POSICIONANDO ELEMENTOS COM FLEXBOX

1. INTRODUÇÃO

* -Objetivo – Criar páginas que vão se adaptar a todas as resoluções sem precisar ficar aplicando uma propriedade para cada, ou seja, sites responsáveis.
* Flexbox – Foi projetado como modelo de layout unidimensional, que oferece uma distribuição de espaço e alinhamento entre os itens.
* Flex Container – Tag que envolve os itens, será nela que aplicaremos as propriedades “display:flex”. Transforma seus itens filhos em flex itens, onde será então aplicado as propriedades. São elas: display (inicializa o container), flex-direction ( fazer o direcionamentos desses itens seja em linha ou coluna), flex-wrap ( para quebra de linha ou não), flex-flow ( abreviação pro Direction ou wrap), justify-content ( alinhar os itens do container de acordo com a sua direção ), align-items ( alinhar os items de acordo com o eixo do container), align-content ( vai alinhar as linhas desse container)
* Flex item – São os elementos filhos diretos do flex container. E também podem ser aplicado a propriedade display neles para se transformarem em flex-container. Também possuem propriedades, são elas : flex-grow ( defini o fator de crescimento), flex-basis ( definir o tamanho inicial desse item antes da distribuição do espaço restante dentro do container), shrink ( defini a capacidade de redução ) flex, ( abreviação para as propriedades citadas, grow, basis, shrink) order ( relacionado a ordem e a listagem desses item) align-self ( definir o alinhamento de um item especifico dentro do container )

1. FUNDAMENTOS DO FLEXBOX -PARTE 1

* Estrutura básica do display:flex – inicializar uma tag ( qualquer ) com um display:flex, vai dar inicio a tornar todos os filhos diretos dessa tag em flex item.
* Flex-direction – é a propriedade que estabelece a direção do eixo do container, definindo assim a direção que os itens são colocados no container.
* Row (padrão) – É do sentido esquerda para direita, como escrevemos.
* Row-reverse – sentido oposto da direção do texto.
* Column – ordenação de cima para baixo na coluna.
* Column-reverse – ordenação oposta de baixo para cima
* Flex-wrap – define ou não a quebra de linha do container, por padrão não quebra linha os itens dentro do container, o que acontece é que os itens podem se deslocar para fora do container a depender. Nowrap (é o comportamento padrão onde não há quebra de linha), wrap (permite a quebra de linha a partir do primeiro momento em que o próximo item não caiba no container), o wrap-reverse ( é a mesma lógica do wrap, porém será de modo oposto, ao invés da quebra de linha ser na linha de baixo, será acima, inverte o fluxo da quebra de linha)
* Flex-flow – um atalho para o flex Direction ou wrap.

* Justify content – Vai se encarregar dos alinhamentos dos itens do container, e tratar da distribuição de espaço entre eles, se há algum item que ocupa 100% do container, não tem sentido usar essa propriedade. Flex-start ( faz o alinhamento no início do container), flex-end (vai levar todos os itens de acordo com a sobra do espaço para o final do container), center (traz os elementos ao centro do container), space-between ( cria uma espaço igual entre os elementos, mas pega o primeiro elemento e coloca ao inicio a borda esquerda do elemento e o ultimo próximo a borda direita), space-around ( vai tratar do espaçamento do meio tornando ele duas vezes maior que o inicial)
* Align items- vai tratar do alinhamento dos itens de fato, de acordo com o eixo do container, a diferença dele para o justify content é porque não vai precisar saber da altura desses itens filho do container. Tipo: center ( trazendo os itens ao centro), stretch ( faz com que crescam de acordo com o maior item dentro do container, deixando-os iguais), flex-start ( trazer os elementos para o começo do contente) flex-end ( levar para o final do do nosso content), baseline ( ele vai utilizar utilizar o eixo da linha relacionado ao texto do conteúdo desses items)
* Align conten – é responsavel para tratar do eixo vertical, trabalhar a orientação da linha do container, a altura do container tem que ser maior do que a soma da linha dos itens. Tipo: center, stretch, flex-start, flex-end, space-between, space-around.

1. FUNDAMENTOS DE FLEXBOX – PARTE 2

* Estrutura básica flex grow - Essa propriedade defini a proporcionalidade de crescimentos desses items, sempre respeitando o tamanho do conteúdo interno dos itens. Obs: não funciona se tiver usado justify-content no container.
* Estrutura flex basis – Estabelece o tamanho inical dos itens antes da distribuição do espaço restante, pega o tamanho mínimo que o item vai precisar ter, e vai fazer a distribuição da sobra do espaço antes e após o conteúdo. Usa-se: auto ( caso o item não tenha um tamanho definido, ele vai ser proporcional ao conteúdo), px, %, em ( valores exatos, que será o tamanho mínimo que o item terá, se o conteúdo do item for maior do que for definido vai acabar vazando, 0 ( vai ter relação direta com o flex-grow )
* Flex-shrink – Estabelecer a capacidade de redução ou compressão do item.
* Flex – Atalho para usar o grow shrink e basis (nessa ordem)
* Order – Lida diretamente com a ordem dos itens.
* Align self – estabelece o alinhamento individual de acordo com o nosso item, ao invés de utilizar o align-items pode-se utilizar o align self para definir o item individualmente. São eles : auto (valor padrão respeita o align-item), start(no início do container), end (no final do container), center ( ao centro de acordo com eixo do container), stretch ( o elemento vai ocupar totalmente o espaço relativo, baseline ( usa a linha base da topografia ).