

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" href="../Examples/normalize.css">
<link rel="stylesheet" href="../Examples/reset.css">
<title>GRIDS ROW-START, ROW-END, COLUMN-START E COLUMN-END</title>
<style>
```

```
    /*      GRIDS ROW-START, ROW-END, COLUMN-START E COLUMN-END:
```

As propriedades "grid-row-start", "grid-row-end", "grid-column-start" e "grid-column-end" são propriedades utilizadas para delimitar se um determinado grid item deverá ir até uma determinada linha ou coluna de grid.

O que cada uma delas faz?

grid-row-start: determina em que linha um grid deve começar

grid-row-end: determina em que linha um grid deve terminar (lembrando que esse valor é exclusivo, ou seja se queremos que termine na linha 3, temos que usar o valor 4)

grid-column-start: determina em que coluna um grid deve começar

grid-column-end: determina em que coluna um grid item deve terminar (lembrando que esse valor é exclusivo, ou seja, se queremos que termine na coluna 3, temos que usar o valor 4)

A partir do momento que nós delimitamos onde um grid item deverá ir e onde deverá terminar o restando do conteúdo do grid container irá preencher os espaços restantes.

```
    */
```

```
#template {
  color: white;
  display: grid;
  grid-template-areas:
    "cabecalho"
```

```
"conteudo"
"rodape";
grid-template-columns: 60vw;
grid-template-rows: 5vh 50vh 5vh;
row-gap: 1%;
}

.cabecalho {
  grid-area: cabecalho;
  background-color: darkorchid;
}

.conteudo {
  grid-area: conteudo;
  background-color: darkorchid;
  display: grid;
  grid-template-areas: "grid1 grid2 grid3"; /*Delimitamos aqui eu os grid-items vão agir em 3 colunas...*/
  grid-template-columns: auto; /*E que as colunas vão crescer de forma automática de acordo com o espaço*/
  grid-template-rows: 10% 10% 10%; /*E aqui delimitamos o tamanho de até 3 linhas com sendo 10% da altura do*/
  gap: 1%; /*container...*/
}

.rodape {
  grid-area: rodape;
  background-color: darkorchid;
}

.grid1 {
  grid-area: grid1;
  background-color: red; /*E aqui colocamos os columns e rows start e end em ação...*/
  grid-column-start: 1; /*Note que o grid 1 começa na coluna 1*/
  grid-column-end: 3; /*E vai até a coluna 2, lembre-se que o valor do end é sempre exclusivo*/
  grid-row-start: 1; /*Além disso ele vai começar na linha 1*/
  grid-row-end: 2; /*E vai terminar na linha 1 também, */
}
```

```

.grid2 {
  grid-area: grid2;
  background-color: seagreen; /*Já o grid 2, em vez de posicionar ele lado a lado com o grid 1, vamos colocar*/
  grid-column-start: 1;      /*eles embaixo. Ele também vai começar na coluna 1*/
  grid-column-end: 2;        /*finalizando também na coluna 1...*/
  grid-row-start: 2;         /*e vai iniciar na linha 2*/
  grid-row-end: 4;           /*finalizando na 3 linha do grid*/
}

.grid3 {
  grid-area: grid3;
  background-color: darkblue; /*E vamos deixar o grid 3 lado a lado com o grid 1...*/
  grid-column-start: 3; /*Para isso ele vai começar na coluna 3*/
  grid-column-end: 3; /*Podemos colocar que ele vai finalizar na 3 também, já que ele começa á partir dela...*/
  grid-row-start: 1; /*Ele vai começar na linha 1 também e...*/
  grid-row-end: 4; /*Finalizar na linha 3.*/
}

</style>
</head>
<body id="template">

  <header class="cabecalho">
    Aqui é o cabeçalho
  </header>
  <main class="conteudo">
    No que qualquer coisa que esteja no meio vai preencher o conteúdo restante <!--Note que por causa do layout do grid essas
    palavras vão preencher o conteúdo restante-->
    <div class="grid1">Aqui temos o grid 01</div>
    <div class="grid2">Aqui temos o grid 02</div>
    <div class="grid3">Aqui temos o grid 03</div>
  </main>
  <footer class="rodape">
    Aqui é o rodapé
  </footer>

```

```
</body>  
</html>
```

RESULTADO NO NAVEGADOR...

