

[Bài đọc] Sử dụng bố cục trang với Flexbox

(Nguồn: *The Complete Guide to CSS Flexbox* | by Jobayer Hossain | Better Programming)

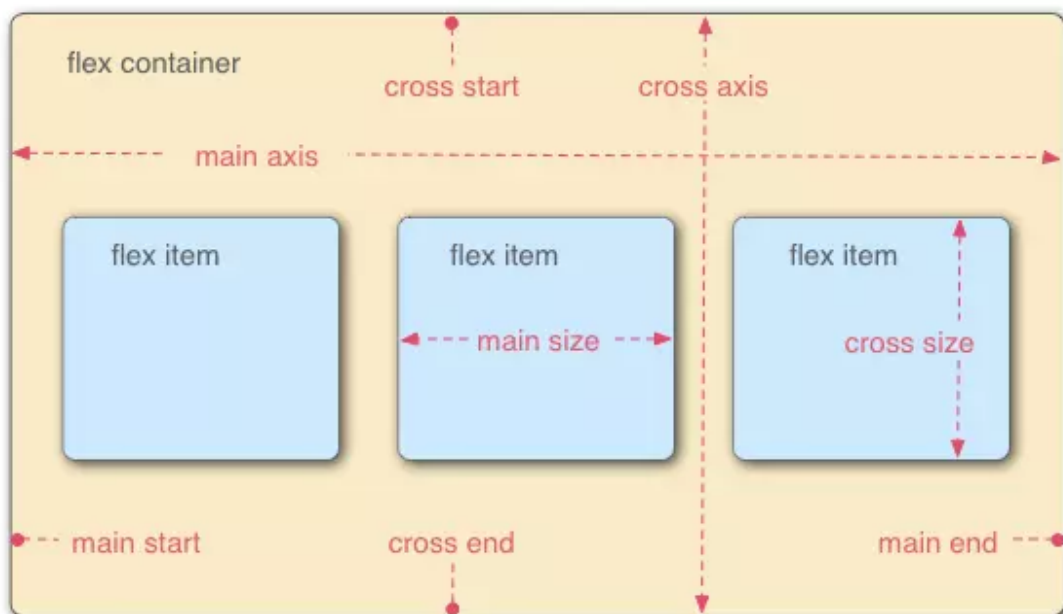
<https://medium.com/better-programming/a-complete-guide-css-flex-box-24f4a9a1e02>

CSS flexbox là một one-dimensional(hay còn gọi là 1D) layout pattern, một trong những pattern giúp bạn dễ dàng thiết kế layout một cách linh hoạt và hiệu quả. Phân chia không gian giữa các items và kiểm soát căn chỉnh chúng trong container flex layout(vùng chứa).

Với flexbox, chúng ta có thể dễ dàng sắp xếp các items từ trái sang phải, từ trên xuống dưới, đồng thời kiểm soát khoảng cách và thứ tự của các items trong container.

1. Làm thế nào để nó hoạt động?

Trước khi đi vào tìm hiểu sâu hơn về Flexbox, chúng ta cần nắm qua cấu trúc của Flexbox là như thế nào đã:



Trong flexbox thì chủ yếu có hai thành phần chính là: **thùng chứa cha** (*flex container*) và **các phần tử con** nằm bên trong (*flex items*).

Ngoài ra chúng ta cũng cần quan tâm tới một số thuộc tính sau:

- **main start, main end, cross start, cross end:** Điểm bắt đầu của container theo main axis được gọi là main start, điểm kết thúc của container theo main axis gọi là main end, với cross start và cross cũng tương tự nhưng dựa theo cross axis.
- **main axis:** Trục này chính là hướng của các item hiển thị, mặc định thì sẽ chạy từ trái qua phải.
- **cross axis:** Trục này vuông góc với main axis, chạy từ trên xuống dưới.
- **main size:** Là kích thước của mỗi item dựa theo trục main axis.
- **cross size:** Là kích thước của mỗi item dựa theo trục cross axis.

2. Các thuộc tính của flex container

Dưới đây là một số thuộc tính có thể sử dụng đối với flex container:

- display
- flex-direction
- flex-wrap
- flex-flow
- justified-content
- align-items
- align-content

Không sử dụng flexbox

```
<div class="box">  
  <div class="box-item">1</div>  
  <div class="box-item">2</div>  
  <div class="box-item">3</div>  
</div>
```



Dùng display để áp dụng flexbox

Chúng ta cần phải sử dụng thuộc tính `display`. Đây là cách mà chúng ta định nghĩa một flex container, và cũng là việc bắt buộc nếu bạn làm việc với flexbox.

```
<div class="box">
  <div class="box-item">1</div>
  <div class="box-item">2</div>
  <div class="box-item">3</div>
</div>
```

```
<style>
  .box {
    display: flex;
  }
</style>
```



flex-direction

`flex-direction` dùng để chỉ định hướng hiển thị của các item, việc thay đổi hướng hiển thị flex cũng có thể cho phép ta thay đổi vị trí của các flex item.

flex-direction: row

`flex-direction: row` là giá trị mặc định khi sử dụng flexbox, không thực hiện bất kỳ thay đổi nào, chỉ đặt các item từ trái qua phải theo trục chính.

```
<div class="box">
  <div class="box-item">1</div>
  <div class="box-item">2</div>
  <div class="box-item">3</div>
</div>
```

```
<style>
  .box {
    display: flex;
    flex-direction: row;
  }
</style>
```



flex-direction: row-reverse

Giống với tên gọi, `flex-direction: row-reverse` ngược lại với `row`, các item sẽ được đặt từ phải qua trái.

```
<div class="box">
  <div class="box-item">1</div>
  <div class="box-item">2</div>
  <div class="box-item">3</div>
</div>
```

```
<style>
  .box {
    display: flex;
    flex-direction: row-reverse;
  }
</style>
```



flex-direction: column

Khi chúng ta xét `flex-direction: column`, lúc này trục chính sẽ đi từ trên xuống dưới vậy nên giờ đây các items sẽ được xếp chồng lên nhau.

```
<div class="box">
  <div class="box-item">1</div>
  <div class="box-item">2</div>
  <div class="box-item">3</div>
</div>
```

```
<style>
  .box {
    display: flex;
    flex-direction: column;
  }
</style>
```



flex-direction: column-reverse

Khi đó các items sẽ được xếp chồng lên nhau nhưng theo chiều ngược lại. Hay để ý sẽ thấy ở ví dụ trên (1) sẽ ở trên cùng, nhưng khi sử dụng `column-reverse` (1) sẽ ở dưới cùng.

```
<div class="box">
  <div class="box-item">1</div>
  <div class="box-item">2</div>
  <div class="box-item">3</div>
</div>
```

```
<style>
  .box {
    display: flex;
    flex-direction: column-reverse;
  }
</style>
```



flex-wrap

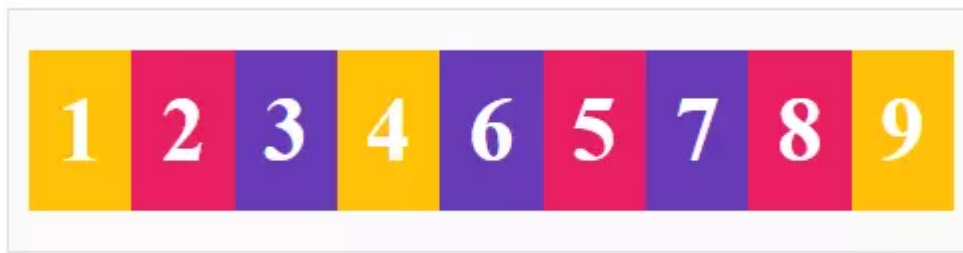
`flex-wrap` dùng để kiểm soát việc bọc các items nằm gọn trong container. Nếu chúng ta giảm chiều rộng của trình duyệt, chúng ta có thể không nhìn thấy một số item trên cùng một dòng. Thuộc tính `flex-wrap` có thể giải quyết vấn đề đó:

- nowrap (mặc định): Không có gì thay đổi
- wrap: các items sẽ được bọc tròn trong container
- wrap-reverse

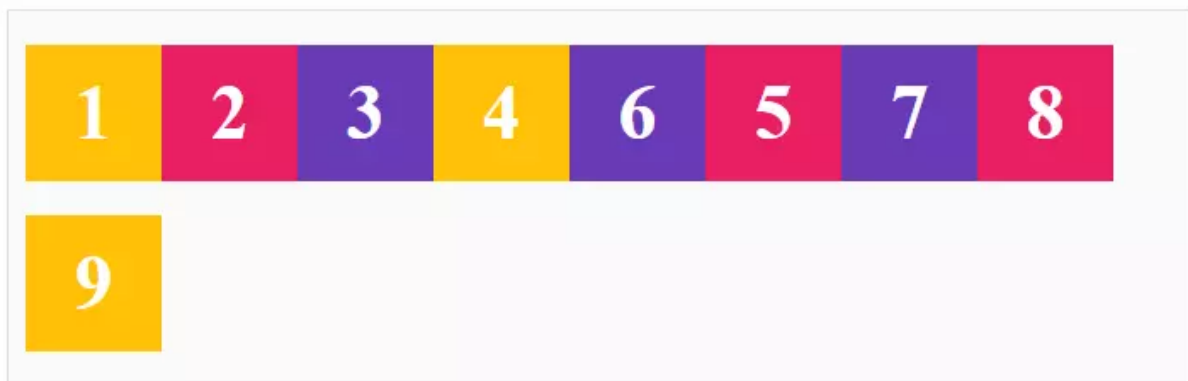
```
<div class="box">
  <div class="box-item">1</div>
  <div class="box-item">2</div>
  <div class="box-item">3</div>
  <div class="box-item">4</div>
  <div class="box-item">5</div>
  <div class="box-item">6</div>
  <div class="box-item">7</div>
  <div class="box-item">8</div>
  <div class="box-item">9</div>
</div>
```

```
<style>
  .box {
    display: flex;
    flex-wrap: nowrap;
  }
</style>
```

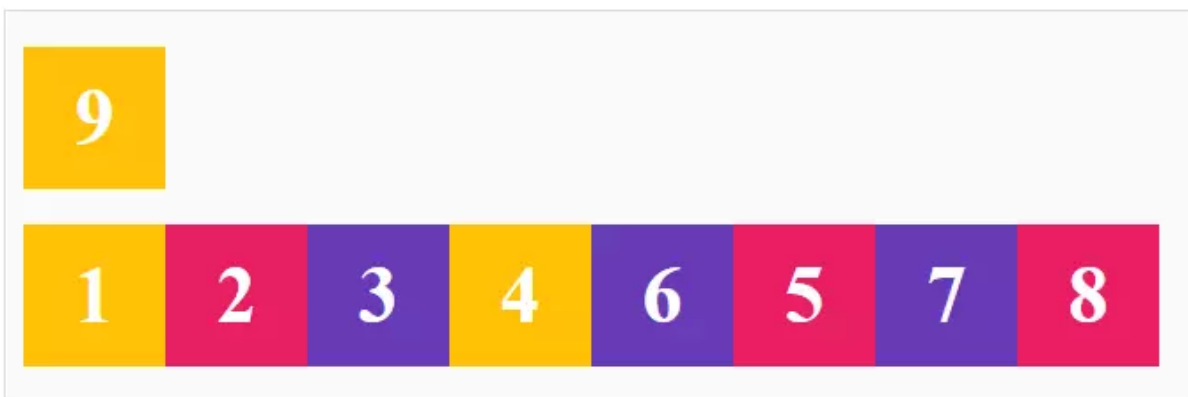
flex-wrap: nowrap



flex-wrap: wrap



flex-wrap: wrap-reverse



flex-flow

`flex-flow` là cách viết rút gọn của `flex-direction` và `flex-wrap`. Trong `flex-flow` giá trị đầu tiên là `flex-direction` và thứ 2 là `flex-wrap`

```
<div class="box">  
  <div class="box-item">1</div>  
  <div class="box-item">2</div>  
  <div class="box-item">3</div>
```

```
<div class="box-item">4</div>
<div class="box-item">5</div>
<div class="box-item">6</div>
<div class="box-item">7</div>
<div class="box-item">8</div>
<div class="box-item">9</div>
</div>
```

```
<style>
  .box {
    display: flex;
    flex-flow: row-reverse wrap;
  }
</style>
```



justified-content

`justified-content` dùng để căn chỉnh vị trí của các items so với trục chính(main axis). Có 6 giá trị có thể dùng đối với thuộc tính `justified-content`:

- `flex-start`: sẽ đặt item bắt đầu từ main start (và đây cũng là giá trị mặc định)
- `flex-end`: sẽ đặt item bắt đầu từ main end
- `center`: sẽ đặt tất cả item ở giữa trục main axis
- `space-between`: sẽ chia đều khoảng cách thừa và thêm nó vào giữa các item
- `space-around`: sẽ chia khoảng cách ở đầu và cuối. Khoảng cách ở đầu và cuối sẽ bằng 1 nửa khoảng cách ở giữa 2 item với nhau
- `space-evenly`: sẽ chia khoảng cách đều khoảng cách giữa các item với item, item và main start, item với main end bằng nhau

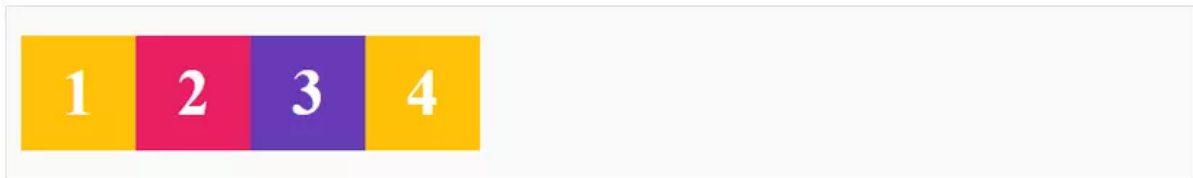
```
<div class="box">
  <div class="box-item">1</div>
  <div class="box-item">2</div>
  <div class="box-item">3</div>
  <div class="box-item">4</div>
</div>
```

```
<style>
  .box {
```

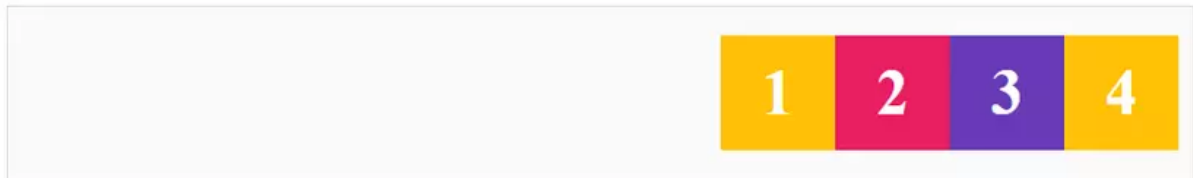


```
display: flex;  
justify-content: flex-start;  
}  
</style>
```

justify-content: flex-start



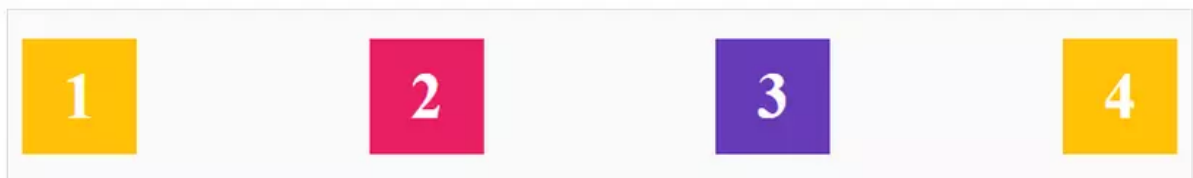
justify-content: flex-end



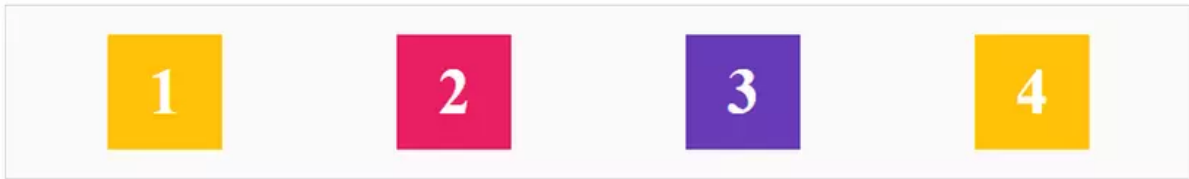
justify-content: center



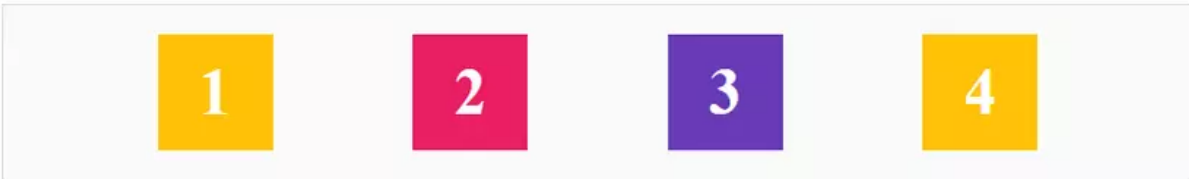
justify-content: space-between



justify-content: space-around



justified-content: space-evenly



align-items

Thuộc tính `align-items` dùng để xác định cách mà các flex item được đặt trong container dọc theo chiều cross axis.

- `align-items: stretch`: Chiều dài của item sẽ bằng chiều dài của cross axis.
- `align-items: flex-start`: Item được đặt ở điểm bắt đầu của cross start(trên cùng bên trái), và kích thước item không bị thay đổi.
- `align-items: flex-end`: Item được đặt ở điểm bắt đầu của cross end(dưới cùng bên trái).
- `align-items: center`: Item được đặt ở giữa điểm bắt đầu của cross start và điểm bắt đầu của cross end (ở giữa bên trái).
- `align-items: baseline`: Item sẽ được đặt dữ theo các ký tự thuộc item đó. Mục đích chính là căn chỉnh dữ liệu dòng văn bản của các item.

```
<div class="box">
  <div class="box-item">1</div>
  <div class="box-item">2</div>
  <div class="box-item">3</div>
  <div class="box-item">4</div>
</div>
```

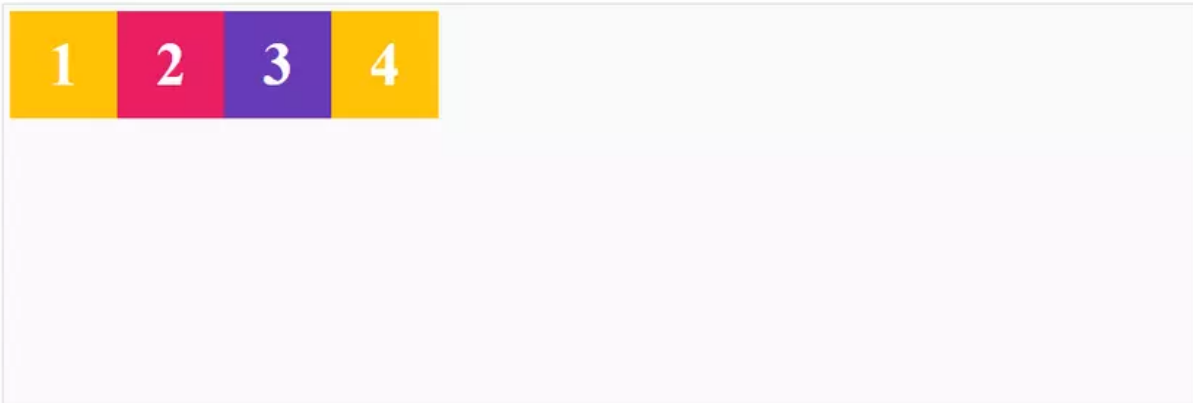
```
<style>
  .box {
    height: 300px;
    display: flex;
```

```
    align-items: stretch;  
  }  
</style>
```

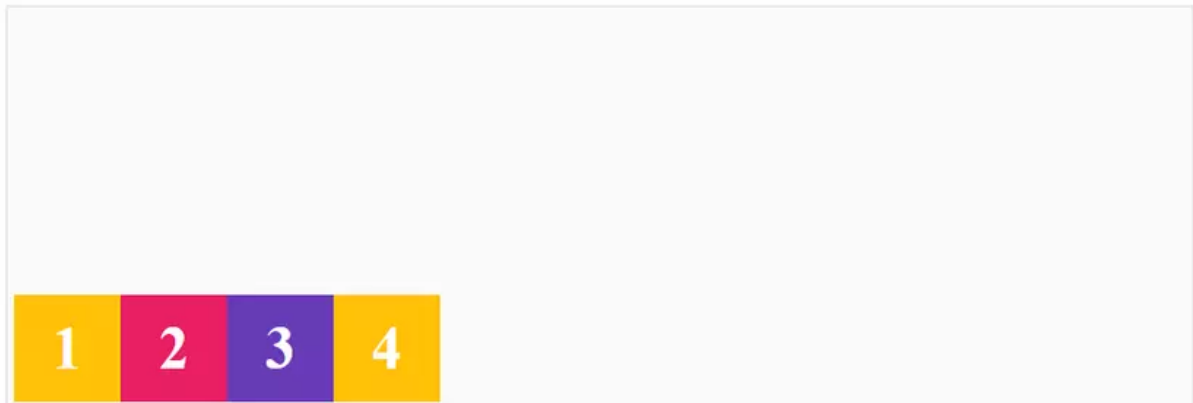
align-items: stretch



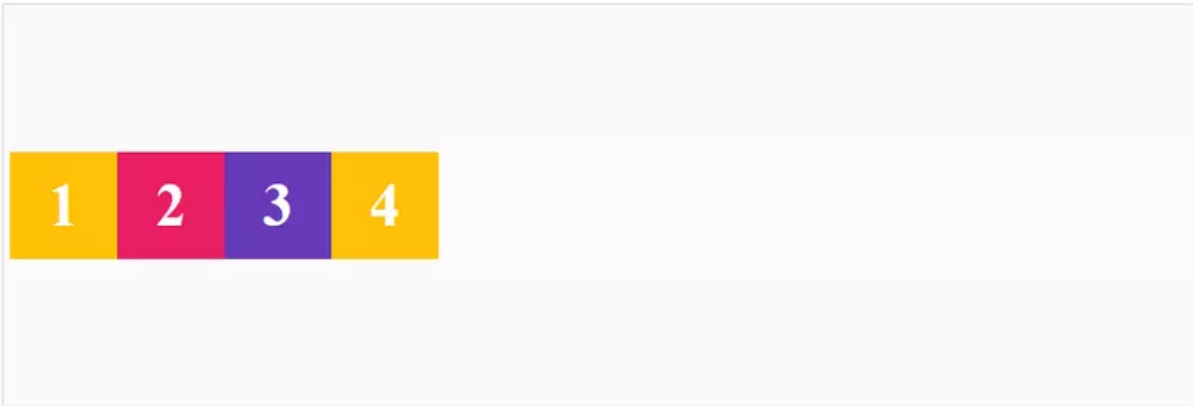
align-items: flex-start



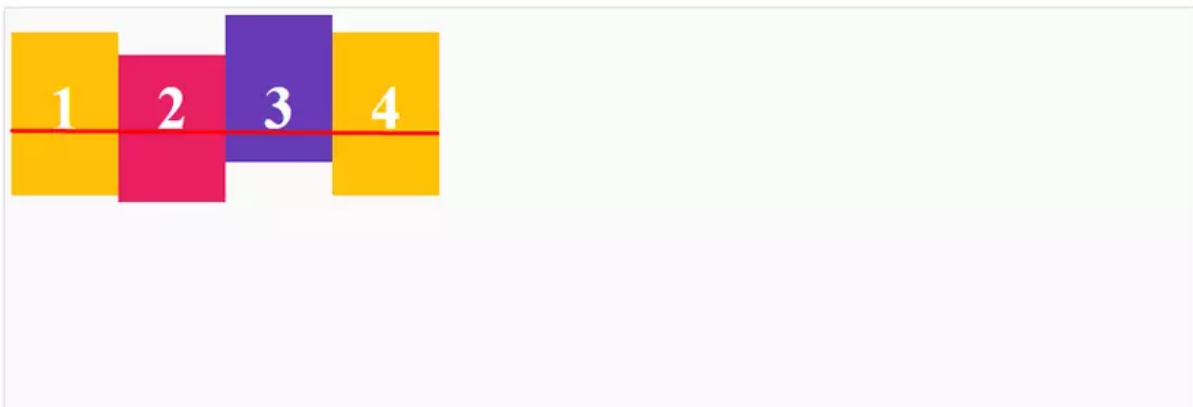
align-items: flex-end



align-items: center



align-items: baseline



align-content

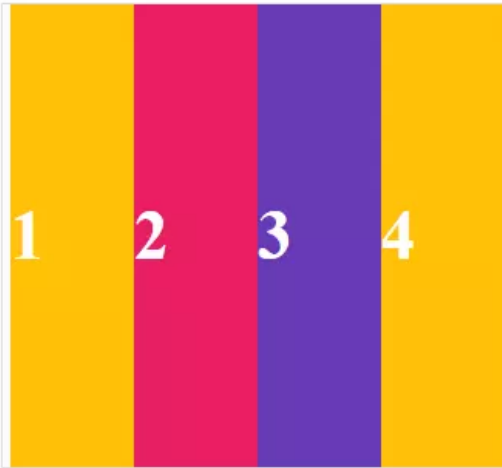
Tương tự như [justify-content](#) chỉ khác một chỗ là thay vì căn theo trục main axis thì [align-content](#) căn theo trục cross axis.

- align-content: stretch
- align-content: flex-start
- align-content: flex-end
- align-content: center
- align-content: space-between
- align-content: space-around

```
<div class="box">  
  <div class="box-item">1</div>  
  <div class="box-item">2</div>  
  <div class="box-item">3</div>  
  <div class="box-item">4</div>  
</div>
```

```
<style>  
  .box {  
    height: 300px;  
    display: flex;  
    flex-wrap: wrap;  
    align-content: stretch;  
  }  
</style>
```

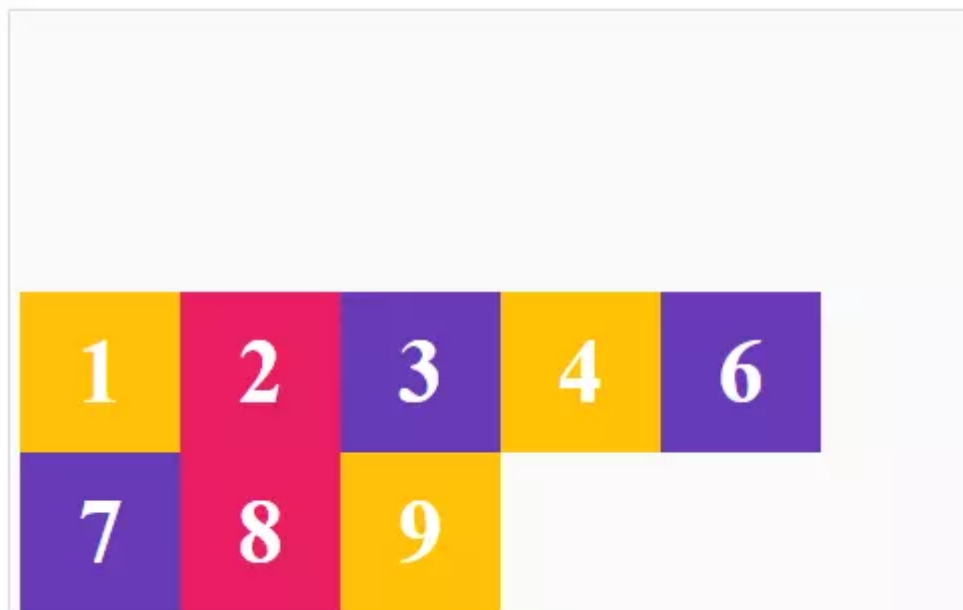
align-content: stretch



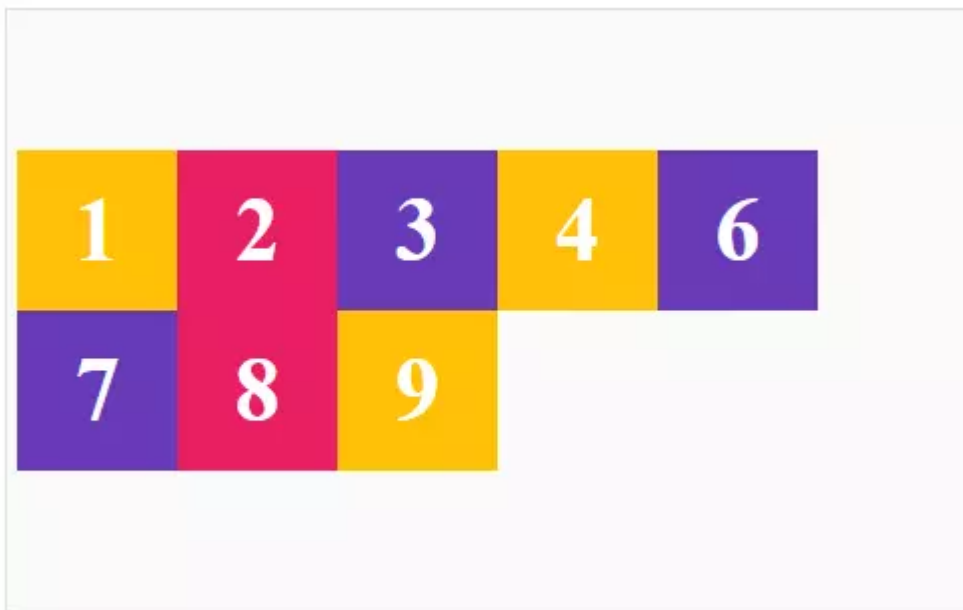
align-content: flex-start



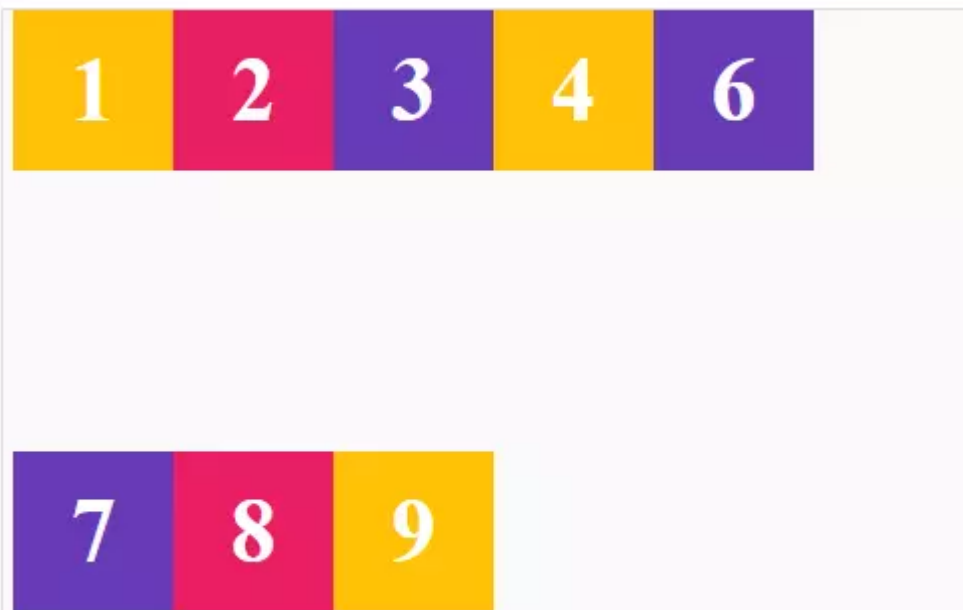
align-content: flex-end



align-content: center



align-content: space-between



align-content: space-around

