

این جزوه برای آموزش CSS در مدرسه پرنیان آماده شده است. برای کشف تمام جزئیات آن میتوانید در دوره ها ثبت نام کرده و همراه ما باشید ❤️ <http://trainingsitedesign.ir>



flex

1. Container

- flex-direction
- flex-wrap
- flex-flow
- justify-content
- align-items
- align-content
- place-content => align-content و justify-content
- place-items => align-items و justify-items in grid

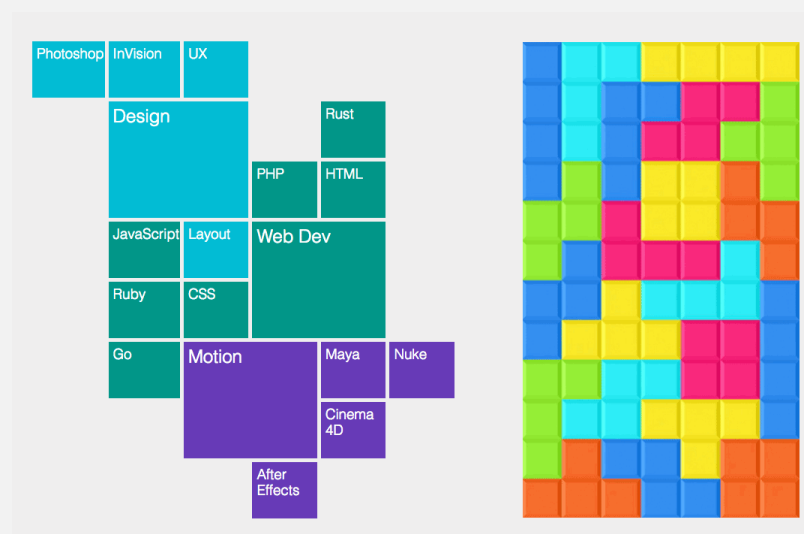
2. items

- order
- flex-grow
- flex-shrink
- flex-basis
- flex
- align-self

3. flex-responsive

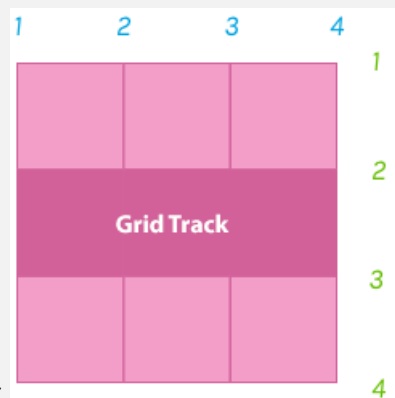
example by flex (slide)

<https://flexbox.help/>



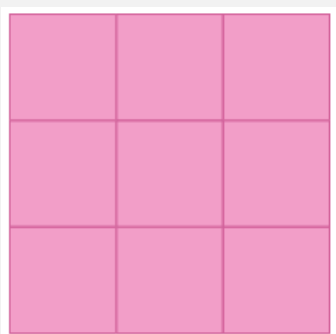
grid

The CSS Grid Layout Module offers a grid-based layout system, with rows and columns, making it easier to design web pages without having to use floats and positioning.



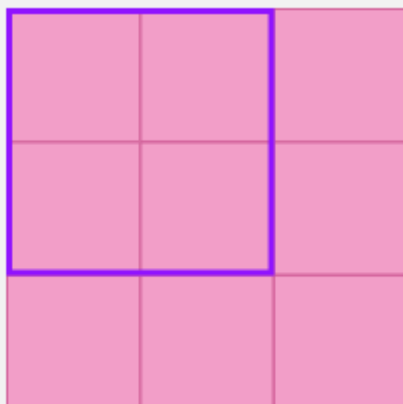
grid track

The space between two neighboring lines in the grid module is called grid track.



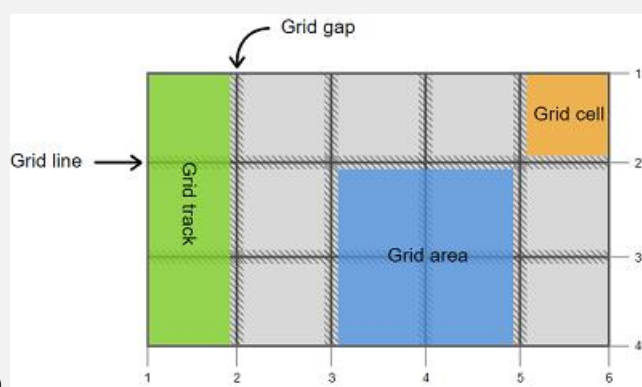
Grid Cell

The space between two horizontal lines next to each other and two vertical lines next to each other is called grid cell.



Grid Area

Any space between grid lines is called a grid area.



Grid Gap

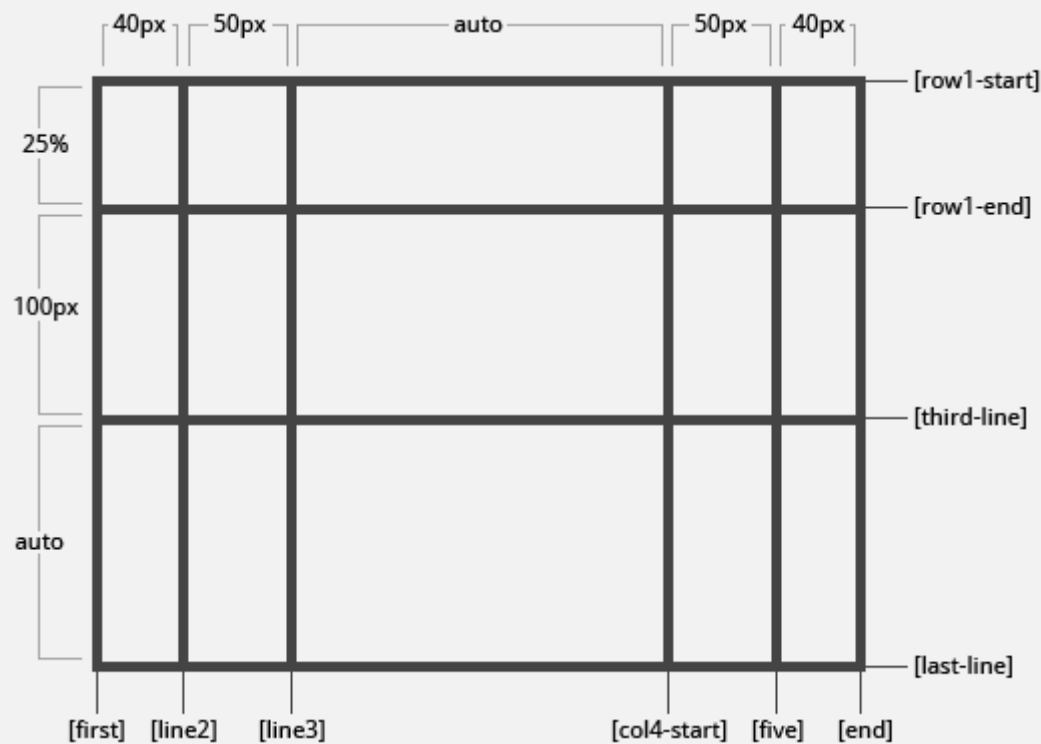
Column and row spacing

Cointainer:

- display: grid | inline-grid | subgrid;
- grid-template-columns
- grid-template-rows
 - (units: auto - px - % - fr(fraction) - min-content - max-content - minmax(100px, auto) - repeat(3, 1fr) - auto-fit - auto-fill)

```
grid-template-columns: [first] 40px [line2] 50px [line3] auto [col4-start] 50px [five] 40px [end];
```

```
grid-template-rows: [row1-start] 25% [row1-end] 100px [third-line] auto [last-line];
```



minmax is : This function takes two values, the first value is the minimum and the second value is the maximum size

auto fill باعث تولید خودکار ترک ها به تعدادی که از گرید سرریز نکنند می شود.

auto-fit مشابه auto-fill عمل می کند با این تفاوت که اندازه ترک های تکراری خالی را صفر در نظر می گیرد.

example : `grid-template-columns: repeat(auto-fill, minmax(40px, 1fr));`

grid-template is shorthand notation!

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

```
grid-template-rows: minmax(100px, auto) 100px 100px;
```

```
grid-template: minmax(100px, auto) 100px 100px / 1fr 1fr 1fr;
```

or:

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

```
grid-template-rows: 100px 200px 300px;
```

```
grid-template-areas: "header header header"
```

```
                    "sidebar main main"
```

```
                    "sidebar footer footer";
```

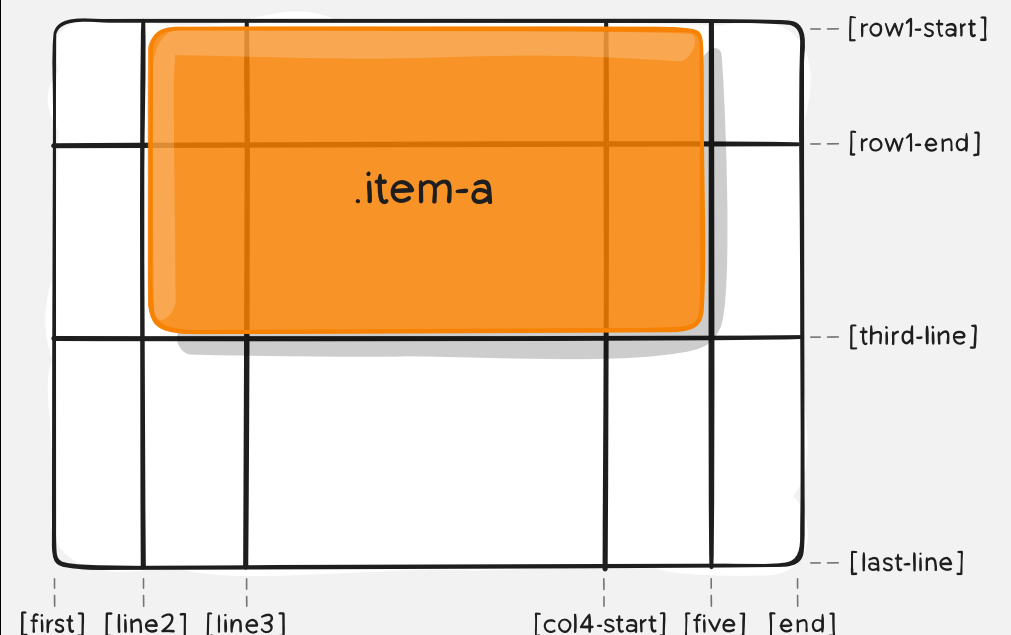
- column-gap
- row-gap
- gap
- grid-auto-columns
- grid-auto-rows
 - For unexpected rows or columns!
- grid-auto-flow: row
 - مقدار پیشفرض و اولیه این ویژگی row می باشد که باعث می شود آیتم های گرید به همان حالت عادی خود یعنی ردیفی کنار هم قرار بگیرند.
- grid-auto-flow: column
 - اگر مقدار این ویژگی را برابر با column قرار دهیم عناصر بصورت ستونی چیده شده
- grid-auto-flow: dense
 - کم استفاده، اگر جای خالی در ستون بندی داشته باشیم، با اولین سلول پر میکند

items:

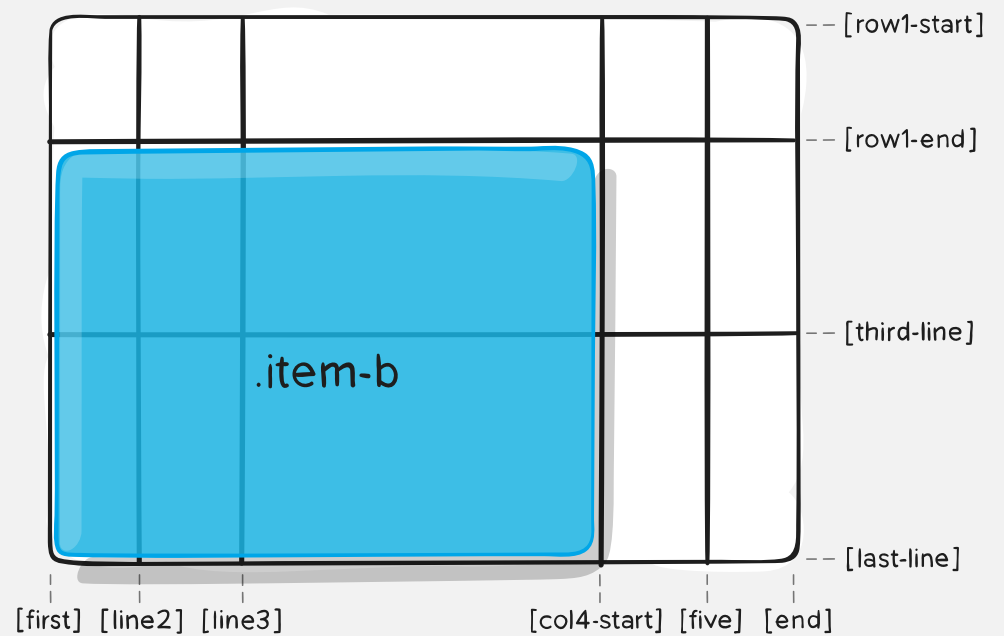
- grid-column-start
- grid-column-end
- grid-column
- grid-row-start
- grid-row-end
- grid-row
- grid-area

number or name or span

```
.item-a {
  grid-column-start: 2;
  grid-column-end: five;
  grid-row-start: row1-start;
  grid-row-end: 3;
}
```



```
.item-b {
  grid-column-start: 1;
  grid-column-end: span col4-start;
  grid-row-start: 2;
  grid-row-end: span 2;
}
```



alignment

- justify-content =>x
- align-content =>y
- justify-items =>into children X
- align-items =>into children Y
- align-self =>in one children, alignment content in column
- order =>Order of grid items
- justify-tracks => for alignment, but not support in chrome and ie and opera
- place-items => align-items + justify-items

```
header {

  grid-area: header;

}main {

  grid-area: main;

}aside {

  grid-area: aside;

}footer {

  grid-area: footer;

}body {

  grid-template-areas: "header header header header"

                        "main main . aside"

                        "footer footer footer footer";

}
```

<https://griddy.io/>