**Layouts HTML**

**Grid Layout**

Nesse tipo de layout trabalha-se com os conceitos de tabelas. A seção que engloba todas as demais será dividida em linhas e colunas e então as demais seções serão alocadas nestes espaços. Pode-se utilizar de duas abordagens diferentes:

1. **Alocar demais seções manualmente:**

Nessa abordagem a seção que engloba as demais é dividida por meio dos comandos grid-template-columns e grid-template-rows, como pode ser observado no exemplo abaixo, que divide a página em 3 colunas e 3 linhas:

section{

display: grid;

grid-template-columns: 1fr 3fr 1fr;

grid-template-rows: 20vh 70vh 10vh;

}

No exemplo acima teríamos 3 colunas dividindo a página cinco (5) frações (fr), mas a coluna do meio seria maior, ocupando 3 frações do espaço total enquanto as demais colunas ocupariam apenas uma (1) fração da página. As linhas seriam divididas da seguinte forma: a primeira linha ocuparia 20% da altura da página, a segunda 70% da página e a terceira 10%. O vh no código faz referência ao termo view height.

Após essa divisão de espaços da seção inicial, basta configurar no código de cada objeto quais espaços eles ocuparão. Observe no exemplo abaixo:

header{

background-color: #7C7C7C;

grid-column: 1/4;

}

nav{

background-color: #F5F3F3;

grid-column: 1/2;

}

main{

background-color: #FF0;

grid-column: 2/3;

}

aside{

background-color: #C4C4C4;

grid-column: 3/4;

}

footer{

background-color: #212121;

grid-column: 1/4;

}

Nessa configuração 3 x 3 cada coluna possui 4 posições (1, 2 , 3 , 4) sendo a posição um no canto esquerdo da página e a posição 4 no canto oposto. No exemplo acima verifica-se que o header ocupa a primeira linha e as 3 colunas (espaços 1- 4). As seções nav, main e aside ocupam a segunda linha cada uma em uma coluna. A seção footer ocupa a terceira linha também nos espaços de 1 a 4.

1. **Alocar as seções designando a posição na seção mais externa**

Outra possibilidade é alocar a posição de cada seção por meio do código na seção mais externa, a que engloba as demais seções.Veja o exemplo abaixo:

section{

display: grid;

grid-template-columns: 1fr 3fr 1fr;

grid-template-rows: 20vh 70vh 10vh;

grid-template-areas: "h h h"

"n m a"

"f f f"

}

Nesse exemplo a section é a seção mais externa, ou seja, ela envolve todas as demais. os comandos grid-template-columns e grid-template-rows dividem a section em 3 coluna e 3 linhas. O comando grid-template-areas estabelece que a na primeira linha os 3 espaços referente as colunas serão ocupados por um elemento h, a segunda linha terá os elementos nomeados por n, m e a em cada uma de suas colunas. A terceira linha será preenchida pelo elemento f em seus 3 espaços.

Agora basta ir em cada elemento e definir qual será: h, n, m, a , f. Na continuação do exemplo header ficou definido como h, nav como n, main como m, aside como a, e footer como f. Observe o código abaixo:

header{

background-color: #7C7C7C;

grid-area: h;

}

nav{

background-color: #F5F3F3;

grid-area: n;

}

main{

background-color: #FF0;

grid-area: m;

}

aside{

background-color: #C4C4C4;

grid-area: a;

}

footer{

background-color: #212121;

grid-area: f;

}