Caixas em CSS

Box-sizing: responsável por calcular o tamanho total da caixa. Ele possui dois valores.

a-) content-box: Faça os cálculos a partir do conteúdo. O *width* será sempre configurado de acordo com o **conteúdo** de um elemento. Em CSS isso já vem como padrão.

b-) border-box: Faça os cálculos a partir da borda. O *width* será sempre configurado de acordo com a **margem** do elemento. Sempre deve-se aplicar isso nas configurações iniciais da página, pois o width sempre será configurado corretamente.

Display: block vs inline

a-) block: ocupa toda a linha, colocando o próximo elemento abaixo desse. As propriedades widht e height são respeitadas. Padding, margin, border irão funcionar normalmente.

b-) inline: elementos são posicionados um ao lado do outro. As propriedades widht e height **não funcionam**. Somente valores *horizontais* de margin, padding e border funcionam.

Layouts em CSS

Position: controla onde, na página, o elemento irá ficar; alterando o *fluxo normal* dos elementos. O fluxo normal de uma página é posicionar os elementos um embaixo do outro.

Valores aceitos:]

a-) static: os elementos por padrão são *static*, ou seja, seguem o fluxo normal da página.

b-) relative: libera 5 propriedades para posicionarmos nossos elementos:

- top: move o elemento para baixo.
- right: move o elemento para esquerda.
- bottom: move o elemento para cima.
- left: move o elemento para direita.
- z-index: configura a prioridade para ficar na frente dos elementos.

A propriedade *position: relative* **não altera** o fluxo normal da página. Os elementos não são afetados conforme ele se mexe, portanto, não irão ocupar o espaço em branco deixado por ele.

c-) absolute: assim como o *relative* ele libera as 5 propriedades. Todavia, seu comportamento é diferente. Ele adiciona uma "camada imaginária" acima do fluxo normal da página.

As 5 propriedades, quando alteradas, são alteradas em relação à página inteira.

Caso um elemento com *position: absolute* estiver dentro de um elemento pai com *position: relative*, ele se move em relação ao elemento pai.

Portanto, configurar um left: 0 e top: 0, iria posicionar o elemento colocado na esquerda-acima.

d-) fixed: se mantém *fixo* durante toda a rolagem da página. Também libera as 5 propriedades, que são configuradas em relação à página.

Element Stacking: traduzido do inglês, significa "empilhamento de elementos". Trata-se da propriedade z-index. Quanto maior for o valor de z-index (aceita qualquer valor inteiro), mais preferência de ficar acima de outros elementos ele terá.

Flexbox: nos permite posicionar os elementos dentro da caixa. Controle em uma dimensão (horizontal **ou** vertical). Alinhamento, direcionamento, ordenar e tamanhos.

Todos os filhos de uma caixa configurada como *display: flex* possuem as seguintes propriedades:

a-) flex-direction: qual a direção do flex: horizontal ou vertical

- row: os elementos são posicionados um ao lado do outro.
- column: os elementos são posicionados um embaixo do outro.

b-) alinhamento:

- **justify-content:** Alinhamento horizontal dos itens.
    - **space-between:** separa igualmente os elementos ao longo da caixa. Conforme a página for mudando de tamanho, os elementos **mantêm** o espaçamento.
    - **center:** os elementos são mantidos unidos e sempre alinhados no centro.

- **align-items:** Utilizo sempre na caixa pai, para que os itens dentro dela fiquem alinhados.
    - **center:** alinha verticalmente os elementos ao centro da página.

Grid: Configura o posicionamento dos elementos dentro da caixa. Configura os posicionamentos horizontal e vertical ao mesmo tempo. Pode ser flexível ou fixo. Cria espaços para os elementos filhos habitarem.

Ele vai criar uma grade dentro do meu site. Preciso adicionar o grid ao pai dos elementos que desejo posicionar.

comando: display: grid;

grid-template-areas: utilizado para definir as minhas áreas de grade

grid-template-areas:
"menu menu" //(Ocupa duas colunas) cada nome que coloco dentro das aspas, separados por espaço, representa uma coluna na minha grade.

"main aside" // Cada conjunto de " " representa uma linha na minha grade. Nesse caso, main e aside irão dividir a linha em 2.

"footer footer" //Irá ocupar duas colunas

Dentro do elemento que quer colocar a característica configurada nos nomes, eu coloco:

grid-area: nomeUtilizado;

grid-template-rows: define o tamanho de cada linha

grid-template-rows: tamL1 tamL2 ... tamLn;