita para a esquerda, a quebra de linha de baico para cima\*/

**}** /\*no lugar do row pode ser também column, ficará em coluna, limitando a altura, faz a quebra da coluna\*/

**.item {**

**background-color: blueviolet;**

**margin: 5px;**

**color: white;**

**text-align: center;**

**width: 100px;** /\*largura do item flex\*/

**height: 50px;** /\*altura do item flex\*/

**line-height: 50px;** /\*altura da linha\*/

**}**

**</style>**

**</head>**

**<body>**

**<h1>default: row nowrap</h1>**

**<div class="flex-container default">**

**<div class="item">1</div>**

**<div class="item">2</div>**

**<div class="item">3</div>**

**<div class="item">4</div>**

**<div class="item">5</div>**

**</div>**

**<h1>wrap</h1>**

**<div class="flex-container wrap">**

**<div class="item">1</div>**

**<div class="item">2</div>**

**<div class="item">3</div>**

**<div class="item">4</div>**

**<div class="item">5</div>**

**<div class="item">6</div>**

**</div>**

**<h1>Wrap reverse</h1>**

**<div class="flex-container wrap-reverse">**

**<div class="item">1</div>**

**<div class="item">2</div>**

**<div class="item">3</div>**

**<div class="item">4</div>**

**<div class="item">5</div>**

**<div class="item">6</div>**

**</div>**

**<h1>Wrap reverse</h1>**

**<div class="flex-container wrap-reverse">**

**<div class="item">1</div>**

**<div class="item">2</div>**

**<div class="item">3</div>**

**<div class="item">4</div>**

**<div class="item">5</div>**

**<div class="item">6</div>**

**</div>**

**<h1>row-reverse nowrap</h1>**

**<div class="flex-container r-nowrap">**

**<div class="item">1</div>**

**<div class="item">2</div>**

**<div class="item">3</div>**

**<div class="item">4</div>**

**<div class="item">5</div>**

**<div class="item">6</div>**

**</div>**

**<h1>row-reverse wrap-reverse</h1>**

**<div class="flex-container rr-wrap">**

**<div class="item">1</div>**

**<div class="item">2</div>**

**<div class="item">3</div>**

**<div class="item">4</div>**

**<div class="item">5</div>**

**<div class="item">6</div>**

**</div>**

**</body>**

**</html>**

**Justify content:** essa propriedade vai se encarregar de alinhar os itens dentro do container de acordo com a direção pretendida e tratar da distribuição de espaçamento entre eles, se tem elementos que ocupam 100% de todo o container, justify content não se aplica, o objetivo do justify é distribuir os espaçamentos e fazer um alinhamento na interface. Justify content tem algumas variações, começando pelo **flex-start** que vai fazer o alinhamento no início desse container, **flex-end** que vai levar todos os itens para o final do container, **center** que vai trazer todos elementos para o centro do container, **space-between** que vai criar um espaço igual entre os elementos, vai pegar o primeiro elemento e colocar próximo a borda esquerda do container e o outro elemento vai colocar próximo ao fim da borda do container, tem o **space-around** os espaçamentos do meio são duas vezes maiores que o inicial e final.

**<style>**

**.container {**

**max-width: 400px;** /\*largura máxima\*/

**border: 2px solid black;** /\*borda do container\*/

**margin: 0;**

**display: flex;** /\*torna os elementos filhos/itens em caixas flexíveis\*/

**}**

**.item {**

**background-color: blueviolet;**

**margin: 5px;** /\*vai dar espaçamento entre os itens\*/

**text-align: center;**

**color: white;**

**}**

**.flex-start {**

**justify-content: flex-start**; /\*posiciona o item no começo\*/

**}**

**.flex-end {**

**justify-content: flex-end;** /\*posiciona o item no final\*/

**}**

**.center {**

**justify-content:** center; /\*posiciona o item e centraliza\*/

**}**

**.space-between {**

**justify-content: space-between;** /\*vai posicionar o primeiro e o ultimo item bem próximo a borda do container e colocar espaçamento entre os itens iguais\*/

**}**

**.space-around {**

**justify-content: space-around;** /\*vai posicionar o primeiro e o ultimo próximo a borda do container e colocar espaçamento entre os itens, nesse caso não fica tão próximo a borda do container comparado ao space-between\*/

**}**

**.column {**

**flex-direction: column;** /\*pode utilizar o justify-content com o flex-directin column, ficará em coluna, podendo ficar centralizado, o item posicionado no começo, no final\*/

**}**

**</style>**

**</head>**

**<body>**

**<h1>flex-start</h1>**

**<section class="container flex-start">**

**<div class="item">1</div>**

**<div class="item">item 2</div>**

**<div class="item">outro item</div>**

**</section>**

**<h1>flex-end</h1>**

**<section class="container flex-end">**

**<div class="item">1</div>**

**<div class="item">item 2</div>**

**<div class="item">outro item</div>**

**</section>**

**<h1>center</h1>**

**<section class="container center">**

**<div class="item">1</div>**

**<div class="item">item 2</div>**

**<div class="item">outro item</div>**

**<div class="item">4</div>**

**<div class="item">5</div>**

**</section>**

**<h1>space-between</h1>**

**<section class="container space-between">**

**<div class="item">1</div>**

**<div class="item">item 2</div>**

**<div class="item">outro item</div>**

**<div class="item">4</div>**

**<div class="item">5</div>**

**</section>**

**<h1>space-around</h1>**

**<section class="container space-around">**

**<div class="item">1</div>**

**<div class="item">item 2</div>**

**<div class="item">outro item</div>**

**<div class="item">4</div>**

**<div class="item">5</div>**

**</section>**

**<h1>flex-start em column</h1>**

**<section class="container flex-start column">**

**<div class="item">1</div>**

**<div class="item">item 2</div>**

**<div class="item">outro item</div>**

**</section>**

**</body>**

**Align-items:** trata do alinhamento dos flex itens de acordo com o eixo do container. O alinhamento é diferente para quando os itens estão em colunas ou linhas, align-items permite o alinhamento central no eixo vertical. Os tipos de alinhamento do align-items são **center** que faz o alinhamento dos itens ao centro, **stretch** que é o alinhamento padrão, vai fazer com que os itens cresçam igualmente, **flex-start** faz alinhamento dos itens no início,  **flex-end** faz o alinhamento dos itens no final, **baseline** faz o alinhamento de acordo com a linha base da tipografia dos itens.

**<style>**

**.container {**

**max-width: 350px;**

**border: 1px solid black;**

**display: flex;** /\*torna os elementos filhos/itens em caixas flexíveis\*/

**.item {**

**background-color: blueviolet;**

**margin: 5px;**

**color: white;**

**text-align: center;**

**flex: 1;** /\*esse código vai tratar da distribuição dos itens, precisa desse código para centralizar os itens em aligm-items\*/

**}**

**.center {**

**align-items: center;** /\*irá centralizar o item dentro do container\*/

**}**

**.stretch {**

**align-items: stretch;** /\*fará com que a margen dos itens menores crescam, que fiquem do tamanho da margem do item maior, vai crescer de acordo com o maior conteúdo\*/

**}**

**.flex-start {**

**align-items: flex-start;** /\*ira alinhar os itens na margem de cima do container\*/

**}**

**.flex-end {**

**align-items: flex-end;** /\*irá alinhar os itens na margem de baixo do container\*/

**}**

**.baseline {**

**align-items: baseline;**

**}**

**.column {**

**flex-direction: column;** /\*ira alinhar os itens em coluna\*/

**}**

**.central {**

**height: 300px;**

**justify-content: center;** /\*vai alinhar o item no centro\*/

**}**

**.central .item {**

**flex: 0;** /\*vai remover algumas propriedades de flex items, fazendo com que fique centralizado o item no container, centralizado em cima, embaixo e doss lados, isso é para quando quer alinha 1 item\*/

**padding: 20px;**

**}**

**</style>**

**</head>**

**<body>**

**<h1>row</h1>**

**<h1>center</h1>**

**<section class="container center">**

**<div class="item">Item 1</div>**

**<div class="item">Lorem ipsum dolor sit amet</div>**

**<div class="item">Item 3</div>**

**</section>**

**<h1>stretch</h1>**

**<section class="container stretch">**

**<div class="item">Item 1</div>**

**<div class="item">Lorem ipsum dolor sit amet</div>**

**<div class="item">Item 3</div>**

**</section>**

**<h1>flex-start</h1>**

**<section class="container flex-start">**

**<div class="item">Item 1</div>**

**<div class="item">Lorem ipsum dolor sit amet</div>**

**<div class="item">Item 3</div>**

**</section>**

**<h1>flex-end</h1>**

**<section class="container flex-end">**

**<div class="item">Item 1</div>**

**<div class="item">Lorem ipsum dolor sit amet</div>**

**<div class="item">Item 3</div>**

**</section>**

**<h1>baseline</h1>**

**<section class="container baseline">**

**<div class="item"><br>Item 1</div>**

**<div class="item">Lorem ipsum dolor sit amet</div>**

**<div class="item">Item 3</div>**

**</section>**

**<h1>flex-start em column</h1>**

**<section class="container flex-start column">**

**<div class="item">Item 1</div>**

**<div class="item">Lorem ipsum dolor sit amet</div>**

**<div class="item">Item 3</div>**

**</section>**

**<h1>cflex-end em column</h1>**

**<section class="container flex-end column">**

**<div class="item">Item 1</div>**

**<div class="item">Lorem ipsum dolor sit amet</div>**

**<div class="item">Item 3</div>**

**</section>**

**<h1>center em column</h1>**

**<section class="container center column">**

**<div class="item">Item 1</div>**

**<div class="item">Lorem ipsum dolor sit amet</div>**

**<div class="item">Item 3</div>**

**</section>**

**<h1>baseline em column</h1>**

**<section class="container baseline column">**

**<div class="item">Item 1</div>**

**<div class="item">Lorem ipsum dolor sit amet</div>**

**<div class="item">Item 3</div>**

**</section>**

**<h1>alinhamento central a tela</h1>**

**<section class="container central center">**

**<div class="item">Lorem ipsum dolor sit amet consectetur adipisicing elit.</div>**

**</section>**

**</body>**

**Align-content:** essa é aa última propriedade relacionada ao flex container, é a propriedade responsável por tratar o alinhamento das linhas do container em relação ao eixo vertical do container, vai trabalhar de fato as linhas do container e precisamos que o container utilize quebra de linhas, a altura do container seja maior que a soma das linhas dos itens. Os tipos de alinhamento no aligin-content é o **center** vai fazer o alinhamento dos itens ao centro, **stretch** é o padrão e os flex itens crescem igualmente, **flex-start** faz o alinhamento dos itens no início, **flex-end** faz o alinhamento dos itens no final, **space-around** os espaçamentos do meio são duas vezes maiores que o inicial e final, por ultimo o **space-between** que cria um espaço igual entre os elementos.

**<style>**

**.container {**

**height: 400px;**

**max-width: 300px;**

**display: flex;** /\*torna os elementos filhos/itens em caixas flexíveis\*/

**flex-wrap: wrap;**

**border: 1px solid black;**

**}**

**.item {**

**background-color: orangered;**

**padding: 0 5px;**

**margin: 5px;**

**flex: 1;** /\*esse código vai tratar da distribuição dos itens, precisa desse código para centralizar os itens em aligm-content\*/

**text-align: center;**

**color: white;**

**}**

**.stretch {**

**align-content: stretch;** /\*é o tipo de alinhamento padrão\*/

**}**

**.center {**

**align-content: center;** /\*irá centralizar os itens no meio do container, e os itens ficarão com a margem menor\*/

**}**

**.flex-start {**

**align-content: flex-start;** /\*vai levar os itens para parte de cima/inicial do container\*/

**}**

**.flex-end {**

**align-content: flex-end;** /\*vai colocar os itens na parte de baixo/final do container\*/

**}**

**.space-around {**

**align-content: space-around;** /\*terá espaço em cima e embaixo, porém o espaço do meio será maior\*/

**}**

**.space-between {**

**align-content: space-between;** /\*os itens de cime e de baixo ficarão colados a margem do container e os itens do meio ficarão espaçados igualmente\*/

**}**

**</style>**

**</head>**

**<body>**

**<h1>stretch</h1>**

**<div class="container stretch">**

**<div class="item">Lorem ipsum dolor sit amet .</div>**

**<div class="item">Lorem ipsum dolor </div>**

**<div class="item">Lorem ipsum dolor sit amet </div>**

**<div class="item">Lorem ipsum dolor, sit amet </div>**

**<div class="item">Lorem, ipsum dolor sit amet consectetur adipisicing elit</div>**

**<div class="item">Lorem ipsum, dolor sit .</div>**

**<div class="item">Lorem ipsum dolor sit </div>**

**</div>**

**<h1>center</h1>**

**<div class="container center">**

**<div class="item">Lorem ipsum dolor sit amet .</div>**

**<div class="item">Lorem ipsum dolor </div>**

**<div class="item">Lorem ipsum dolor sit amet </div>**

**<div class="item">Lorem ipsum dolor, sit amet </div>**

**<div class="item">Lorem, ipsum dolor sit amet consectetur adipisicing elit</div>**

**<div class="item">Lorem ipsum, dolor sit .</div>**

**<div class="item">Lorem ipsum dolor sit </div>**

**</div>**

**<h1>flex-start</h1>**

**<div class="container flex-start">**

**<div class="item">Lorem ipsum dolor sit amet .</div>**

**<div class="item">Lorem ipsum dolor </div>**

**<div class="item">Lorem ipsum dolor sit amet </div>**

**<div class="item">Lorem ipsum dolor, sit amet </div>**

**<div class="item">Lorem, ipsum dolor sit amet consectetur adipisicing elit</div>**

**<div class="item">Lorem ipsum, dolor sit .</div>**

**<div class="item">Lorem ipsum dolor sit </div>**

**</div>**

**<h1>center</h1>**

**<div class="container flex-end">**

**<div class="item">Lorem ipsum dolor sit amet .</div>**

**<div class="item">Lorem ipsum dolor </div>**

**<div class="item">Lorem ipsum dolor sit amet </div>**

**<div class="item">Lorem ipsum dolor, sit amet </div>**

**<div class="item">Lorem, ipsum dolor sit amet consectetur adipisicing elit</div>**

**<div class="item">Lorem ipsum, dolor sit .</div>**

**<div class="item">Lorem ipsum dolor sit </div>**

**</div>**

**<h1>center</h1>**

**<div class="container space-around">**

**<div class="item">Lorem ipsum dolor sit amet .</div>**

**<div class="item">Lorem ipsum dolor </div>**

**<div class="item">Lorem ipsum dolor sit amet </div>**

**<div class="item">Lorem ipsum dolor, sit amet </div>**

**<div class="item">Lorem, ipsum dolor sit amet consectetur adipisicing elit</div>**

**<div class="item">Lorem ipsum, dolor sit .</div>**

**<div class="item">Lorem ipsum dolor sit </div>**

**</div>**

**<h1>space-between</h1>**

**<div class="container space-between">**

**<div class="item">Lorem ipsum dolor sit amet .</div>**

**<div class="item">Lorem ipsum dolor </div>**

**<div class="item">Lorem ipsum dolor sit amet </div>**

**<div class="item">Lorem ipsum dolor, sit amet </div>**

**<div class="item">Lorem, ipsum dolor sit amet consectetur adipisicing elit</div>**

**<div class="item">Lorem ipsum, dolor sit .</div>**

**<div class="item">Lorem ipsum dolor sit </div>**

**</body>**

**Flex-itens**

**Flex-grow:** define a proporcionalidade de crescimentos dos itens, respeitando o tamanho de seus conteúdos internos, a box ficará do tamanho do conteúdo interno, **não irá funcionar casso tenhamos adicionado justify-content ao nosso flex container.**

**<style>**

**.flex-container {**

**display: flex;** /\*torna os elementos filhos/itens em caixas flexíveis\*/

**border: 2px solid black;** /\*para a borda do container ficar evidente\*/

**max-width: 300px;**

**}**

**.item {**

**background-color: orangered;**

**margin: 5px;**

**padding: 10px;**

**text-align: center;**

**color: white;**

**}**

**.flex-grow-0 { /\***todas as box terão tamanhos iguais**\*/**

**flex-grow: 0;**

**}**

**.flex-grow-1 { /\***misturar flex-grow 1, 2, 3 a box terá tamanhos diferentes**\*/**

**flex-grow: 1;**

**}**

**.flex-grow-2 {**

**flex-grow: 2;**

**}**

**.flex-grow-3 {**

**flex-grow: 3;**

**}**

**</style>**

**</head>**

**<body>**

**<h1>flex grow 0</h1>**

**<div class="flex-container">**

**<div class="item flex-grow-0">A</div>**

**<div class="item flex-grow-0">B</div>**

**<div class="item flex-grow-0">C</div>**

**<div class="item flex-grow-0">D</div>**

**</div>**

**<h1>flex grow 1</h1>**

**<div class="flex-container">**

**<div class="item flex-grow-1">A</div>**

**<div class="item flex-grow-1">B</div>**

**<div class="item flex-grow-1">C</div>**

**<div class="item flex-grow-1">D</div>**

**</div>**

**<h1>flex grow alternados</h1>**

**<div class="flex-container">**

**<div class="item flex-grow-1">A</div>**

**<div class="item flex-grow-2">B</div>**

**<div class="item flex-grow-2">C</div>**

**<div class="item flex-grow-1">D</div>**

**</div>**

**<h1>flex grow alternados</h1>**

**<div class="flex-container">**

**<div class="item flex-grow-2">AAAAA</div>**

**<div class="item flex-grow-2">BBBB</div>**

**<div class="item flex-grow-2">C</div>**

**<div class="item flex-grow-2">D</div>**

**</div>**

**</body>**

**Flex-basis:** é a propriedade que estabelece o tamanho inicial do item antes das distribuições de espaço restante dentro dele, usando como base o conteúdo interno disposto. Os valores possíveis são **auto** caso o item não tenha tamanho, este será proporcional ao conteúdo do item, **px**, **%, em,** são valores exatos previamente definidos se o conteúdo for maior ao px, % ou em definido, o conteúdo irá vazar do item, **0(zero)** terá relação com a definição do flex-grow.

**<style>**

**.flex-container {**

**max-width: 500px;**

**border: 1px solid black;**

**display: flex;** /\*torna os elementos filhos/itens em caixas flexíveis\*/

**}**

**.items {**

**background-color: coral;**

**color: white;**

**text-align: center;**

**margin: 5px;**

**}**

**.bauto {**

**flex-basis: auto;** /\*vai se adaptar ao tamanho do conteúdo do item automaticamente, vai tentar ocupar todo o restante do espaço dentro do container\*/

**}**

**.bzero {**

**flex-basis:** 0; /\*faz com que todos os ítens terão o mesmo tamanho, até que não possa mais comprimir esse conteúdo e o item crescer\*/

**}**

**.flex-grow1 {**

**flex-grow: 1;** /\*código de fator de crescimento do item\*/

**}**

**.flex-grow2 {**

**flex-grow: 2;** /\*nesse caso vai multiplicar o grow em 2x\*/

**}**

**.width100 {**

**flex-basis: 100px;** /\*cada item terá 100px\*/

**}**

**</style>**

**</head>**

**<body>**

**<h1>basis auto</h1>**

**<div class="flex-container">**

**<div class="items bauto flex-grow1">A</div>**

**<div class="items bauto flex-grow1">BBB</div>**

**<div class="items bauto flex-grow1">C</div>**

**<div class="items bauto flex-grow2">DEF</div>**

**</div>**

**<h1>basis 0</h1>**

**<div class="flex-container">**

**<div class="items bzero flex-grow1">AAAAAAA</div>**

**<div class="items bzero flex-grow1">BBB</div>**

**<div class="items bzero flex-grow1">C</div>**

**<div class="items bzero flex-grow1">DEF</div>**

**</div>**

**<h1>basis + width</h1>**

**<div class="flex-container">**

**<div class="items width100 flex-grow1">AA</div>**

**<div class="items width100 flex-grow2">BBB</div>**

**<div class="items width100 flex-grow1">CCCC</div>**

**<div class="items width100 flex-grow1">DEF</div>**

**</div>**

**</body>**

**Flex-shrink:** é a propriedade de estabelecer a capacidade de redução ou compressão do tamanho de um item.

**<style>**

**.flex-container {**

**display: flex;** /\*torna os elementos filhos/itens em caixas flexíveis\*/

**border: 1px solid black;**

**max-width: 350px;**

**}**

**.item {**

**background-color: orangered;**

**font-size: 24px;**

**margin: 5px;**

**text-align: center;**

**}**

**.shrink1 {**

**flex-shrink: 1;** /\*vai comprimir o conteúdo para caber dentro do container\*/

**}**

**.shrink0 {**

**flex-shrink: 0;** /\*vai comprimir porém não vai permitir a compressão dos itens, os itens vão vazar do container\*/

**flex-basis: 100px**; /\*precisa ter minimamente 100px porém devido o shrink for 0, não vai permitir a compressão dos itens\*/

**}**

**.shrink2 {**

**flex-shrink: 2;** /\*pode ser comprimido 2 vezes\*/

**flex-basis: 100px;**

**}**

**.shrink3 {**

**flex-shrink: 3;** /\*pode ser comprimido 3 vezes\*/

**flex-basis: 100px;**

**}**

**.grow1 {**

**flex-grow: 1;**

**}**

**</style>**

**</head>**

**<body>**

**<h1>shrink1</h1>**

**<div class="flex-container">**

**<div class="item shrink1 grow1">A</div>**

**<div class="item shrink1 grow1">B</div>**

**<div class="item shrink1 grow1">C</div>**

**<div class="item shrink1 grow1">D</div>**

**<div class="item shrink1 grow1">E</div>**

**</div>**

**<h1>shrink1</h1>**

**<div class="flex-container">**

**<div class="item shrink1 grow1">A</div>**

**<div class="item shrink1 grow1">Lorem ipsum dolor</div>**

**<div class="item shrink1 grow1">C</div>**

**<div class="item shrink1 grow1">D</div>**

**</div>**

**<h1>shrink0 + basis 100px</h1>**

**<div class="flex-container">**

**<div class="item shrink0 grow1">A</div>**

**<div class="item shrink0 grow1">Lorem ipsum dolor</div>**

**<div class="item shrink0 grow1">Cipsum dolor</div>**

**<div class="item shrink0 grow1">D</div>**

**</div>**

**<h1>shrink mix + basis 100px</h1>**

**<div class="flex-container">**

**<div class="item shrink1 grow1">A</div>**

**<div class="item shrink2 grow1">Lorem ipsum dolor</div>**

**<div class="item shrink2 grow1">C</div>**

**<div class="item shrink3 grow1">D</div>**

**</div>**

**</body>**

**Flex:** esta propriedade é um atalho ou abreviação de escrita para as propriedades: grow, shrink e basis, exemplo flex: 1 0 auto; o grow é 1 o shrink é 0 e o basis é auto.

**<style>**

**.flex-container {**

**display: flex;**

**border: 1px solid black;**

**max-width: 400px;**

**margin-bottom: 10px;**

**}**

**.item {**

**background-color: brown;**

**color: white;**

**font-size: 24px;**

**text-align: center;**

**margin: 5px;**

**}**

**.flex1 {**

**flex: 1;** /\*aqui está dizendo que o grow=1 (o espaçamento que sobra do conteúdo está sendo alinhado com a proporcionalidade de 1) o shrink=1 (é a compressão de 1 vez do item relacionado ao conteúdo) e o basis=0 (vai fazer com que não haja quebra no item, então se continuaar crescendo o conteúdo, irá crescer o item sem a quebra de linha)\*/

**}**

**.largura {**

**width: 200px;**

**}**

**.flex {**

**flex: 2;** /\*aqui o grow=2 o shrink=1 e o basis=0\*/

**}**

**</style>**

**</head>**

**<body>**

**<h1>flex 1</h1>**

**<div class="flex-container">**

**<div class="item flex1">A</div>**

**<div class="item flex1">BBB</div>**

**<div class="item flex1">CC</div>**

**<div class="item flex1">D</div>**

**<div class="item flex1">EF</div>**

**</div>**

**<h1>flex 2</h1>**

**<div class="flex-container">**

**<div class="item flex1">A</div>**

**<div class="item flex2">BBB</div>**

**<div class="item flex2">CC</div>**

**<div class="item flex1">D</div>**

**<div class="item flex1">EF</div>**

**</div>**

**<h1>width ignorado basis=0</h1>**

**<div class="flex-container">**

**<div class="item largura">A</div>**

**<div class="item largura">BBB</div>**

**<div class="item largura">CC</div>**

**<div class="item largura">D</div>**

**<div class="item largura">EF</div>**

**</div>**

**</body>**

**Order:** vai lhe dar diretamente com a ordenação dos itens, é mais uma propriedade relacionada ao flex items.

**<style>**

**.flex-container {**

**display: flex;**

**max-width: 350px;**

**border: 1px solid black;**

**margin-bottom: 20px;**

**}**

**.item {**

**flex: 1;** /\*aqui está dizendo que o grow=1 (o espaçamento que sobra do conteúdo está sendo alinhado com a proporcionalidade de 1) o shrink=1 (é a compressão de 1 vez do item relacionado ao conteúdo) e o basis=0 (vai fazer com que não haja quebra no item, então se continuaar crescendo o conteúdo, irá crescer o item sem a quebra de linha)\*/

**background-color: brown;**

**color: white;**

**text-align: center;**

**font-size: 24px;**

**margin: 5px;**

**}**

**.order1 {**

**order: 1;** /\*eu mudo a ordem dos itens usando order 1, order 2, order 3 basta criar a classe e adicionar no item div\*/

**}**

**.order2 {**

**order: 2;**

**}**

**.order3 {**

**order: 3;**

**}**

**.order4 {**

**order: 4;**

**}**

**.direction {**

**flex-direction: column;** /\*muda o container para coluna os itens ficam um embaixo do outro\*/

**}**

**</style>**

**</head>**

**<body>**

**<h1>order 0 row</h1>**

**<div class="flex-container">**

**<div class="item">1</div>**

**<div class="item">2</div>**

**<div class="item">3</div>**

**<div class="item">4</div>**

**</div>**

**<h1>order diferentes</h1>**

**<div class="flex-container">**

**<div class="item">1</div>**

**<div class="item">2</div>**

**<div class="item">3</div>**

**<div class="item order2">4</div>**

**<div class="item order3">4n</div>**

**<div class="item order1">4m</div>**

**</div>**

**<h1>order diferentes em column</h1>**

**<div class="flex-container direction">**

**<div class="item">1</div>**

**<div class="item">2</div>**

**<div class="item">3</div>**

**<div class="item order2">4</div>**

**<div class="item order3">4n</div>**

**<div class="item order1">4m</div>**

**</div>**

**</body>**

**Align-self:** ultima propriedade relacionada a itens, é a propriedade que estabelece o alinhamento de modo individual sobre um determinado item, preciso que o container não tenha uma align definida. Os valores possíveis do align-self são **auto** que é o valor padrão, irá respeitar a definição de align-items do container, **flex-start** vai alinha ao início do container, **flex-end** vai alinhar ao final do container, **center** vai alinhar ao centro de acordo com o eixo, **stretch** ocupa todo o espaço relativo, **baseline** utiliza a linha base da tipografia.

**<style>**

**.flex-container {**

**display: flex;**

**border: 1px solid black;**

**margin-bottom: 10px;**

**max-width: 350px;**

**}**

**.align-container {**

**align-items: center;** /\*os itens irá respeitar o alinhamento setado no\*/

**}**

**.item {**

**background-color: brown;**

**color: white;**

**font-size: 24px;**

**text-align: center;**

**margin: 5px;** /\*se não colocar margin, os itens ficam grudados\*/

**}**

**.center {**

**align-self: center;**

**}**

**.start {**

**align-self: flex-start;**

**}**

**.end {**

**align-self: flex-end;**

**}**

**.baseline {**

**align-self: baseline;**

**}**

**.stretch {**

**align-self: stretch;**

**}**

**.auto {**

**align-self: auto;**

**}**

**.direction {**

**flex-direction: column;**

**}**

**</style>**

**</head>**

**<body>**

**<h1>align container </h1>**

**<section class="flex-container align-container">**

**<div class="item auto">AAA</div>**

**<div class="item auto">123</div>**

**<div class="item auto">Lorem ipsum dolor sit amet</div>**

**<div class="item auto">ITEM 1</div>**

**</section>**

**<h1>align diferente</h1>**

**<section class="flex-container">**

**<div class="item center">AAA</div>**

**<div class="item start">123</div>**

**<div class="item stretch">Lorem ipsum dolor sit amet Lorem ipsum dolor sit amet</div>**

**<div class="item end">ITEM 1</div>**

**</section>**

**<section class="flex-container">**

**<div class="item center">AAA</div>**

**<div class="item start">123</div>**

**<div class="item end">ITEM 1</div>**

**<div class="item stretch">Lorem ipsum dolor sit amet Lorem ipsum dolor sit amet</div>**

**</section>**

**<h1>column</h1>**

**<section class="flex-container align-container direction">**

**<div class="item auto">AAA</div>**

**<div class="item auto">123</div>**

**<div class="item auto">ITEM 1</div>**

**<div class="item auto">Lorem ipsum dolor sit amet Lorem ipsum dolor sit amet</div>**

**</section>**

**<section class="flex-container direction">**

**<div class="item center">AAA</div>**

**<div class="item start">123</div>**

**<div class="item stretch">Lorem ipsum dolor sit amet Lorem ipsum dolor sit amet</div>**

**<div class="item end">ITEM 1</div>**

**</section>**

**<section class="flex-container direction">**

**<div class="item center">AAA</div>**

**<div class="item start">123</div>**

**<div class="item end">ITEM 1</div>**

**<div class="item stretch">Lorem ipsum dolor sit amet Lorem ipsum dolor sit amet</div>**

**</section>**

**</body>**

**Projeto flexbox  
Codigo HTML**

**<!DOCTYPE html>**

**<html lang="pt-br">**

**<head>**

**<meta charset="UTF-8">**

**<meta name="viewport" content="width=device-width, initial-scale=1.0">**

**<title>Flex Turismos</title>**

**<link rel="stylesheet" href="./style.css">**

**</head>**

**<body>**

**<header>**

**<div class="flex-container menu">**

**<div>**

**<h1>FlexTurismos</h1>**

**</div>**

**<ul class="list-items">**

**<li><a href="#quem-somos">Quem Somos</a></li>**

**<li><a href="#servicos">Serviços</a></li>**

**<li><a href="#planos">Planos</a></li>**

**</div>**

**</ul>**

**</header>**

**<div class="flex-container apresentacao">**

**<div class="texto-apresentacao">**

**<h1>Flex <br>Turismos</h1>**

**<p>O melhor serviço para você!</p>**

**<a href="" class="btn">Saiba Mais!</a>**

**</div>**

**<div>**

**<div><img src="./images/0-main.png" alt="banner de apresentação"></div>**

**</div>**

**</div>**

**<div>**

**<div class="flex-container" id="quem-somos">**

**<div>**

**<h2>Quem somos</h2>**

**<p>It is a long established fact that a reader will be distracted by the readable content of a page when**

**looking at its layout. </p>**

**<p>The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as**

**opposed to using 'Content here, content here', making it look like readable English.</p>**

**</div>**

**<div>**

**<img src="./images/1-quem-somos.png" alt="balcão de atendimento">**

**</div>**

**</div>**

**<div class="container-externo">**

**<div class="flex-container" id="servicos">**

**<div>**

**<h2>Serviços</h2>**

**</div>**

**<div class="list-servicos">**

**<div class="item-servico">**

**<div><img src="./images/icon-2.png" alt="hospedagens"></div>**

**<p>Hospedagens</p>**

**<a href="#">Comprar Agora</a>**

**</div>**

**<div class="item-servico">**

**<div><img src="./images/icon-1.png" alt="pacote de viagens"></div>**

**<p>Pacotes de viagens</p>**

**<a href="#">Comprar Agora</a>**

**</div>**

**<div class="item-servico">**

**<div><img src="./images/icon-3.png" alt="roteiros personalizados"></div>**

**<p>Roteiros personalizados</p>**

**<a href="#">Comprar Agora</a>**

**</div>**

**</div>**

**</div>**

**</div>**

**<div class="flex-container" id="planos">**

**<div><h2>Planos</h2></div>**

**<div class="list-planos">**

**<div class="item-plano">**

**<h3>Plano 2</h3>**

**<ul>**

**<li>Suporte 24h</li>**

**<li>Serviços de quarto</li>**

**<li>Guia turístico</li>**

**</ul>**

**<a href="#" class="btn">Saiba Mais!</a>**

**</div>**

**<div class="item-plano">**

**<h3>Plano 2</h3>**

**<br>**

**<ul>**

**<li>Suporte 24h</li>**

**<li>Serviços de quarto</li>**

**<li>Guia turístico</li>**

**<li>Roteiro de trilhas</li>**

**<li>Serviço personalizado</li>**

**</ul>**

**<a href="#" class="btn">Saiba Mais!</a>**

**</div>**

**<div class="item-plano">**

**<h3>Plano 3</h3>**

**<br>**

**<ul>**

**<li>Suporte 24h</li>**

**<li>Serviços de quarto</li>**

**<li>Guia turístico</li>**

**<li>Roteiro de trilhas</li>**

**<li>Serviço personalizado</li>**

**<li>Área Vip</li>**

**</ul>**

**<a href="#" class="btn">Saiba Mais!</a>**

**</div>**

**</div>**

**</div>**

**<footer>**

**<div class="flex-container footer">**

**<p>&copy; 2021 CSS Flexbox</p>**

**<p>Desenvolvido por: Igor Ferreira</p>**

**</div>**

**</footer>**

**</body>**

**</html>**

**Código CSS**

**@charset "UTF-8";**

**\*{**

**margin: 0;**

**padding: 0;**

**font-family: 'open sans', sans-serif;**

**}**

**ul {**

**list-style: none;** /\*tirar o estilo das listas, tirar os pontinhos ou números da frente da lista\*/

**margin: 0;**

**padding: 0;**

**}**

**ul li a{**

**text-decoration: none;** /\*tirar underline da ancora da lista dentro de ul\*/

**}**

**div img {** /\*o que estiver em div e for imagem, aqui vai trabalhar a responsividade das imagens\*/

**display: block;**

**width: 100%;**

**}**

**.flex-container {**

**display: flex;** /\*cria o container e faz com que seja possível manipular os itens filhos do container\*/

**max-width: 992px;** /\*maxima largura do container\*/

**margin: auto;** /\*margem automática\*/

**width: 100%;**

**min-width: 320px;**

**}**

**header {** /\*header é uma tag, não é uma classe, por isso não utilizar ponto, header é cabeçalho\*/

**background-color: #122A57;**

**height: 100px;**

**display: flex;**

**align-items: center;** /\*alinhamento dos items ao centro\*/

**color: white;**

**}**

**header .list-items {** /\*class list-items que está dentro da tag header\*/

**display: flex;** /\*cria o container na classe list-items dentro de header e faz com que seja possível manipular os itens filhos do container\*/

**max-width: 260px;**

**width: 100%;** /\*largura do item, vai fazer com que fique espaçado\*/

**justify-content: space-between;** /\*vai alinhar os itens do início e do final próximo a linha inicial e final do container\*/

**align-items: center;** /\*irá alinha os itens ao centro na vertical\*/

**}**

**.list-items li a{** /\*aconra dentro de li que esta dentro da class list-items\*/

**color: white;** /\*vai dar cor a ancora da lista dentro da classe list-items\*/

**}**

**header .menu {** /\*class menu dentro da tag header\*/

**justify-content: space-between;** /\*vai alinhar o menu para a direita\*/

**}**

**.apresentacao {** /\*class apresentacao\*/

**height: 100vh;** /\*vh é viewport\*/

**align-items: center;** /\*alinhamento dos items ao centro\*/

**justify-content: space-between;** /\*vai alinhar os itens do início e do final próximo a linha inicial e final do container\*/

**}**

**.apresentacao .texto-apresentacao {** /\*classe texto-apresentacao dentro da classe apresentacao\*/

**min-width: 200px;**

**}**

**.texto-apresentacao h1 {** /\*h1 dentro da class texto-apresentação\*/

**color: #122A57;**

**font-size: 48px;**

**margin-bottom: 10px;**

**}**

**.btn {** /\*texto ancora é o saiba mais\*/

**background-color: #122A57;**

**color: white;**

**text-align: center;**

**border-radius: 30px;** /\*a borda fica arredondada\*/

**width: 220px;**

**display: block;**

**text-decoration: none;**

**height: 50px;**

**line-height: 50px;** /\*centraliza o texto verticalmente\*/

**margin-top: 10px;**

**margin-bottom: 10px;**

**}**

**#quem-somos {** /\*id quem somos\*/

**flex-direction: row-reverse;** /\*inverte a posição ou troca de lado os itens, o item que ta na direita vai para esquerda e o item que está na esquerdaa vai para direita\*/

**align-items: center;** /\*alinhamento dos items ao centro\*/

**justify-content: space-between;** /\*vai alinhar os itens do início e do final próximo a linha inicial e final do container\*/

**}**

**#quem-somos h2 {** /\*h2 dentro do id quem-somos\*/

**font-size: 32px;**

**color: #122A57;**

**display: flex;** /\*cria o container e faz com que seja possível manipular os itens filhos do container, nesse caso o quem somos se transformará em container\*/

**margin-bottom: 20px;**

**}**

**#quem-somos h2::before { /**\*before é um pseudo-elemento, irá criar uma barra (|) de destaque no quem somos\*/

**content: "";** /\*para acrescentar o risco content tem que estar vazio\*/

**height: 50px;** /\*dimensionamento da barra\*/

**width: 5px;** /\*dimensionamento da barra\*/

**margin-right: 5px;**

**background-color: #122A57;**

**position: relative;**

**}**

**#quem-somos p {** /\*texto dentro do id quem-somos\*/

**margin-bottom: 10px;**

**width: 90%;**

**}**

**#quem-somos img {** /\*img dentro do id quem-somos\*/

**min-width: 420px;** /\*irá aumentar a largura da imagem\*/

**}**

**.container-externo {**

**background-color: #122A57;**

**width: 100%;**

**}**

**#servicos { /\*id serviços\*/**

**flex-direction: column;** /\*os itens vão ficar em coluna\*/

**padding-top: 50px;** /\*preenchimento no topo\*/

**padding-bottom: 100px;** /\*preenchimento embaixo\*/

**}**

**#servicos h2 {** /\*h2 drntro de id serviços\*/

**color: white;**

**font-size: 32px;**

**margin-bottom: 20px;**

**}**

**.list-servicos {**

**display: flex;** /\*irá transformar o list-servicos em container e os itens que estão dentro poderão ser editador\*/

**justify-content: space-between;** /\*vai alinhar os itens do início e do final próximo a linha inicial e final do container\*/

**}**

**.list-servicos .item-servico {** /\*item serviço dentro de list-serviço\*/

**text-align: center;**

**}**

**.item-servico a {** /\*ancora dentro de item-serviço\*/

**width: 220px;**

**background-color: white;**

**border-radius: 30px;** /\*faz um arredondamento\*/

**height: 50px;**

**text-align: center;**

**margin-top: 20px;**

**line-height: 50px;** /\*altera a centralização vertical\*/

**padding: 5px 10px;** /\*preenchimento\*/

**text-decoration: none;** /\*tira o underline\*/

**color: #122A57;**

**font-size: 12px;** /\*tamanho da fonte\*/

**font-weight: 700;**

**}**

**.item-servico p {** /\*texto dentro de item-serviço\*/

**font-weight: 700;**

**font-size: 18px;**

**color: white;**

**margin-top: 20px;**

**}**

**.item-servico img {** /\*img dentro de item-servico\*/

**width: 80%;**

**margin: auto;**

**}**

**#planos {**

**flex-direction: column;** /\*os itens ficarão em coluna\*/

**min-height: 100vh;**

**padding-top: 50px;**

**}**

**#planos h2{**

**font-size: 32px;**

**}**

**.list-planos {**

**display: flex;** /\*cria o container na classe list-items dentro de header e faz com que seja possível manipular os itens filhos do container\*/

**align-items: flex-end; /\*os itens ficaarão alinhados na borda inferior do container\*/**

**justify-content: space-between;** /\*vai alinhar os itens do início e do final próximo a linha inicial e final do container\*/

}

**.item-plano {**

**flex: 1; ;** /\*aqui está dizendo que o grow=1 (o espaçamento que sobra do conteúdo está sendo alinhado com a proporcionalidade de 1) o shrink=1 (é a compressão de 1 vez do item relacionado ao conteúdo) e o basis=0 (vai fazer com que não haja quebra no item, então se continuar crescendo o conteúdo, irá crescer o item sem a quebra de linha)\*/

**border: 5px solid #122A57;**

**margin-right: 20px;**

**padding: 10px;** /\*preenchimento no item plano, fez com que afastasse o risco embaixo do título plano\*/

**max-width: 240px;**

**}**

**.item-plano .btn {** /\*class btn que está dentro da class item-plano\*/

**margin: auto;**

**margin-bottom: 20px;**

**}**

**.item-plano h3 {** /\*h3 dentro de item-plano\*/

**font-size: 24px;**

**display: flex;**

**flex-direction: column;** /\*os itens ficarão em coluna\*/

**text-align: center;**

**margin-top: 20px;**

**}**

**.item-plano h3::after {** /\*after é um pseudo-codigo, fará um risco abaixo do titulo plano que está dentro da class item-plano\*/

**content: "";** /\*para acrescentar o risco content tem que estar vazio\*/

**background-color: #122A57;**

**width: 100%;**

**height: 3px;**

**margin-top: 20px;**

**margin-bottom: 10px;**

**}**

**.item-plano ul {** /\*ul que está dentro da class item-plano\*/

**padding: 10px 20px;** /\*preenchimento em cima e embaixo de 10px e preenchimento na direita e esquerda 20px\*/

**flex-direction: column;** /\*os itens ficarão em coluna\*/

**margin-bottom: 10px;**

**}**

**.item-plano ul li {**

**display: flex;** /\*vai transformar o li do item-plano em container\*/

**flex-wrap: nowrap;** /\*sem quebra de linha\*/

**align-items: center;**

**}**

**.item-plano ul li::before {** /\*before é um pseudo-elemento, irá criar nesse caso um ponto de marcação na frente do texto\*/

**content: "";** /\*para acrescentar o ponto content tem que estar vazio\*/

**width: 10px;** /\*dimensionaamento do ponto\*/

**height: 10px;** /\*dimensionaamento do ponto\*/

**background-color: #122A57;**

**margin-right: 5px;**

**}**

**footer {**

**background-color: #122A57;**

**height: 70px;**

**display: flex;** /\*transforma o footer em container\*/

**align-items: center;**

**}**

**footer .footer {** /\*class footer que está dentro da tag footer\*/

**justify-content: space-between;** /\*vai alinhar os itens do início e do final próximo a linha inicial e final do container\*/

**color: white;**

**}**

**/\* mobile - responsividade do site\*/**

**@media(max-width: 992px) {** /\*se passaar de 992px o container plano fica em coluna\*/

**.flex-container {**

**flex-direction: column;**

**}**

**.apresentacao {**

**flex-direction: column-reverse; /\*colune reversa, inverte os itens\*/**

**}**

**.apresentacao .texto-apresentacao {** /\*classe texto-apresentacao dentro da classe apresentacao\*/

**width: 100%;** /\*código que trata a responsividade do texto-apresentação\*/

**}**

**#quem-somos {**

**flex-direction: column-reverse;** /\*colune reversa, inverte os itens\*/

**}**

**#quem-somos img {**

**min-width: 320px;**

**margin: auto;**

**}**

**.list-servicos {**

**flex-direction: column;**

**}**

**.item-servico img {**

**width: 50%;**

**margin: auto;**

**}**

**.list-planos {**

**flex-direction: column;**

**align-items: flex-start;** /\*posicionará o item no começo do container\*/

**margin-bottom: 20px;**

**}**

**.list-planos .item-plano {**

**max-width: 90%;**

**margin: auto;**

**width: 100%;**

**margin-bottom: 20px;**

**}**

**}**