



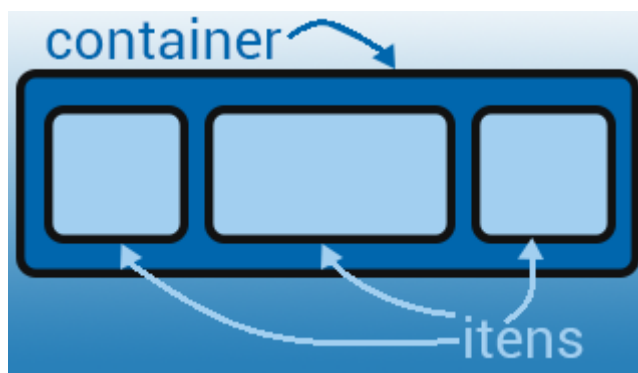
CSS Grid

O Que É CSS Grid?

O CSS Grid é um sistema de estruturação de layout que o CSS nos fornece. Diferente do Flexbox, que apenas nos permite trabalhar em uma **única dimensão**, o CSS Grid nos permite configurar layouts **em duas dimensões** (linhas e colunas). A junção de linhas e colunas formam uma grade, o que dá o nome a esse sistema (Grid).

O CSS Grid é bem poderoso e completo, permitindo que certas coisas sejam feitas de mais de uma maneira, o que pode confundir muitos iniciantes.

Primeiro de tudo temos que saber que teremos propriedades CSS para trabalhar com o elemento que possui nossos itens (grid container ou elemento pai) e propriedades para os nossos itens da grid (elementos filhos).



Grid Container

1 • display

Define o elemento como um grid container.

- `display: grid;`

```
// Torna o elemento um grid container.
```

- `display: inline-grid;`

```
// Torna o elemento um grid container porém com comportamento inline.
```

- `display: subgrid;`

```
// Para grids dentro de grids (ainda não é suportado, porém você pode normalmente colocar display: grid; no grid dentro do grid que funciona).
```

HTML

```
<div>
<h1>display: block;</h1>
<section class="grid">
  <div class="item">1</div>
  <div class="item">2</div>
  <div class="item">3</div>
  <div class="item">4</div>
</section>
</div>

<div>
<h1>display: grid;</h1>
<section class="grid">
  <div class="item">1</div>
  <div class="item">2</div>
  <div class="item">3</div>
  <div class="item">4</div>
</section>
</div>

<div>
<h1>grid-template-columns: 200px 200px;</h1>
<section class="container grid grid-template-columns">
  <div class="item">1</div>
  <div class="item">2</div>
  <div class="item">3</div>
  <div class="item">4</div>
</section>
```

CSS

```
/* Grid Container */
.grid {
  display: grid;
}

.grid-template-columns {
  grid-template-columns: 200px 200px;
}

.grid-template-columns-3 {
  grid-template-columns: 1fr 2fr 1fr;
}

.subgrid {
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
  grid-gap: 10px;
}

.subgrid > div {
  background: rgba(0,0,0,.3);
}

/* Estilos Gerais */
.item {
  margin: 5px;
  background: tomato;
  text-align: center;
```

<pre> </div> <div> <h1>grid-template-columns: 1fr 2fr 1fr;</h1> <section class="container grid grid-template-columns-3"> <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> </section> </div> <div> <h1>subgrid</h1> <section class="container grid grid-template-columns-3"> <div class="item">1</div> <div class="item">2</div> <div class="item">3</div> <div class="item">4</div> <div class="item subgrid"> <div>5.1</div> <div>5.2</div> <div>5.3</div> </div> <div class="item">6</div> </section> </div> </pre>	<pre> font-size: 1.5em; } .grid { max-width: 400px; margin: 0 auto; border: 1px solid #ccc; } h1 { text-align: center; font-size: 1.25em; font-weight: normal; } body { font-family: monospace; color: #333; margin: 0px; padding-bottom: 60px; } body > div { padding: 40px 0; } body > div:nth-of-type(even) { background: rgba(0,0,0,.03); } </pre>
--	---

2 • grid-template-columns

Define o número total de colunas que serão criadas no grid.

- `grid-template-columns: 100px 100px 100px 100px;`

`// Quatro colunas de 100px de largura são criadas`

- `grid-template-columns: 1fr 2fr;`

`// Duas colunas são criadas, sendo a segunda com o dobro do tamanho da primeira.`

`// fr é uma unidade fracional, o tamanho do conteúdo é respeitado, ou seja, se o conteúdo na primeira coluna for maior que o da segunda, a primeira será maior ou vice versa.`

- `grid-template-columns: minmax(200px, 1fr) 1fr 1fr;`

// Três colunas são criadas, a primeira terá no mínimo 200px de largura e no máximo 1fr (isso significa que após 200px ela se expande da mesma forma que as outras colunas). As outras duas colunas vão ter 1fr.

- `grid-template-columns: repeat(3, 1fr);`

// Cria 3 colunas com 1fr de tamanho. O repeat seria a mesma coisa que escrever 1fr 1fr 1fr.

- `grid-template-columns: repeat(auto-fit, minmax(100px, auto));`

// Cria automaticamente um total de colunas que acomode itens com no mínimo 100px de largura.

HTML

CSS

<pre><div> <h1>grid-template-columns: 100px 100px 100px 100px;</h1> <section class="grid grid-template-columns-1"> <div class="item">1</div> <div class="item">2</div> <div class="item">3</div> <div class="item">4</div> </section> </div> <div> <h1>grid-template-columns: 100px 100px 100px 100px;</h1> <section class="grid grid-template-columns-1"> <div class="item">PalavraTeste</div> <div class="item">2</div> <div class="item">3</div> <div class="item">PalavraTeste</div> </section> </div> <div> <h1>grid-template-columns: 1fr 2fr;</h1> <section class="grid grid-template-columns-2"> <div class="item">1</div> <div class="item">2</div> <div class="item">3</div> <div class="item">4</div> </section> </div></pre>	<pre>/* Grid */ .grid { display: grid; } /* 100px é o valor total, ignora conteúdo, margem e etc. Respeita apenas o min-width do item.*/ .grid-template-columns-1 { grid-template-columns: 100px 100px 100px 100px; } /* fr é uma unidade fracional */ .grid-template-columns-2 { grid-template-columns: 1fr 2fr; } /* minmax define um valor mínimo e máximo para a coluna */ .grid-template-columns-3 { grid-template-columns: minmax(200px, 1fr) 1fr 1fr; } /* O repeat vai repetir o número de colunas */ .grid-template-columns-4 { grid-template-columns: repeat(3, 1fr); }</pre>
--	---

<pre> <div> <h1>grid-template-columns: 1fr 2fr;</h1> <section class="grid grid-template-columns-2"> <div class="item">PalavraTesteGrandeRespeita</div> <div class="item">2</div> <div class="item">3</div> <div class="item">4</div> </section> </div> <div> <h1>grid-template-columns: minmax(200px, 1fr) 1fr 1fr;</h1> <section class="grid grid-template-columns-3"> <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> </section> </div> <div> <h1>grid-template-columns: minmax(200px, 1fr) 1fr 1fr;</h1> <section class="grid grid-template-columns-3"> <div class="item">TestePalavraTeste</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> </section> </div> <div> <h1>grid-template-columns: minmax(200px, 1fr) 1fr 1fr;</h1> <section class="grid grid-template-columns-3"> <div class="item">1</div> <div class="item">TestePalavraTeste</div> <div class="item">3</div> <div class="item">4</div> <div class="item">5</div> <div class="item">6</div> </section> </div> <div> <h1>grid-template-columns: repeat(3, 1fr);</h1> <section class="grid grid-template-columns-4"> <div class="item">1</div> <div class="item">2</div> </pre>	<pre> /* Auto-fit vai calcular automaticamente o total de colunas */ .grid-template-columns-5 { grid-template-columns: repeat(auto-fit, 100px); } /* Com o auto-fit realmente é criado um layout responsivo */ .grid-template-columns-6 { grid-template-columns: repeat(auto-fit, minmax(100px, auto)); } /* Com o auto-fit realmente é criado um layout responsivo */ .grid-template-columns-7 { grid-template-columns: repeat(auto-fill, minmax(100px, auto)); } /* Estilos Gerais */ .item { margin: 5px; background: tomato; text-align: center; font-size: 1.5em; } .grid { max-width: 400px; margin: 0 auto; border: 1px solid #ccc; } h1 { text-align: center; font-size: 1.25em; font-weight: normal; } body { font-family: monospace; color: #333; margin: 0px; padding-bottom: 60px; } body > div { padding: 40px 0; } body > div:nth-of-type(even) { </pre>
---	---

<pre> <div class="item">3</div> <div class="item">4</div> <div class="item">5</div> <div class="item">6</div> </section> </div> <div> <h1>grid-template-columns: repeat(auto-fit, 100px);</h1> <section class="grid grid-template-columns-5"> <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> </section> </div> <div> <h1>grid-template-columns: repeat(auto-fit, minmax(100px, auto));</h1> <section class="grid grid-template-columns-6"> <div class="item">PalavraTesteMesmo1</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> </section> </div> <div> <h1>grid-template-columns: repeat(auto-fill, minmax(100px, auto));</h1> <section class="grid grid-template-columns-7"> <div class="item">PalavraTesteMesmo1</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> </section> </div> </pre>	<pre> background: rgba(0,0,0,.03); } </pre>
--	---

3 • grid-template-rows

Define a quantidade de linhas no grid.

- `grid-template-rows: 50px 100px 50px 150px;`

```
// Cria 4 linhas no grid, sendo a primeira com 50px,
segunda 100px, terceira 50px e quarta 150px.
```

```
// Caso o grid necessite de mais linhas, elas terão o
tamanho de acordo com o conteúdo.
```

- `grid-template-rows: 1fr 2fr;`

```
// Cria 2 linhas no grid, sendo a segunda com cerca de
duas vezes o tamanho da primeira.
```

HTML

CSS

```
<div>
<h1>grid-template-rows: 50px 100px 50px
150px;</h1>
<section class="grid grid-template-rows-1">
  <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 class="item">7</div>
  <div class="item">8</div>
</section>
</div>

<div>
<h1>grid-template-rows: 100px 100px;</h1>
<section class="grid grid-template-rows-2">
  <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 class="item">7</div>
  <div class="item">8</div>
</section>
</div>

<div>
<h1>grid-template-rows: 1fr 2fr 1fr 3fr;</h1>
<section class="grid grid-template-rows-3">
  <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 class="item">7</div>
```

```
/* Grid */
.grid {
  display: grid;
  grid-template-columns: 1fr 1fr;
}

.grid-template-rows-1 {
  grid-template-rows: 50px 100px 50px 150px;
}

.grid-template-rows-2 {
  grid-template-rows: 100px 100px;
}

.grid-template-rows-3 {
  grid-template-rows: 1fr 2fr 1fr 3fr;
}

.grid-template-rows-4 {
  grid-template-columns: 100px auto 50px;
  grid-template-rows: 50px 200px auto;
}

/* Grid Item */
.item {
  margin: 5px;
  background: tomato;
  text-align: center;
  font-size: 1.5em;
}

.grid {
  max-width: 400px;
  margin: 0 auto;
  border: 1px solid #ccc;
}

h1 {
  text-align: center;
```

<pre> <div class="item">8</div> </section> </div> <div> <h1>grid-template-columns / grid-template-rows</h1> <section class="grid grid-template-rows-4"> <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 class="item">7</div> <div class="item">8</div> </section> </div> <div> <h1>grid-template-columns / grid-template-rows</h1> </div> </pre>	<pre> font-size: 1.25em; font-weight: normal; } body { font-family: monospace; color: #333; padding-bottom: 60px; margin: 0px; } body > div { padding: 40px 0; } body > div:nth-of-type(even) { background: rgba(0,0,0,.03); } body > div > img { display: block; margin: 0 auto; max-width: 100%; } </pre>
--	---

4. gap

Define o gap (espaçamento) entre os elementos do grid.

- **gap: 20px;**

```
// Define 20px entre os elementos do grid (linha e
coluna).
```

- **column-gap: 20px;**

```
// Define 20px de distância entre as colunas.
```

- **row-gap: 20px;**

```
// Define 20px de distância entre as linhas.
```

HTML

CSS

<pre> <div> <h1>gap: 20px;</h1> <section class="grid grid-template gap"> <div class="item logo">logo</div> <div class="item nav">nav</div> <div class="item content">content</div> </pre>	<pre> /* Grid */ .grid { display: grid; } .grid-template { grid-template: </pre>
---	--

<pre> <div class="item sidenav">sidenav</div> <div class="item advert">advert</div> <div class="item footer">footer</div> </section> </div> <div> <h1>column-gap: 20px;</h1> <section class="grid grid-template column-gap"> <div class="item logo">logo</div> <div class="item nav">nav</div> <div class="item content">content</div> <div class="item sidenav">sidenav</div> <div class="item advert">advert</div> <div class="item footer">footer</div> </section> </div> <div> <h1>row-gap: 20px;</h1> <section class="grid grid-template row-gap"> <div class="item logo">logo</div> <div class="item nav">nav</div> <div class="item content">content</div> <div class="item sidenav">sidenav</div> <div class="item advert">advert</div> <div class="item footer">footer</div> </section> </div> <div> <h1>margem nos itens</h1> <section class="grid grid-template gap"> <div class="item logo">logo</div> <div class="item nav">nav</div> <div class="item content">content</div> <div class="item sidenav">sidenav</div> <div class="item advert margin">advert</div> <div class="item footer">footer</div> </section> </div> </pre>	<pre> "logo nav nav" 50px "sidenav content advert" 150px "sidenav footer footer" 100px / 100px 1fr 50px; } .gap { gap: 20px; } .column-gap { gap: 2px; column-gap: 20px; } .row-gap { gap: 2px; row-gap: 20px; } .margin { margin-top: 10px; margin-left: 20px; margin-bottom: 50px; } .logo { grid-area: logo; } .nav { grid-area: nav; } .content { grid-area: content; } .sidenav { grid-area: sidenav; } .advert { grid-area: advert; } .footer { grid-area: footer; } /* Grid Item */ .item { background: tomato; text-align: center; font-size: 1.5em; } </pre>
---	--

```
.grid {
    max-width: 400px;
    margin: 0 auto;
    border: 1px solid #ccc;
}

h1 {
    text-align: center;
    font-size: 1.25em;
    font-weight: normal;
}

body {
    font-family: monospace;
    color: #333;
    padding-bottom: 60px;
    margin: 0px;
}

body > div {
    padding: 40px 0;
}

body > div:nth-of-type(even) {
    background: rgba(0,0,0,.03);
}

body > div > img {
    display: block;
    margin: 0 auto;
    max-width: 100%;
}
```

5 • grid

Atalho geral para definir o grid: grid-template-rows, grid-template-columns, grid-template-areas, grid-auto-rows, grid-auto-columns e grid-auto-flow

- `grid: 100px / 1fr 1fr;`

```
// Gera uma linha com 100px de altura e 2 colunas com 1fr.
```

- `grid: 100px / auto-flow 100px 50px;`

```
// Gera uma linha com 100px de altura.
```

```
// O grid-auto-flow é definido como column (pois está logo antes da definição das colunas). Ele também define o grid-auto-columns com 100px 50px
```

HTML

```
<div>
<h1>grid: 100px / 1fr 1fr;</h1>
<section class="grid grid-1">
  <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>
</section>
</div>

<div>
<h1>grid: 100px / auto-flow 100px 50px;</h1>
<section class="grid grid-2">
  <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>
</section>
</div>

<div>
<h1>grid diversos valores</h1>
<section class="grid grid-3">
  <div class="item logo">logo</div>
  <div class="item nav">nav</div>
  <div class="item content">content</div>
  <div class="item sidenav">sidenav</div>
  <div class="item advert">advert</div>
  <div class="item footer">footer</div>
</section>
</div>
```

CSS

```
/* Grid */
.grid {
  display: grid;
}

.grid-1 {
  grid: 100px / 1fr 1fr;
}

.grid-2 {
  grid: 100px / auto-flow 100px 50px;
}

.grid-3 {
  grid:
    "logo nav nav" 50px
    "sidenav content advert" 150px
    "sidenav footer footer" 100px
    / 100px 1fr 50px;
}

.logo {
  grid-area: logo;
}

.nav {
  grid-area: nav;
}

.content {
  grid-area: content;
}

.sidenav {
  grid-area: sidenav;
}

.advert {
  grid-area: advert;
}

.footer {
  grid-area: footer;
}

/* Grid Item */
.item {
  margin: 5px;
  background: tomato;
  text-align: center;
  font-size: 1.5em;
}
```

```
.grid {
    max-width: 400px;
    margin: 0 auto;
    border: 1px solid #ccc;
}

h1 {
    text-align: center;
    font-size: 1.25em;
    font-weight: normal;
}

body {
    font-family: monospace;
    color: #333;
    padding-bottom: 60px;
    margin: 0px;
}

body > div {
    padding: 40px 0;
}

body > div:nth-of-type(even) {
    background: rgba(0,0,0,.03);
}

body > div > img {
    display: block;
    margin: 0 auto;
    max-width: 100%;
}
```

6 • justify-content

Justifica os itens do grid em relação ao eixo x (horizontal).

- `justify-content: start;`

```
// Justifica os itens ao início.
```

- `justify-content: end;`

```
// Justifica os itens ao final.
```

- `justify-content: stretch;`

```
// Estica os itens.
```

- `justify-content: space-around;`

```
// Distribui espaço entre os elementos. (O início e final são menores que os espaços internos).
```

- `justify-content: space-between;`

// Cria um espaço entre os elementos, ignorando o início e final.

- `justify-content: space-evenly;`

// Cria um espaço igual entre as colunas (no início e final também).

- `justify-content: center;`

// Centraliza o conteúdo.

HTML

CSS

<pre> <div> <h1>justify-content: start;</h1> <section class="grid start"> <div class="item">1</div> <div class="item item-2">2</div> <div class="item item-3">3</div> <div class="item">4</div> <div class="item item-5">5</div> <div class="item">6</div> </section> </div> <div> <h1>justify-content: end;</h1> <section class="grid end"> <div class="item">1</div> <div class="item item-2">2</div> <div class="item item-3">3</div> <div class="item">4</div> <div class="item item-5">5</div> <div class="item">6</div> </section> </div> <div> <h1>justify-content: center;</h1> <section class="grid center"> <div class="item">1</div> <div class="item item-2">2</div> <div class="item item-3">3</div> <div class="item">4</div> <div class="item item-5">5</div> <div class="item">6</div> </section> </div> </pre>	<pre> /* Grid */ .grid { display: grid; grid-template-columns: repeat(3, 100px); grid-template-rows: 100px 100px 100px; } .start { justify-content: start; } .end { justify-content: end; } .center { justify-content: center; } .stretch { grid-template-columns: auto; justify-content: stretch; } .space-around { justify-content: space-around; } .space-between { justify-content: space-between; } .space-evenly { justify-content: space-evenly; } </pre>
---	---

<pre> <div> <h1>justify-content: stretch;</h1> <section class="grid stretch"> <div class="item">1</div> <div class="item item-2">2</div> <div class="item item-3">3</div> <div class="item">4</div> <div class="item item-5">5</div> <div class="item">6</div> </section> </div> <div> <h1>justify-content: space-around;</h1> <section class="grid space-around"> <div class="item">1</div> <div class="item item-2">2</div> <div class="item item-3">3</div> <div class="item">4</div> <div class="item item-5">5</div> <div class="item">6</div> </section> </div> <div> <h1>justify-content: space-between;</h1> <section class="grid space-between"> <div class="item">1</div> <div class="item item-2">2</div> <div class="item item-3">3</div> <div class="item">4</div> <div class="item item-5">5</div> <div class="item">6</div> </section> </div> <div> <h1>justify-content: space-evenly;</h1> <section class="grid space-evenly"> <div class="item">1</div> <div class="item item-2">2</div> <div class="item item-3">3</div> <div class="item">4</div> <div class="item item-5">5</div> <div class="item">6</div> </section> </div> </pre>	<pre> .item-2, .item-3 { grid-row: span 2; } .item-5 { grid-column: span 2; } /* Grid Item */ .item { margin: 5px; background: tomato; text-align: center; font-size: 1.5em; } .grid { max-width: 400px; margin: 0 auto; border: 1px solid #ccc; } h1 { text-align: center; font-size: 1.25em; font-weight: normal; } body { font-family: monospace; color: #333; padding-bottom: 60px; margin: 0px; } body > div { padding: 40px 0; } body > div:nth-of-type(even) { background: rgba(0,0,0,.03); } body > div > img { display: block; margin: 0 auto; max-width: 100%; } </pre>
---	---

7 • align-content

Alinha os itens do grid em relação ao eixo y (vertical).

- `align-content: start`
`// Alinha os itens ao início.`
- `align-content: end`
`// Alinha os itens ao final.`
- `align-content: stretch`
`// Estica os itens.`
- `align-content: space-around`
`// Distribui espaço entre os elementos. (O início e final são menores que os espaços internos).`
- `align-content: space-between`
`// Cria um espaço entre os elementos, ignorando o início e final.`
- `align-content: space-evenly`
`// Cria um espaço igual entre as colunas (no início e final também).`
- `align-content: center`
`// Centraliza o conteúdo.`

HTML

CSS

<pre> <div> <h1>align-content: start;</h1> <section class="grid start"> <div class="item">1</div> <div class="item item-2">2</div> <div class="item item-3">3</div> <div class="item">4</div> <div class="item item-5">5</div> <div class="item">6</div> </section> </div> <div> <h1>align-content: end;</h1> <section class="grid end"> <div class="item">1</div> <div class="item item-2">2</div> </pre>	<pre> /* Grid */ .grid { display: grid; grid-template-columns: repeat(3, 1fr); grid-template-rows: repeat(3, 50px); height: 300px; } .start { align-content: start; } .end { align-content: end; } .center { </pre>
---	--

<pre> <div class="item item-3">3</div> <div class="item">4</div> <div class="item item-5">5</div> <div class="item">6</div> </section> </div> <div> <h1>align-content: center;</h1> <section class="grid center"> <div class="item">1</div> <div class="item item-2">2</div> <div class="item item-3">3</div> <div class="item">4</div> <div class="item item-5">5</div> <div class="item">6</div> </section> </div> <div> <h1>align-content: stretch;</h1> <section class="grid stretch"> <div class="item">1</div> <div class="item item-2">2</div> <div class="item item-3">3</div> <div class="item">4</div> <div class="item item-5">5</div> <div class="item">6</div> </section> </div> <div> <h1>align-content: space-around;</h1> <section class="grid space-around"> <div class="item">1</div> <div class="item item-2">2</div> <div class="item item-3">3</div> <div class="item">4</div> <div class="item item-5">5</div> <div class="item">6</div> </section> </div> <div> <h1>align-content: space-between;</h1> <section class="grid space-between"> <div class="item">1</div> <div class="item item-2">2</div> <div class="item item-3">3</div> <div class="item">4</div> <div class="item item-5">5</div> <div class="item">6</div> </section> </div> <div> <h1>align-content: space-evenly;</h1> </pre>	<pre> align-content: center; } .stretch { grid-template-columns: auto; align-content: stretch; } .space-around { align-content: space-around; } .space-between { align-content: space-between; } .space-evenly { align-content: space-evenly; } .item-2, .item-3 { grid-row: span 2; } .item-5 { grid-column: span 2; } /* Grid Item */ .item { margin: 5px; background: tomato; text-align: center; font-size: 1.5em; } .grid { max-width: 400px; margin: 0 auto; border: 1px solid #ccc; } h1 { text-align: center; font-size: 1.25em; font-weight: normal; } body { font-family: monospace; color: #333; padding-bottom: 60px; margin: 0px; } body > div { padding: 40px 0; } </pre>
---	---

<pre> <section class="grid space-evenly"> <div class="item">1</div> <div class="item item-2">2</div> <div class="item item-3">3</div> <div class="item">4</div> <div class="item item-5">5</div> <div class="item">6</div> </section> </div> </pre>	<pre> } body > div:nth-of-type(even) { background: rgba(0,0,0,.03); } body > div > img { display: block; margin: 0 auto; max-width: 100%; } </pre>
---	---

8 • justify-items

Justifica o conteúdo dos itens do grid em relação ao eixo x (horizontal). Justifica em relação a célula.

- `justify-items: start`

```
// Justifica os itens ao início.
```
- `justify-items: end`

```
// Justifica os itens ao final.
```
- `justify-items: center`

```
// Centraliza o conteúdo.
```
- `justify-items: stretch`

```
// Estica os itens.
```

HTML

CSS

<pre> <div> <h1>justify-items: start;</h1> <section class="grid start"> <div class="item">Teste 1</div> <div class="item item-2">Ae 2</div> <div class="item item-3">Isso 3</div> <div class="item">Ab 4</div> <div class="item item-5">Aqui 5</div> <div class="item">Ela 6</div> </section> </div> <div> <h1>justify-items: end;</h1> <section class="grid end"> <div class="item">Teste 1</div> <div class="item item-2">Ae 2</div> </pre>	<pre> /* Grid */ .grid { display: grid; grid-template-columns: repeat(3, 1fr); grid-template-rows: repeat(3, 50px); } .start { justify-items: start; } .end { justify-items: end; } .center { justify-items: center; } </pre>
--	--

```

    <div class="item item-3">Isso 3</div>
    <div class="item">Ab 4</div>
    <div class="item item-5">Aqui 5</div>
    <div class="item">Ela 6</div>
</section>
</div>

<div>
<h1>justify-items: center;</h1>
<section class="grid center">
    <div class="item">Teste 1</div>
    <div class="item item-2">Ae 2</div>
    <div class="item item-3">Isso 3</div>
    <div class="item">Ab 4</div>
    <div class="item item-5">Aqui 5</div>
    <div class="item">Ela 6</div>
</section>
</div>

<div>
<h1>justify-items: stretch;</h1>
<section class="grid stretch">
    <div class="item">Teste 1</div>
    <div class="item item-2">Ae 2</div>
    <div class="item item-3">Isso 3</div>
    <div class="item">Ab 4</div>
    <div class="item item-5">Aqui 5</div>
    <div class="item">Ela 6</div>
</section>
</div>

```

```

}

.stretch {
    justify-items: stretch;
}

.item-2, .item-3 {
    grid-row: span 2;
}

.item-5 {
    grid-column: span 2;
}

/* Grid Item */
.item {
    margin: 5px;
    background: tomato;
    text-align: center;
    font-size: 1.5em;
}

.grid {
    max-width: 400px;
    margin: 0 auto;
    border: 1px solid #ccc;
}

h1 {
    text-align: center;
    font-size: 1.25em;
    font-weight: normal;
}

body {
    font-family: monospace;
    color: #333;
    padding-bottom: 60px;
    margin: 0px;
}

body > div {
    padding: 40px 0;
}

body > div:nth-of-type(even) {
    background: rgba(0,0,0,.03);
}

body > div > img {
    display: block;
    margin: 0 auto;
    max-width: 100%;
}

```

9 • align-items

Alinha o conteúdo dos itens do grid em relação ao eixo y (vertical). Alinha em relação a célula.

- `align-items: start`

`// Alinha os itens ao início.`

- `align-items: end`

`// Alinha os itens ao final.`

- `align-items: center`

`// Centraliza o conteúdo.`

- `align-items: stretch`

`// Estica os itens.`

• HTML

CSS

<pre><div> <h1>align-items: start;</h1> <section class="grid start"> <div class="item">Teste 1</div> <div class="item">Ae 2</div> <div class="item">Isso 3</div> <div class="item">Ab 4</div> <div class="item">Aqui 5</div> <div class="item">Ela 6</div> </section> </div> <div> <h1>align-items: end;</h1> <section class="grid end"> <div class="item">Teste 1</div> <div class="item">Ae 2</div> <div class="item">Isso 3</div> <div class="item">Ab 4</div> <div class="item">Aqui 5</div> <div class="item">Ela 6</div> </section> </div> <div> <h1>align-items: center;</h1> <section class="grid center"> <div class="item">Teste 1</div> <div class="item">Ae 2</div></pre>	<pre>/* Grid */ .grid { display: grid; grid-template-columns: repeat(3, 1fr); grid-template-rows: repeat(2, 100px); } .start { align-items: start; } .end { align-items: end; } .center { align-items: center; } .stretch { align-items: stretch; } .item-2, .item-3 { grid-row: span 2; } .item-5 { grid-column: span 2;</pre>
---	---

<pre> <div class="item">Isso 3</div> <div class="item">Ab 4</div> <div class="item">Aqui 5</div> <div class="item">Ela 6</div> </section> </div> <div> <h1>align-items: stretch;</h1> <section class="grid stretch"> <div class="item">Teste 1</div> <div class="item">Ae 2</div> <div class="item">Isso 3</div> <div class="item">Ab 4</div> <div class="item">Aqui 5</div> <div class="item">Ela 6</div> </section> </div> </pre>	<pre> } /* Grid Item */ .item { margin: 5px; background: tomato; text-align: center; font-size: 1.5em; } .grid { max-width: 400px; margin: 0 auto; border: 1px solid #ccc; } h1 { text-align: center; font-size: 1.25em; font-weight: normal; } body { font-family: monospace; color: #333; padding-bottom: 60px; margin: 0px; } body > div { padding: 40px 0; } body > div:nth-of-type(even) { background: rgba(0,0,0,.03); } body > div > img { display: block; margin: 0 auto; max-width: 100%; } </pre>
--	--

Grid Item

Os Grid Itens são os filhos diretos do Grid Container. Um grid item pode ser explícito ou implícito. Explícito é quanto você define ele explicitamente no container e implícito é quanto ele é criado automaticamente para preencher o conteúdo no grid.

1 • grid-column

Define quais colunas serão ocupadas pelo grid item. É possível definir uma linha de início e final, assim o item irá ocupar múltiplas colunas.

- `grid-column: 1`

`// O item ocupará a coluna 1.`

- `grid-column: 1 / 3`

`// O item ocupará a coluna 1 e 2 (Sim, isso mesmo, 1 e 2, pois os valores 1 / 3 são referentes as linhas da coluna. Isso significa que começa na linha 1 (início do grid) e vai até a linha 3, que é o começo da terceira coluna).`

- `grid-column-start: 2`

`// O item vai começar na linha 2.`

- `grid-column-end: 4`

`// O item vai terminar na linha 4.`

- `grid-column: span 2`

`// O item irá ocupar duas colunas a partir de onde ele estiver.`

HTML

CSS

<pre><div> <h1>grid-column</h1> <section class="grid grid-column-1"> <div class="item item-1">1</div> <div class="item item-2">2</div> <div class="item item-3">3</div> <div class="item item-4">4</div> <div class="item item-5">5</div> <div class="item item-6">6</div> </section> </div> <div> <h1>grid-column span</h1> <section class="grid grid-column-2"> <div class="item item-1">1</div> <div class="item item-2">2</div> <div class="item item-3">3</div> <div class="item item-4">4</div> <div class="item item-5">5</div></pre>	<pre>/* Grid */ .grid { display: grid; grid-template-columns: repeat(3, 1fr); grid-auto-rows: 50px; } /* 1 */ .grid-column-1 .item-1 { grid-column: 3; } .grid-column-1 .item-3 { grid-column: 1; } /* 2 */ .grid-column-2 .item-3 {</pre>
---	---

<pre> <div class="item item-6">6</div> </section> </div> <div> <h1>grid-column-start / grid-column-end</h1> <section class="grid grid-column-3"> <div class="item item-1">1</div> <div class="item item-2">2</div> <div class="item item-3">3</div> <div class="item item-4">4</div> <div class="item item-5">5</div> <div class="item item-6">6</div> </section> </div> <div> <h1>[col1-inicio]</h1> <section class="grid grid-column-4"> <div class="item item-1">1</div> <div class="item item-2">2</div> <div class="item item-3">3</div> <div class="item item-4">4</div> <div class="item item-5">5</div> <div class="item item-6">6</div> </section> </div> <div> <h1>grid-template-areas</h1> <section class="grid grid-column-5"> <div class="item item-1">1</div> <div class="item item-2">2</div> <div class="item item-3">3</div> <div class="item item-4">4</div> <div class="item item-5">5</div> </section> </div> <div> <h1>grid-column normal</h1> <section class="grid grid-column-6"> <div class="item item-1">1</div> <div class="item item-2">2</div> <div class="item item-3">3</div> <div class="item item-4">4</div> <div class="item item-5">5</div> </section> </div> </pre>	<pre> grid-column: 1 / -1; } .grid-column-2 .item-5 { grid-column: span 2; } .grid-column-2 .item-6 { grid-column: 2 / span 2; } /* 3 */ .grid-column-3 .item-2 { grid-column: 1 / 3; } .grid-column-3 .item-4 { grid-column-start: 1; grid-column-end: 3; } .grid-column-3 .item-1 { grid-column-start: 2; grid-column-end: 4; } /* 4 */ .grid-column-4 { grid-template-columns: [col1-inicio] 1fr [col1-final] 1fr; } .grid-column-4 .item-2 { grid-column: col1-inicio / col1-final; } /* 5 */ .grid-column-5 { grid-template-areas: "header header header" "sidenav content ads" "sidenav footer footer" ; } .grid-column-5 .item-1 { grid-column-start: header; grid-column-end: ads; } .grid-column-5 .item-2 { grid-column-start: sidenav; grid-column-end: sidenav; grid-row-start: sidenav; </pre>
--	--

```
        grid-row-end: sidenav;
    }

    .grid-column-5 .item-3 {
        grid-column: content / content;
    }

    .grid-column-5 .item-4 {
        grid-column-start: ads;
    }

    .grid-column-5 .item-5 {
        grid-column: footer;
    }

    /* 6 */

    .grid-column-6 .item-1 {
        grid-column: 1 / -1;
    }

    .grid-column-6 .item-2 {
        grid-column: 1;
        grid-row: 2 / 4;
    }

    .grid-column-6 .item-3 {
        grid-column: 2;
    }

    .grid-column-6 .item-4 {
        grid-column: 3;
    }

    .grid-column-6 .item-5 {
        grid-column: 2 / -1;
    }

    /* Grid Item */
    .item {
        margin: 5px;
        background: tomato;
        text-align: center;
        font-size: 1.5em;
    }

    .grid {
        max-width: 400px;
        margin: 0 auto;
        border: 1px solid #ccc;
    }

    h1 {
        text-align: center;
```

	<pre> font-size: 1.25em; font-weight: normal; } body { font-family: monospace; color: #333; padding-bottom: 60px; margin: 0px; } body > div { padding: 40px 0; } body > div:nth-of-type(even) { background: rgba(0,0,0,.03); } body > div > img { display: block; margin: 0 auto; max-width: 100%; } img { display: block; margin: 0 auto; } </pre>
--	--

2 • grid-row

Define quais linhas serão ocupadas pelo grid item.

- `grid-row: 1`

`// O item ocupará a linha 1.`

- `grid-row: 1 / 3`

`// O item ocupará a linha 1 e 2 (Sim, isso mesmo, 1 e 2, pois os valores 1 / 3 são referentes as linhas do grid. Isso significa que começa na linha 1 (início do grid) e vai até a linha 3 do grid, que é o começo da terceira linha).`

- `grid-row-start: 2`

`// O item vai começar na linha do grid 2.`

- `grid-row-end: 4`

// O item vai terminar na linha do grid 4.

- **grid-row: span 2**

// O item irá ocupar duas linhas a partir de onde ele estiver.

HTML

CSS

<pre><div> <h1>grid-row</h1> <section class="grid grid-row-1"> <div class="item item-1">1</div> <div class="item item-2">2</div> <div class="item item-3">3</div> <div class="item item-4">4</div> <div class="item item-5">5</div> <div class="item item-6">6</div> </section> </div> <div> <h1>grid-row span</h1> <section class="grid grid-row-2"> <div class="item item-1">1</div> <div class="item item-2">2</div> <div class="item item-3">3</div> <div class="item item-4">4</div> <div class="item item-5">5</div> <div class="item item-6">6</div> </section> </div> <div> <h1>grid-row-start / grid-row-end</h1> <section class="grid grid-row-3"> <div class="item item-1">1</div> <div class="item item-2">2</div> <div class="item item-3">3</div> <div class="item item-4">4</div> <div class="item item-5">5</div> <div class="item item-6">6</div> </section> </div> <div> <h1>[row1]</h1> <section class="grid grid-row-4"> <div class="item item-1">1</div> <div class="item item-2">2</div> <div class="item item-3">3</div> <div class="item item-4">4</div> <div class="item item-5">5</div></pre>	<pre>/* Grid */ .grid { display: grid; grid-template-columns: repeat(3, 1fr); grid-auto-rows: 50px; } /* 1 */ .grid-row-1 .item-1 { grid-row: 3; } .grid-row-1 .item-3 { grid-row: 1; } /* 2 */ .grid-row-2 .item-1 { grid-row: 1 / 4; } .grid-row-2 .item-5 { grid-row: span 3; } .grid-row-2 .item-6 { grid-row: span 2; } /* 3 */ .grid-row-3 .item-2 { grid-row: 1 / 3; } .grid-row-3 .item-4 { grid-row-start: 1; grid-row-end: 3; } .grid-row-3 .item-1 { grid-row-start: 2;</pre>
--	--

```

        <div class="item item-6">6</div>
    </section>
</div>

<div>
<h1>grid-template-areas</h1>
<section class="grid grid-row-5">
    <div class="item item-1">1</div>
    <div class="item item-2">2</div>
    <div class="item item-3">3</div>
    <div class="item item-4">4</div>
    <div class="item item-5">5</div>
</section>
</div>

```

```

        grid-row-end: 4;
    }

/* 4 */

.grid-row-4 {
    grid-template-rows: [row1] 1fr [row2] 1fr
[ row3] 1fr;
}

.grid-row-4 .item-2 {
    grid-row: row1 / row3;
}

.grid-row-4 .item-4 {
    grid-row: row1 / 4;
}

/* 5 */

.grid-row-5 {
    grid-template-areas:
        "header header header"
        "sidenav content ads"
        "sidenav footer footer"
    ;
}

.grid-row-5 .item-1 {
    grid-column: span 3;
    grid-row-start: header;
    grid-row-end: header;
}

.grid-row-5 .item-2 {
    grid-column: 1;
    grid-row-start: sidenav;
    grid-row-end: sidenav;
}

.grid-row-5 .item-3 {
    grid-row: content / content;
}

.grid-row-5 .item-4 {
    grid-row-start: ads;
}

.grid-row-5 .item-5 {
    grid-row: footer;
    grid-column: span 2;
}

/* Grid Item */
.item {
    margin: 5px;
    background: tomato;
}

```

	<pre> text-align: center; font-size: 1.5em; } .grid { max-width: 400px; margin: 0 auto; border: 1px solid #ccc; } h1 { text-align: center; font-size: 1.25em; font-weight: normal; } body { font-family: monospace; color: #333; padding-bottom: 60px; margin: 0px; } body > div { padding: 40px 0; } body > div:nth-of-type(even) { background: rgba(0,0,0,.03); } body > div > img { display: block; margin: 0 auto; max-width: 100%; } img { display: block; margin: 0 auto; } </pre>
--	---

3 • justify-self

Justifica o item do grid em relação ao eixo x (horizontal). Justifica em relação a célula.

- `justify-self: start`

`// Justifica o item ao início.`

- `justify-self: end`

`// Justifica o item ao final.`

- `justify-self: center`

`// Centraliza o conteúdo.`

- `justify-self: stretch`

`// Estica o item`

HTML

CSS

```
<div>
<h1>justify-self: start;</h1>
<section class="grid start">
  <div class="item item-1">1</div>
  <div class="item item-2">2</div>
  <div class="item item-3">3</div>
  <div class="item item-4">4</div>
  <div class="item item-5">5</div>
  <div class="item item-6">6</div>
</section>
</div>

<div>
<h1>justify-self: end;</h1>
<section class="grid end">
  <div class="item item-1">1</div>
  <div class="item item-2">2</div>
  <div class="item item-3">3</div>
  <div class="item item-4">4</div>
  <div class="item item-5">Item 5</div>
  <div class="item item-6">6</div>
</section>
</div>

<div>
<h1>justify-self: center;</h1>
<section class="grid center">
  <div class="item item-1">1</div>
  <div class="item item-2">2</div>
  <div class="item item-3">3</div>
  <div class="item item-4">4</div>
  <div class="item item-5">Item 5</div>
  <div class="item item-6">6</div>
</section>
</div>

<div>
<h1>justify-self: stretch;</h1>
<section class="grid stretch">
  <div class="item item-1">1</div>
  <div class="item item-2">2</div>
  <div class="item item-3">3</div>
```

```
/* Grid */
.grid {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  grid-template-rows: repeat(3, 50px);
}

/* 1 */
.start .item-2 {
  justify-self: start;
}

.start .item-5 {
  justify-self: start;
}

/* 2 */
.end .item-2 {
  justify-self: end;
}

.end .item-5 {
  justify-self: end;
}

/* 3 */
.center .item-2 {
  justify-self: center;
}

.center .item-5 {
  justify-self: center;
}

/* 4 */
.stretch {
  justify-items: start;
}
```

```
<div class="item item-4">4</div>
<div class="item item-5">Item 5</div>
<div class="item item-6">6</div>
</section>
</div>
```

```
.stretch .item-2 {
    justify-self: stretch;
}

.stretch .item-5 {
    justify-self: stretch;
}

.item-2, .item-3 {
    grid-row: span 2;
}

.item-5 {
    grid-column: span 2;
}

/* Grid Item */
.item {
    margin: 5px;
    background: tomato;
    text-align: center;
    font-size: 1.5em;
}

.grid {
    max-width: 400px;
    margin: 0 auto;
    border: 1px solid #ccc;
}

h1 {
    text-align: center;
    font-size: 1.25em;
    font-weight: normal;
}

body {
    font-family: monospace;
    color: #333;
    padding-bottom: 60px;
    margin: 0px;
}

body > div {
    padding: 40px 0;
}

body > div:nth-of-type(even) {
    background: rgba(0,0,0,.03);
}

body > div > img {
    display: block;
    margin: 0 auto;
    max-width: 100%;
}
```

4 • align-self

Justifica o item do grid em relação ao eixo y (vertical). Alinha em relação a célula.

- `align-self: start`

`// Alinha o item ao início.`

- `align-self: end`

`// Alinha o item ao final.`

- `align-self: center`

`// Centraliza o conteúdo.`

- `align-self: stretch`

`// Estica o item.`

HTML

CSS

<pre><div> <h1>align-self: start;</h1> <section class="grid start"> <div class="item item-1">1</div> <div class="item item-2">2</div> <div class="item item-3">3</div> <div class="item item-4">4</div> <div class="item item-5">5</div> <div class="item item-6">6</div> </section> </div> <div> <h1>align-self: end;</h1> <section class="grid end"> <div class="item item-1">1</div> <div class="item item-2">2</div> <div class="item item-3">3</div> <div class="item item-4">4</div> <div class="item item-5">Item 5</div> <div class="item item-6">6</div> </section> </div> <div> <h1>align-self: center;</h1> <section class="grid center"> <div class="item item-1">1</div> <div class="item item-2">2</div></pre>	<pre>/* Grid */ .grid { display: grid; grid-template-columns: repeat(3, 1fr); grid-template-rows: repeat(3, 50px); } /* 1 */ .start .item-2 { align-self: start; } .start .item-5 { align-self: start; } /* 2 */ .end .item-2 { align-self: end; } .end .item-5 { align-self: end; } /* 3 */</pre>
--	---

```
<div class="item item-3">3</div>
<div class="item item-4">4</div>
<div class="item item-5">Item 5</div>
<div class="item item-6">6</div>
</section>
</div>

<div>
<h1>align-self: stretch;</h1>
<section class="grid stretch">
  <div class="item item-1">1</div>
  <div class="item item-2">2</div>
  <div class="item item-3">3</div>
  <div class="item item-4">4</div>
  <div class="item item-5">Item 5</div>
  <div class="item item-6">6</div>
</section>
</div>
```

```
.center .item-2 {
  align-self: center;
}

.center .item-5 {
  align-self: center;
}

/* 4 */

.stretch {
  align-items: start;
}

.stretch .item-2 {
  align-self: stretch;
}

.stretch .item-5 {
  align-self: stretch;
}

.item-2, .item-3 {
  grid-row: span 2;
}

.item-5 {
  grid-column: span 2;
}

/* Grid Item */
.item {
  margin: 5px;
  background: tomato;
  text-align: center;
  font-size: 1.5em;
}

.grid {
  max-width: 400px;
  margin: 0 auto;
  border: 1px solid #ccc;
}

h1 {
  text-align: center;
  font-size: 1.25em;
  font-weight: normal;
}

body {
  font-family: monospace;
  color: #333;
  padding-bottom: 60px;
  margin: 0px;
}
```

	<pre>body > div { padding: 40px 0; } body > div:nth-of-type(even) { background: rgba(0,0,0,.03); } body > div > img { display: block; margin: 0 auto; max-width: 100%; }</pre>
--	--

Mas qual a diferença entre Flexbox e Grid??

Estude este material: <https://ishadeed.com/article/learn-box-alignment/>

Saiba mais:

https://www.w3schools.com/css/css_grid.asp

<https://www.origamid.com/projetos/css-grid-layout-guia-completo/>

<https://www.treinaweb.com.br/blog/css-grid-um-guia-interativo-parte-1-containers>

<https://triangulo.dev/posts/usando-grid-template-areas-do-css-grid/>

<https://triangulo.dev/posts/criando-layouts-responsivos-css-grid/>

<https://css-tricks.com/snippets/css/complete-guide-grid/>

https://developer.mozilla.org/pt-BR/docs/Web/CSS/CSS_Grid_Layout/Basic_Concepts_of_Grid_Layout

<https://youtu.be/HN1UjzRSdBk>

https://youtu.be/x-4z_u8LcGc

Divirta-se com os jogos:

<https://cssgridgarden.com/>

<https://www.espai.es/blog/2022/04/8-juegos-para-aprender-css/>

Façam estes exemplos do vídeo <https://www.youtube.com/watch?v=HN1UjzRSdBk>