

INTERACTION PROGRAMMING 2

인터랙션 프로그래밍 2

CSS Flexbox

2018. 10. 11.

CSS Flexbox

레이아웃을 잡는 새로운 표준, 현대적인 방식

기존 CSS 레이아웃에 대한 기존 지식과는 많이 다르다.

어려워 보이지만, 어렵지 않습니다.

**flexbox 모델을 사용하려면
flex container 를 정의해야 한다.**

flex container

-

flex item flex item flex item

```
<!-- CSS Flexbox -->
<!-- flex Container -->
<ul><!-- Parent element -->
  <!-- flex Items -->
  <li>1</li><!-- Child element -->
  <li>2</li><!-- Child element -->
  <li>3</li><!-- Child element -->
</ul>
```

flexbox

flexbox

display : flex;

flexbox

```
/* flexbox */  
ul{display:flex}
```

flex container

flex container

flex-direction flex-wrap

flex-flow

justify-content align-items align-content

flex-direction

flex container :: flex-direction

flex-items 에 어떤 방향성을 줄 것인지 제어

row row-reverse

column column-reverse

flex container :: flex-direction

```
/* flex-direction */
```

```
ul{flex-direction: row | row-reverse | column | column-reverse}
```

flex container :: flex-direction

flex-direction : row;

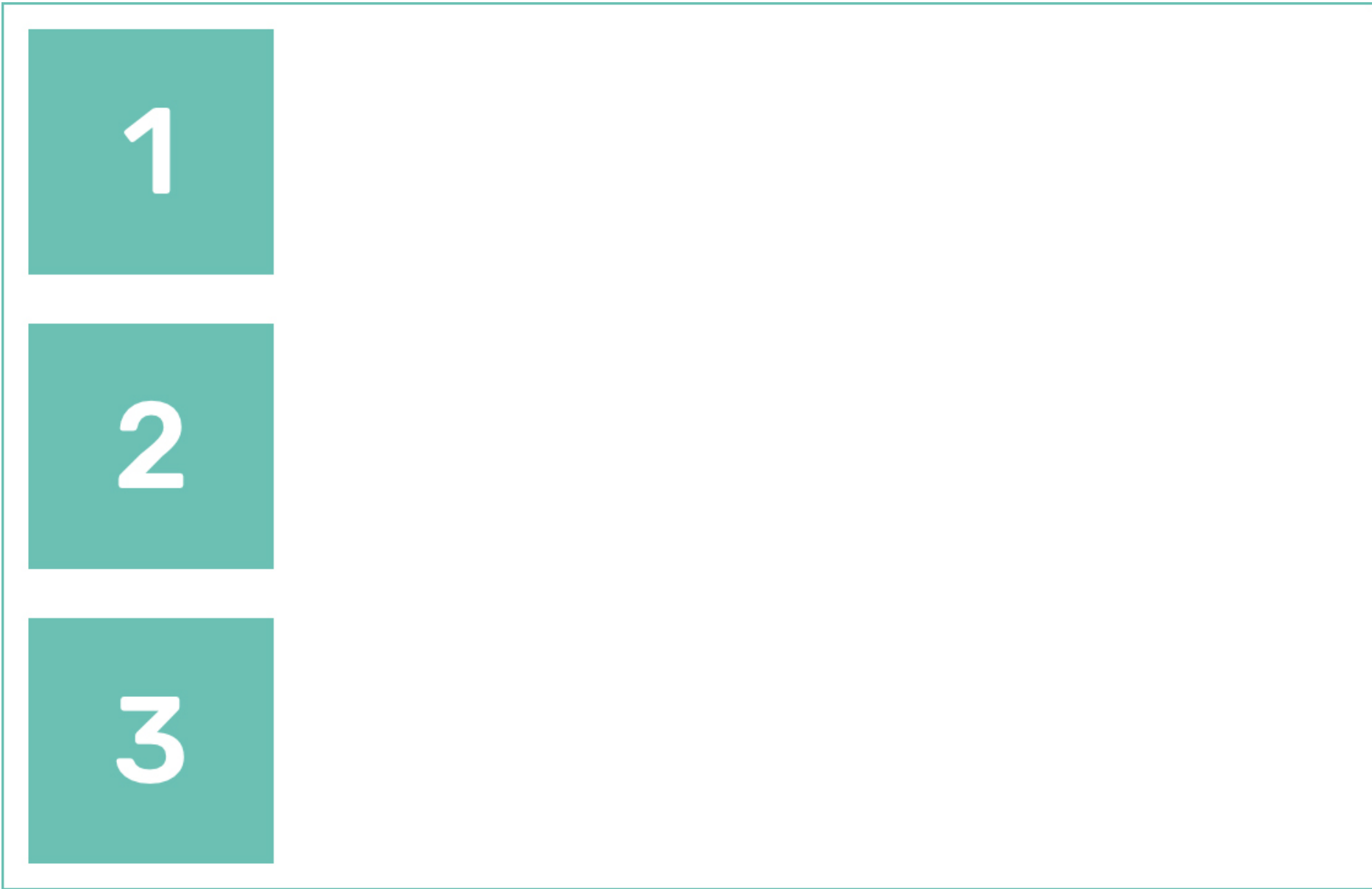


flex-direction : row-reverse;



flex container :: flex-direction

flex-direction : column;



flex-direction : column-reverse;



flex-wrap

flex container :: flex-wrap

flex-items 의 줄 넘김을 제어

nowrap wrap wrap-reverse

flex container :: flex-wrap

```
/* flex-wrap */
```

```
ul{flex-wrap: nowrap | wrap | wrap-reverse}
```

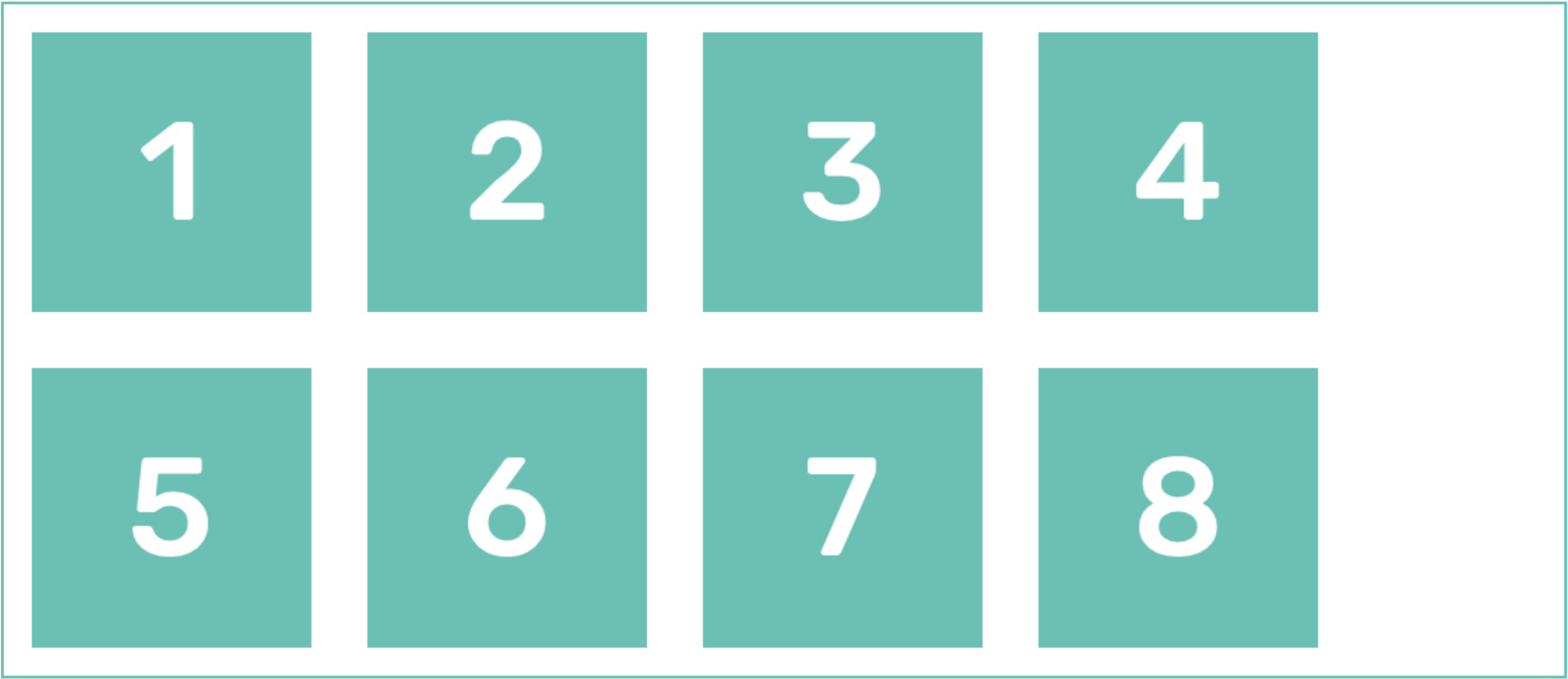
flex container :: flex-wrap

flex-wrap : nowrap;



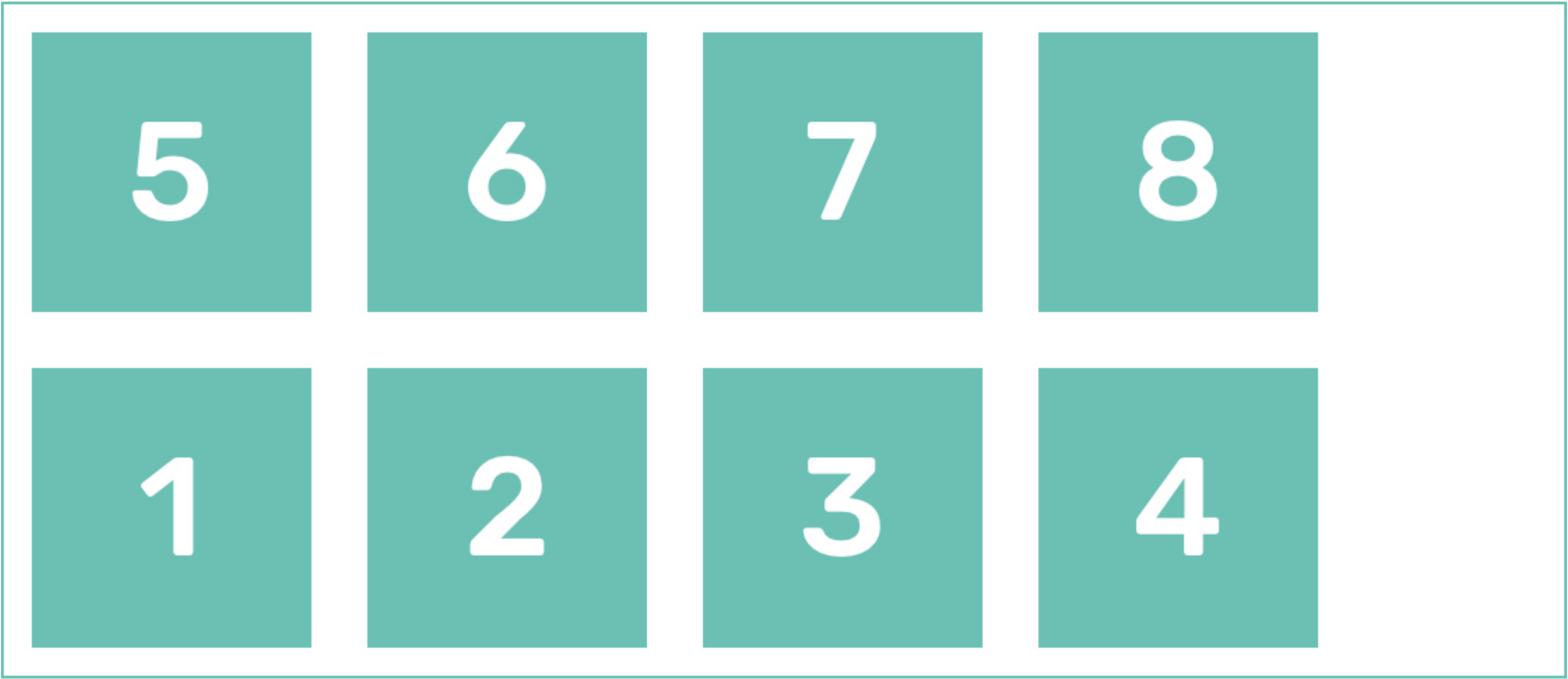
flex container :: flex-wrap

flex-wrap : wrap;



flex container :: flex-wrap

flex-wrap : wrap-reverse;



flex-flow

flex container :: flex-flow

flex-direction, flex-wrap 의 속성을 단축형으로 사용

flex container :: flex-flow

```
/* flex-flow */
```

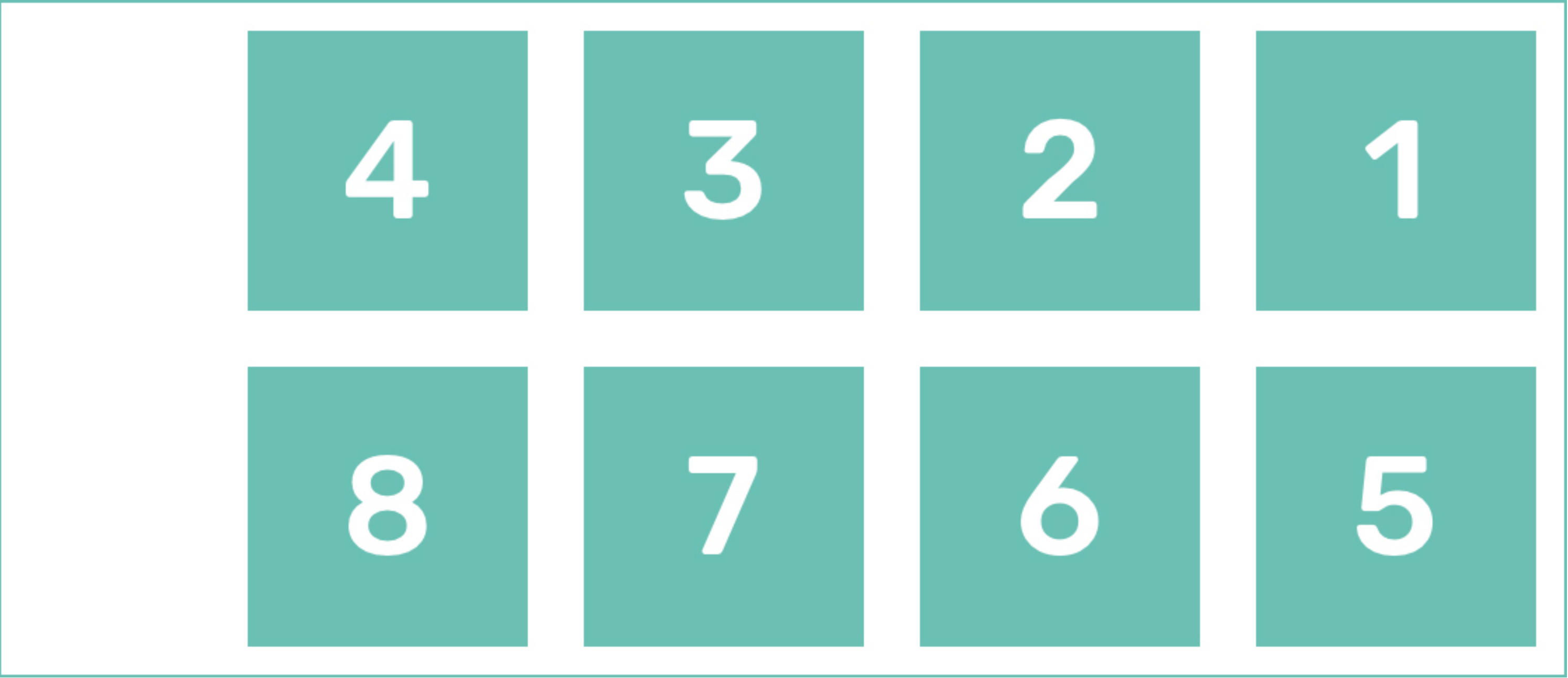
```
ul{flex-flow: <'flex-direction'> || <'flex-wrap'>}
```

flex container :: flex-flow

```
ul{flex-flow:row-reverse wrap}
```

flex container :: flex-flow

flex-flow : row-reverse wrap



justify-content

flex container :: justify-content

flex-container 에 빈 공간이 있을 경우,
flex-items 을 수평방향 (Main Axis) 으로 정렬하는 방식을 선언

flex-start flex-end

center

space-between space-around

flex container :: justify-content

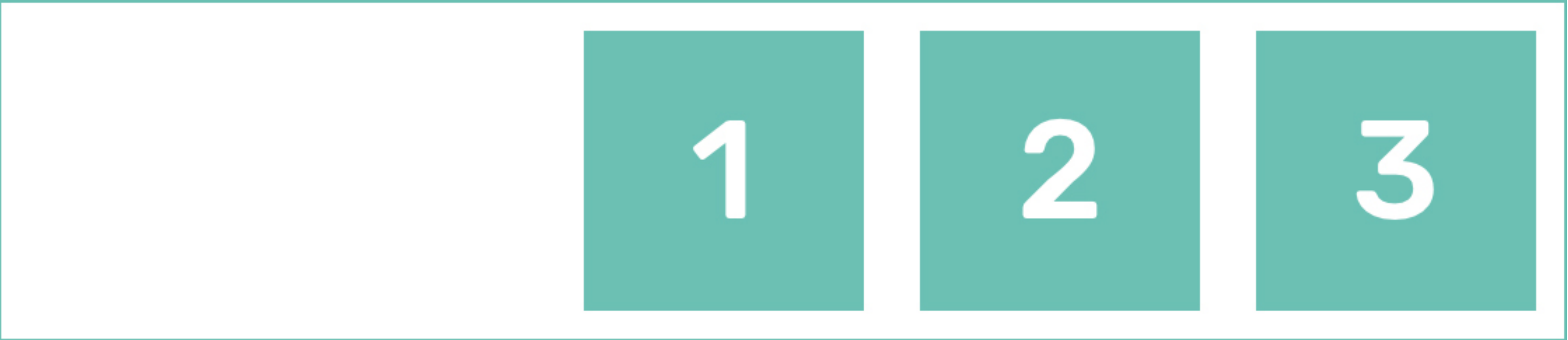
```
/* justify-content */
ul{justify-content: flex-start | flex-end | center | space-between | space-around | space-evenly}
```

flex container :: justify-content

justify-content : flex-start;



justify-content : flex-end;



flex container :: justify-content

justify-content : center;



justify-content : space-between;



flex container :: justify-content

justify-content : space-around;



justify-content : space-evenly;



align-items

flex container :: align-items

flex-container 에 빈 공간이 있을 경우,
flex-items 을 수직방향 (Cross Axis) 으로 정렬하는 방식을 선언

flex-start flex-end

center

baseline stretch

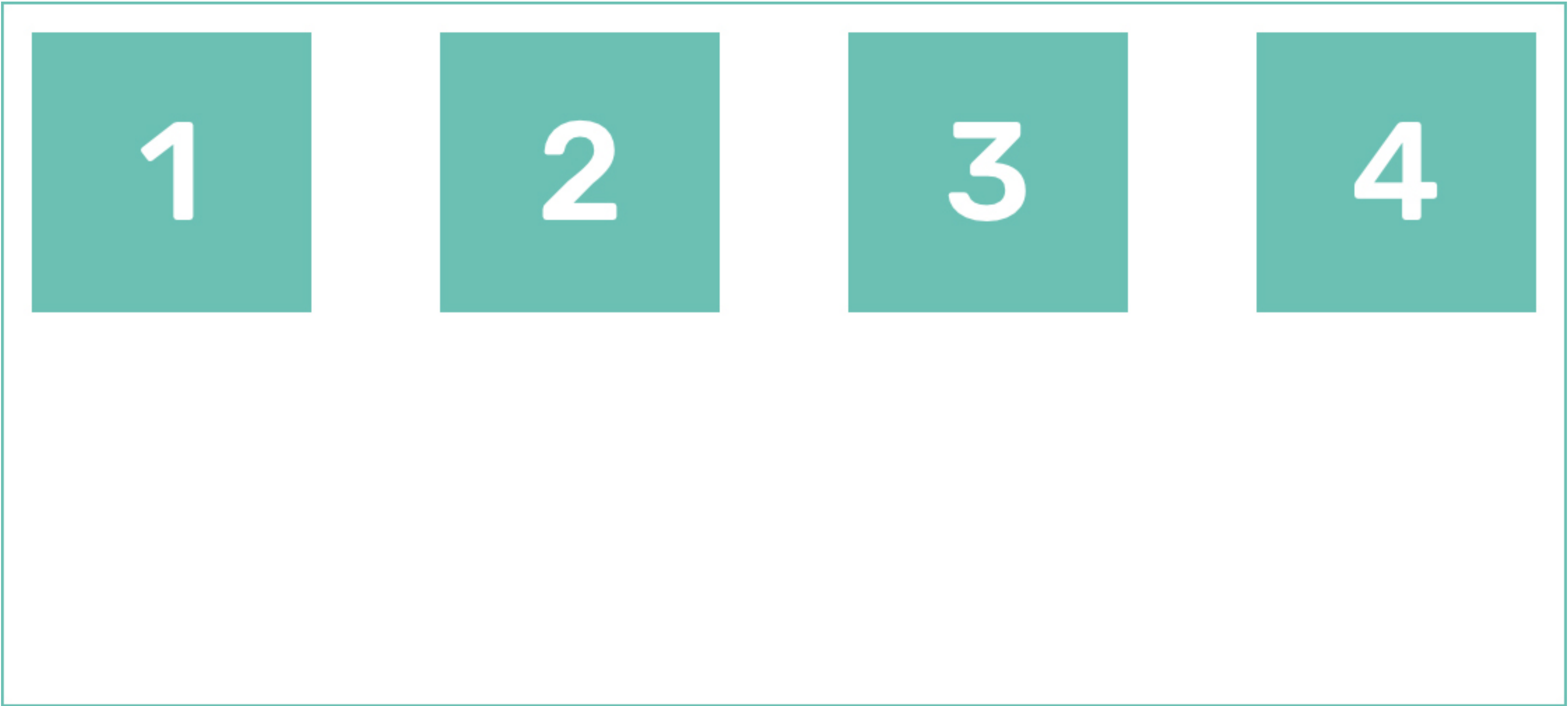
flex container :: align-items

```
/* align-items */
```

```
ul{align-items: flex-start | flex-end | center | baseline | stretch}
```

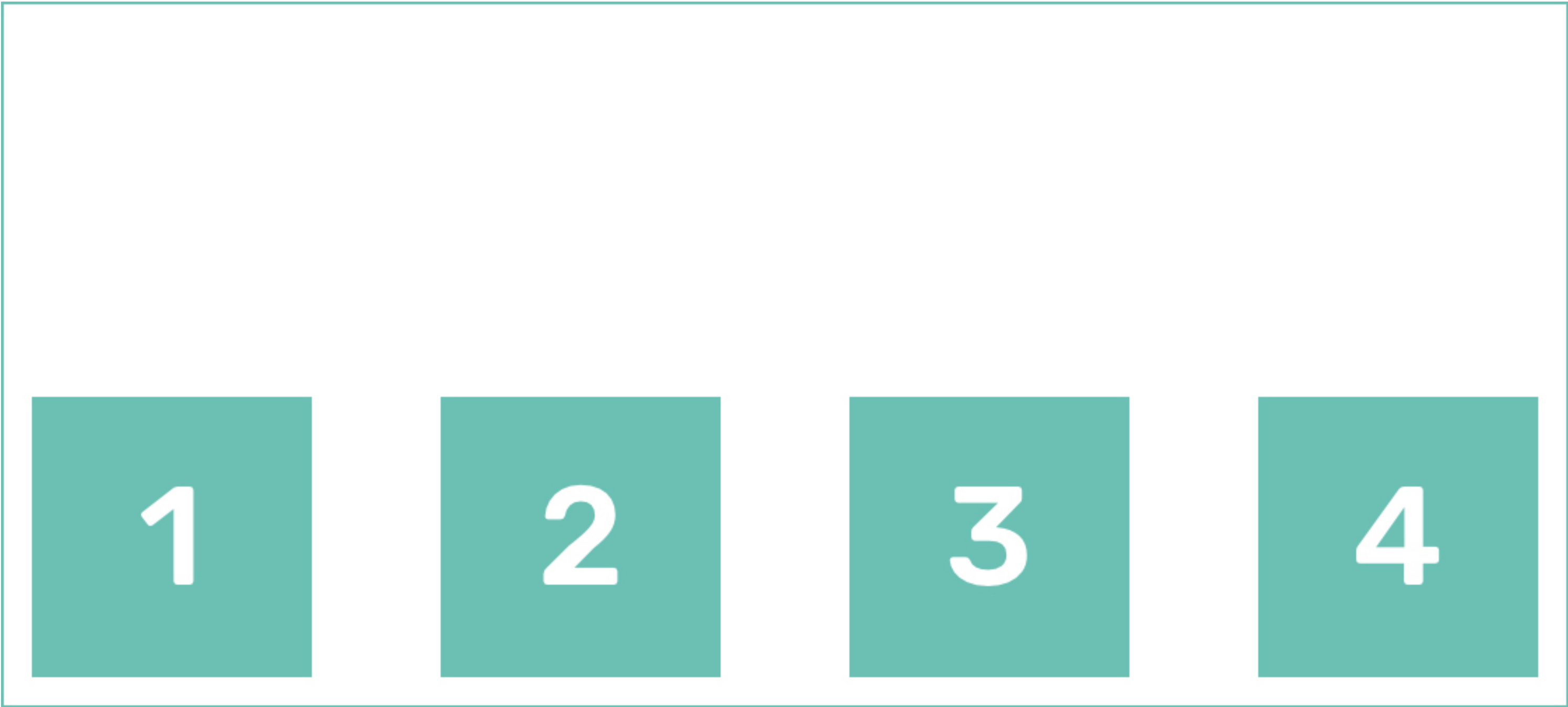
flex container :: align-items

align-items : flex-start;



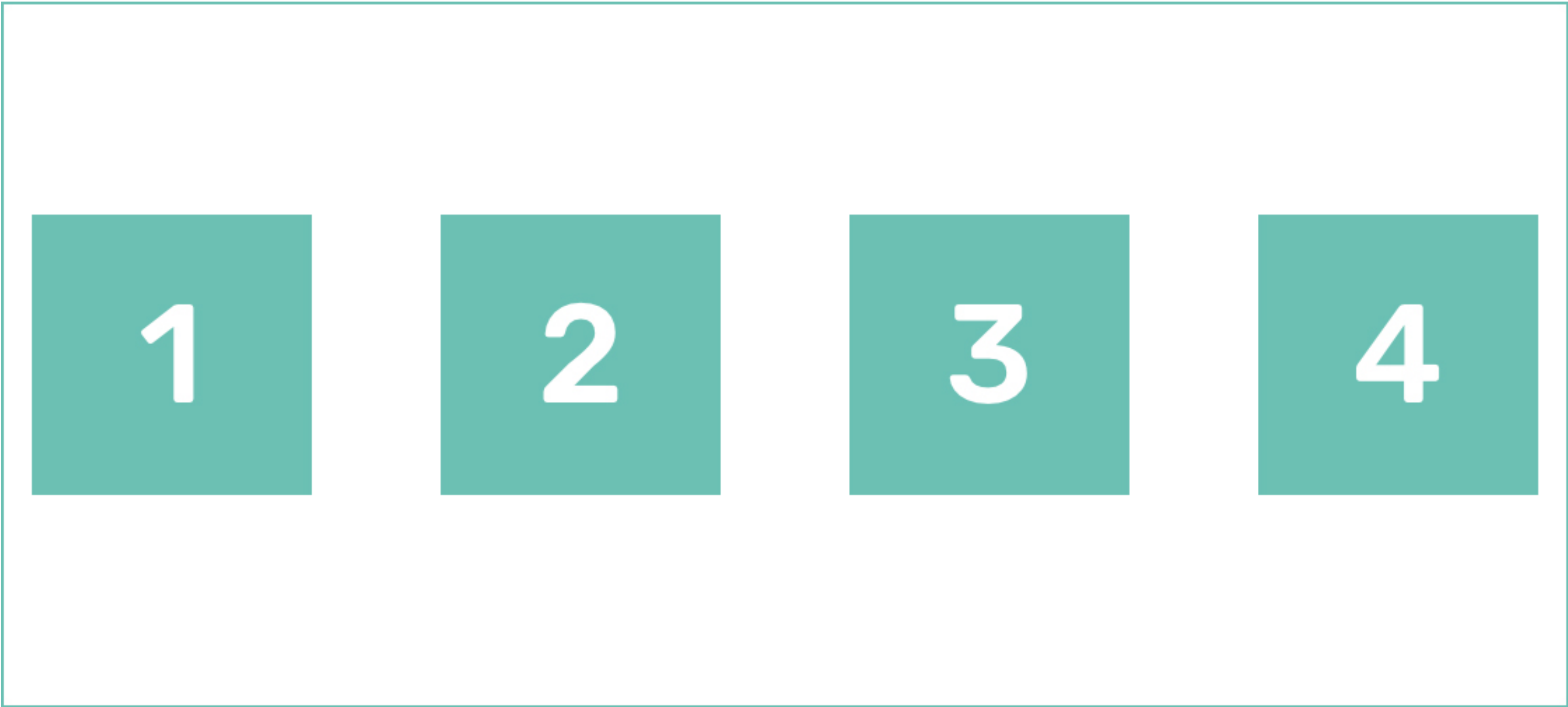
flex container :: align-items

align-items : flex-end;



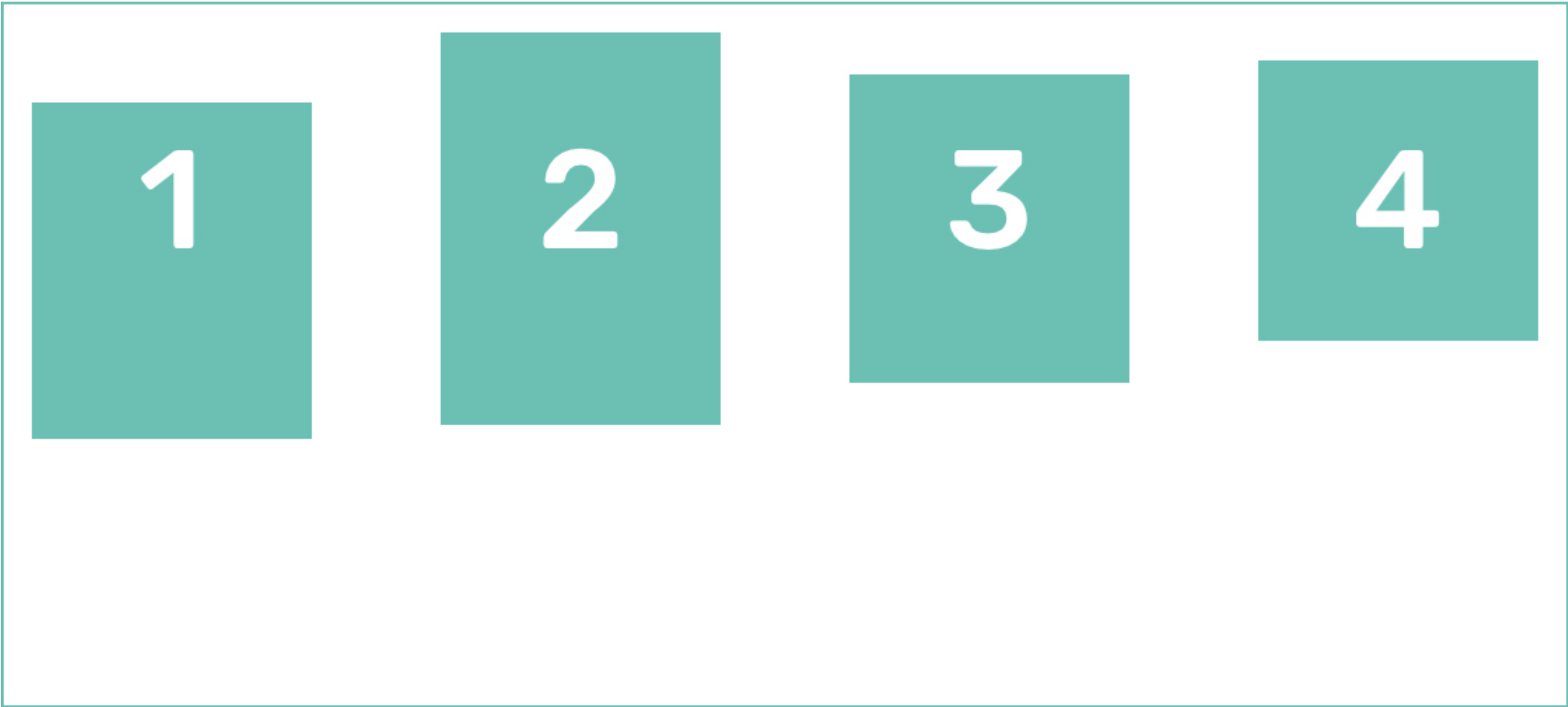
flex container :: align-items

align-items : center;



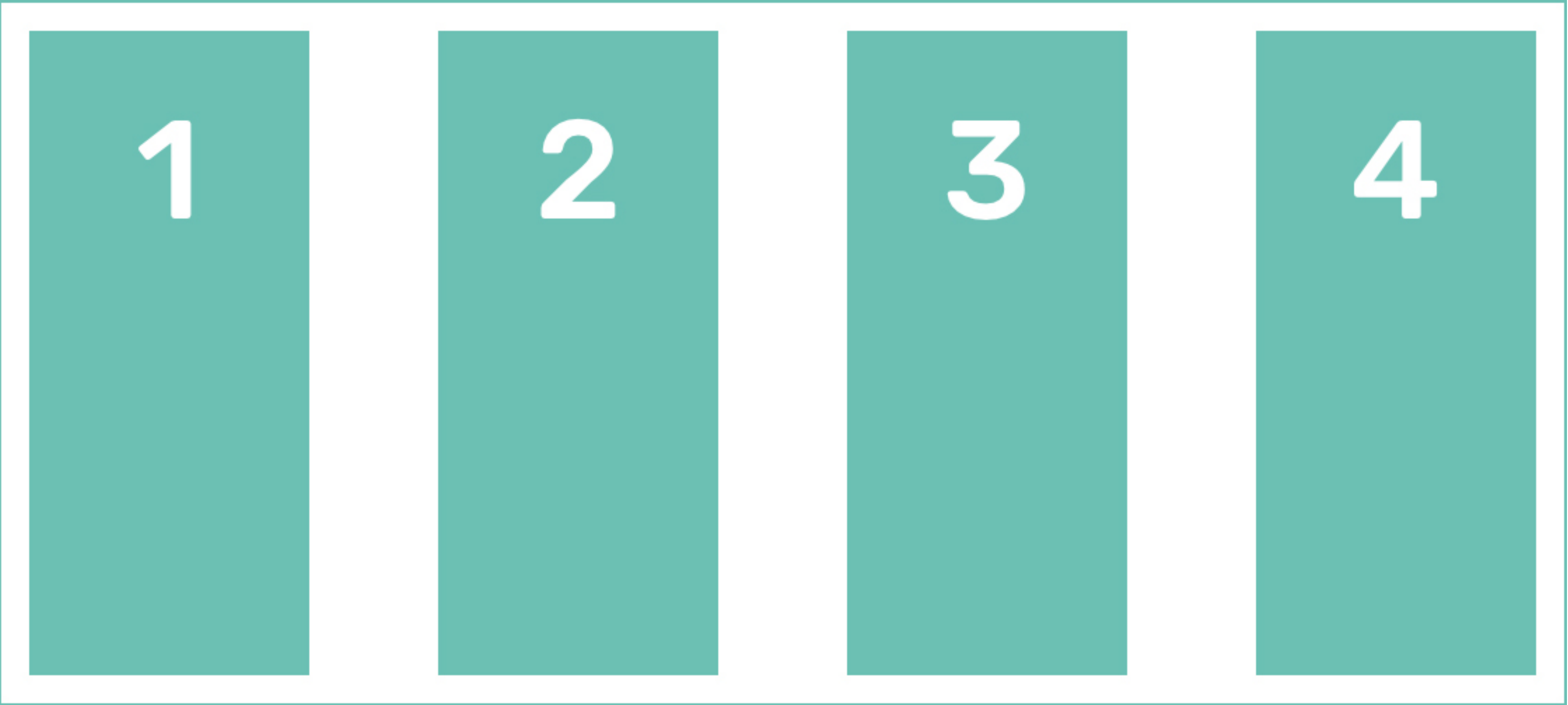
flex container :: align-items

align-items : baseline;



flex container :: align-items

align-items : stretch;



align-content

flex container :: align-content

flex-container 에 빈 공간이 있을 경우,
한 줄을 넘기는 flex-items 을 수직방향 (Cross Axis) 으로 정렬하는 방식을 선언

flex-start flex-end

center

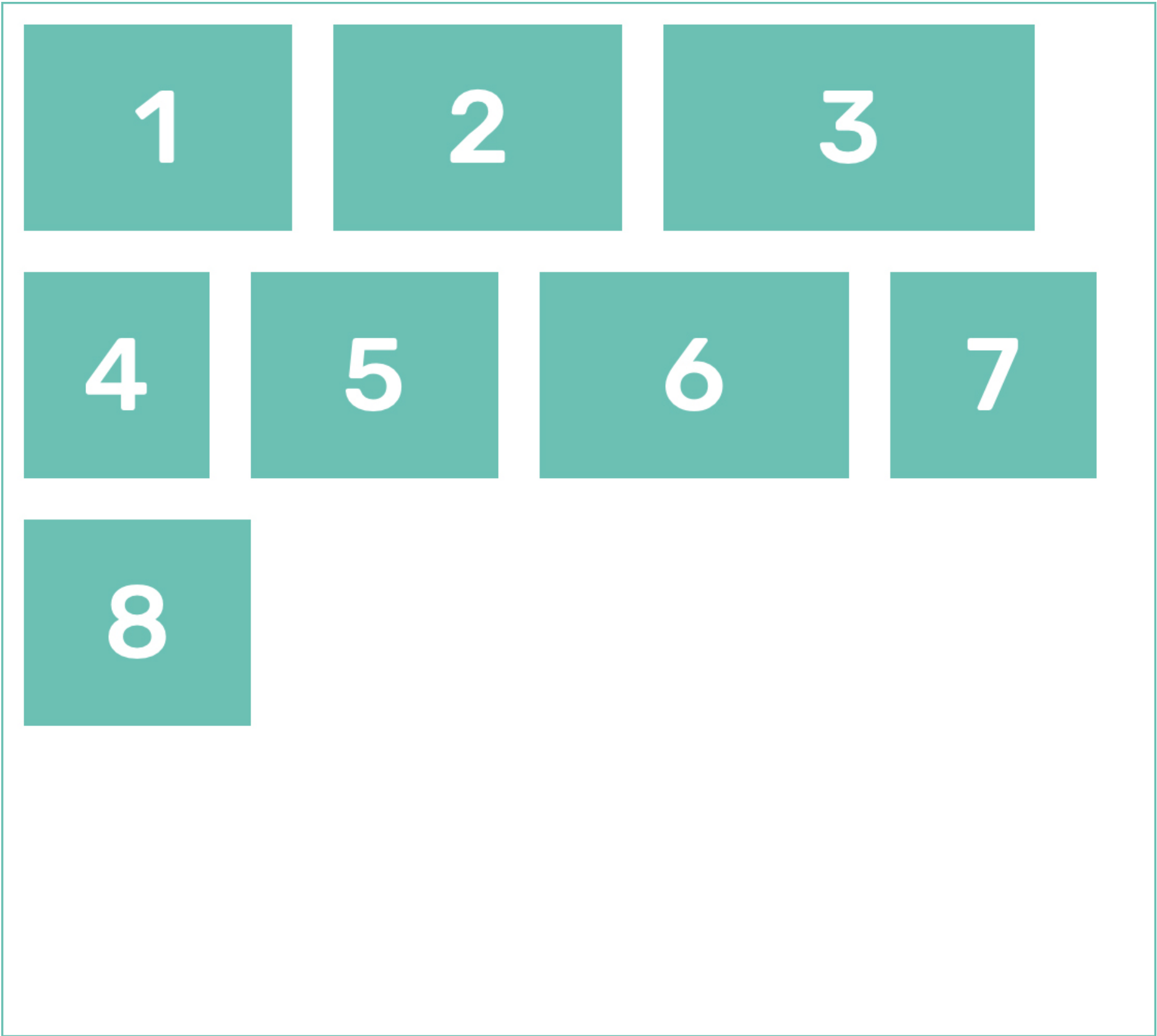
space-between space-around stretch

flex container :: align-content

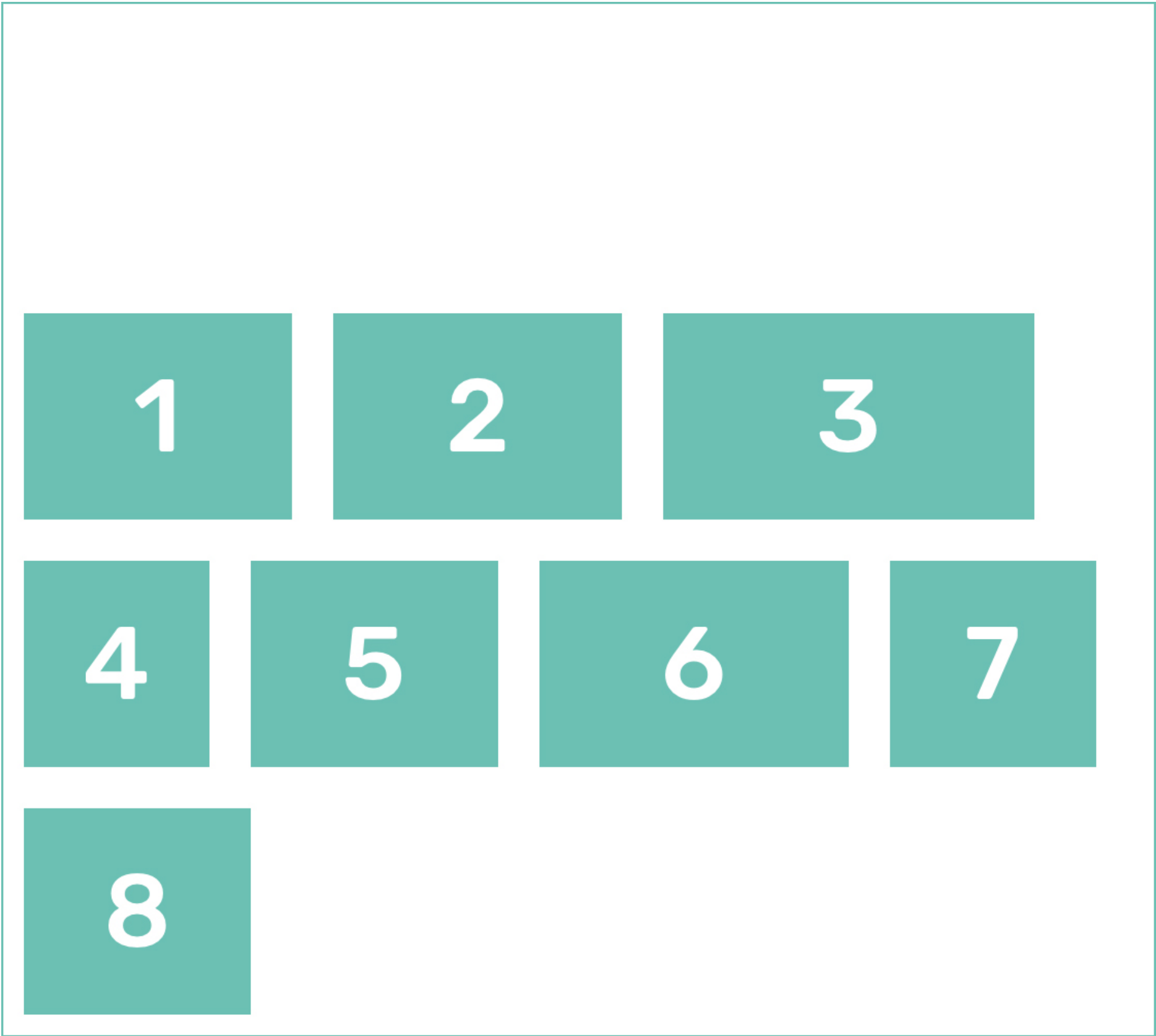
```
/* align-content */
ul{align-content: flex-start | flex-end | center | space-between | space-around | stretch}
```

flex container :: align-content

align-content : flex-start;

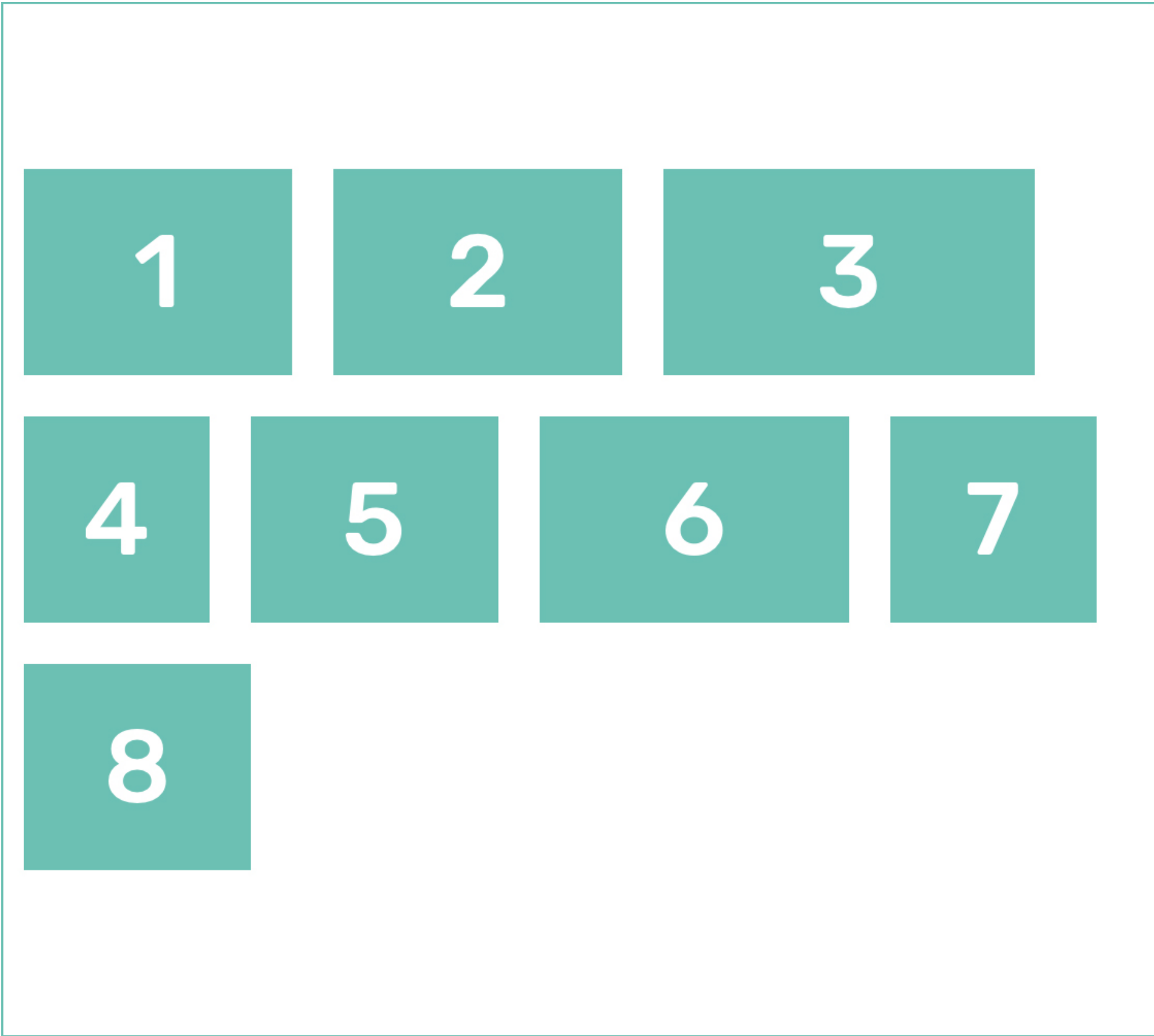


align-content : flex-end;

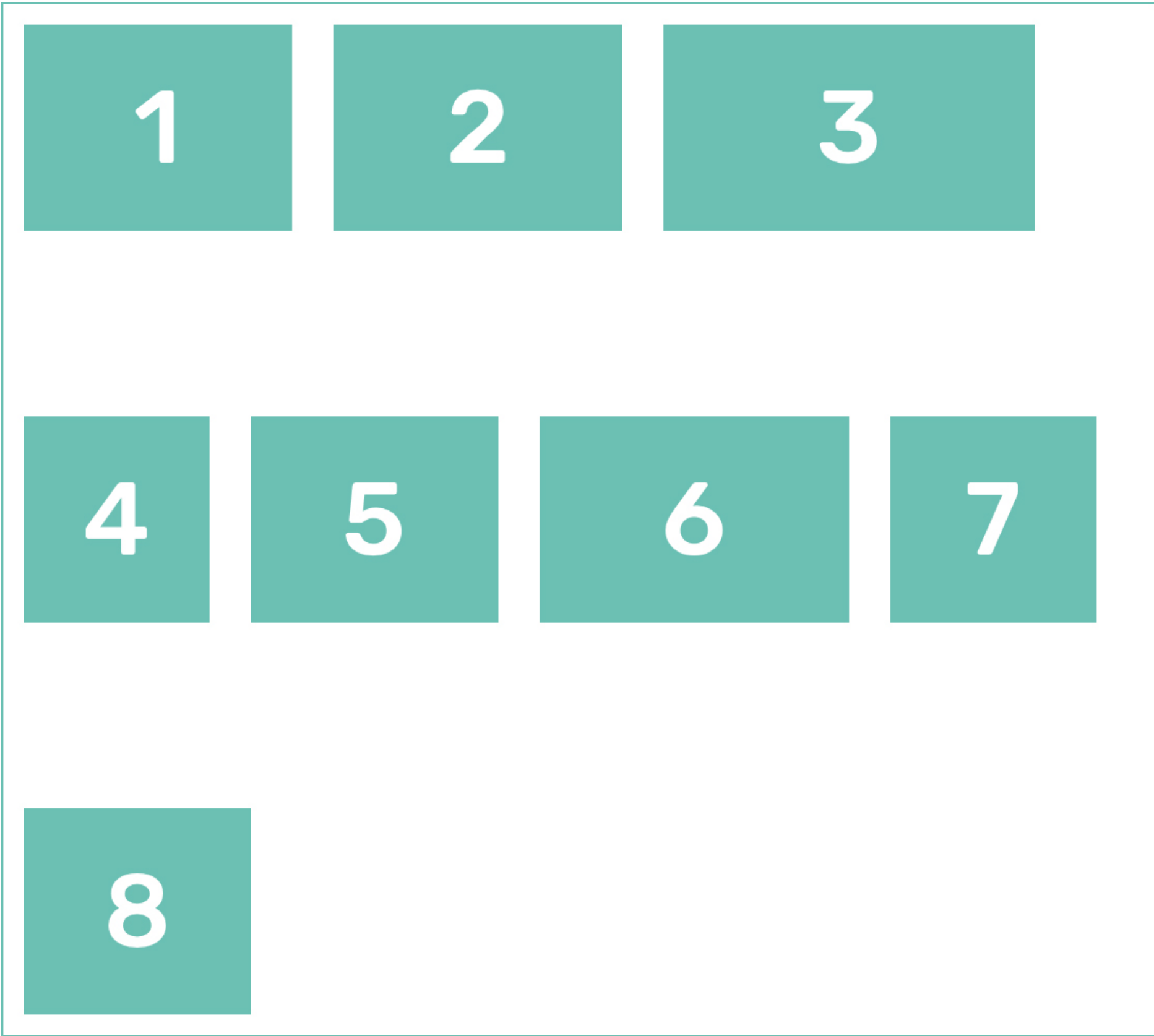


flex container :: align-content

align-content : center;

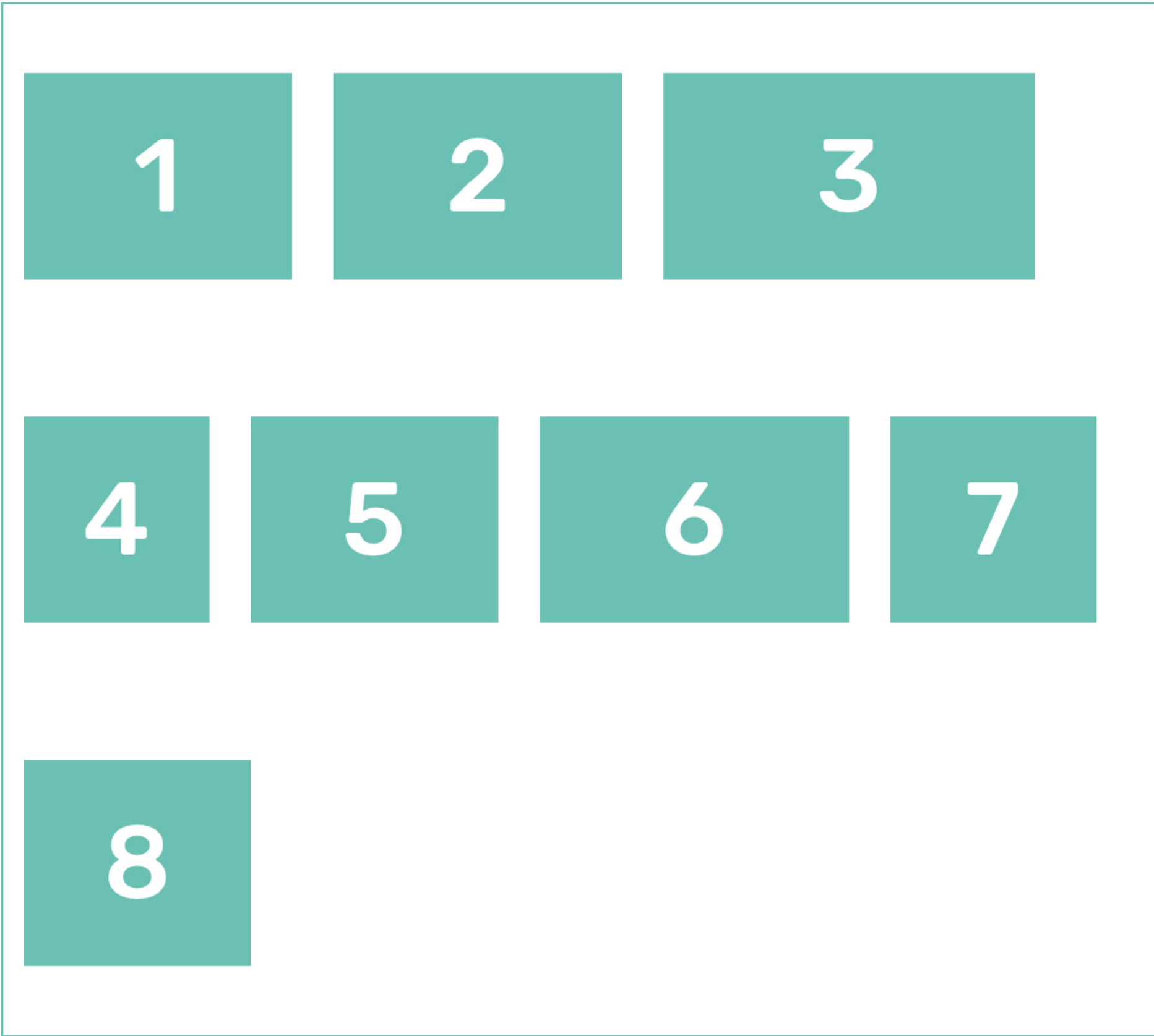


align-content : space-between;

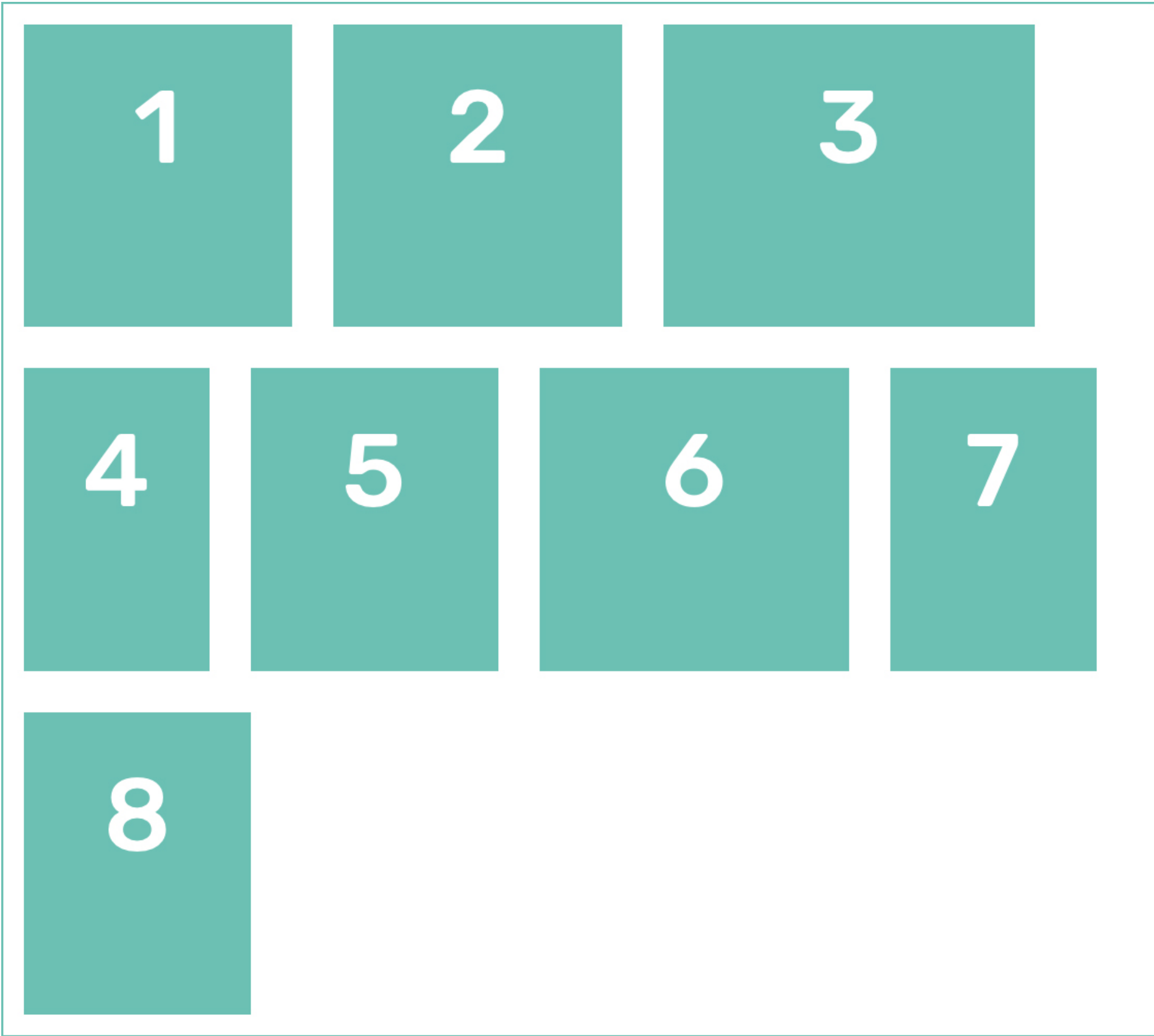


flex container :: align-content

align-content : space-around;



align-content : stretch;



flex items

flex items

order flex-grow flex-shrink flex-basis

flex

align-self

order

flex items :: order

flex-container 안에서 flex-items 이 나열되는 순서를 선언

숫자가 작을수록 시작이며, 높을수록 끝에 위치.

음수 사용이 가능

flex items :: order

```
/* order */
```

```
li{order: <'integer'>}
```

flex items :: order

order : 2;



flex-grow

flex items :: flex-grow

flex-items 이 flex-container 에서 차지할 부피를 확장

기본값은 0. 높을 수록 다른 flex-item 의 곱절 단위로 부피가 확장

flex items :: flex-grow

```
/* flex-grow */  
li{flex-grow: <'number'>}
```

flex items :: flex-grow

flex-grow : 2;



flex-shrink

flex items :: flex-shrink

flex-items 0 이 flex-container 에서 차지할 부피를 축소

기본값은 1. 높을 수록 다른 flex-item 보다 상대적으로 부피가 축소

flex items :: flex-shrink

```
/* flex-shrink */  
li{flex-shrink: <'number'>}
```

flex items :: flex-shrink

flex-shrink : 3;



flex-basis

flex items :: flex-basis

flex-items 의 기본 크기값을 선언

길이(px, em, rem...), 퍼센트(%) 모두 사용 가능

flex items :: flex-basis

```
/* flex-basis */
```

```
li{flex-basis: <'length'>}
```

flex items :: flex-basis

flex-basis : 25%;



flex

flex items :: flex

flex-grow, flex-shrink, flex-basis 의 속성을 단축형으로 사용

flex items :: flex

```
/* flex */
```

```
li{flex: none | [ <'flex-grow'> <'flex-shrink'>? || <'flex-basis'> ]}
```

flex items :: flex

```
li{flex:1 1 100px}
```

align-self

flex items :: align-self

flex-container 에서 선언된 정렬방식 안에서
특정 **flex-item** 에 독자적인 정렬값을 선언

auto flex-start flex-end

center

baseline stretch

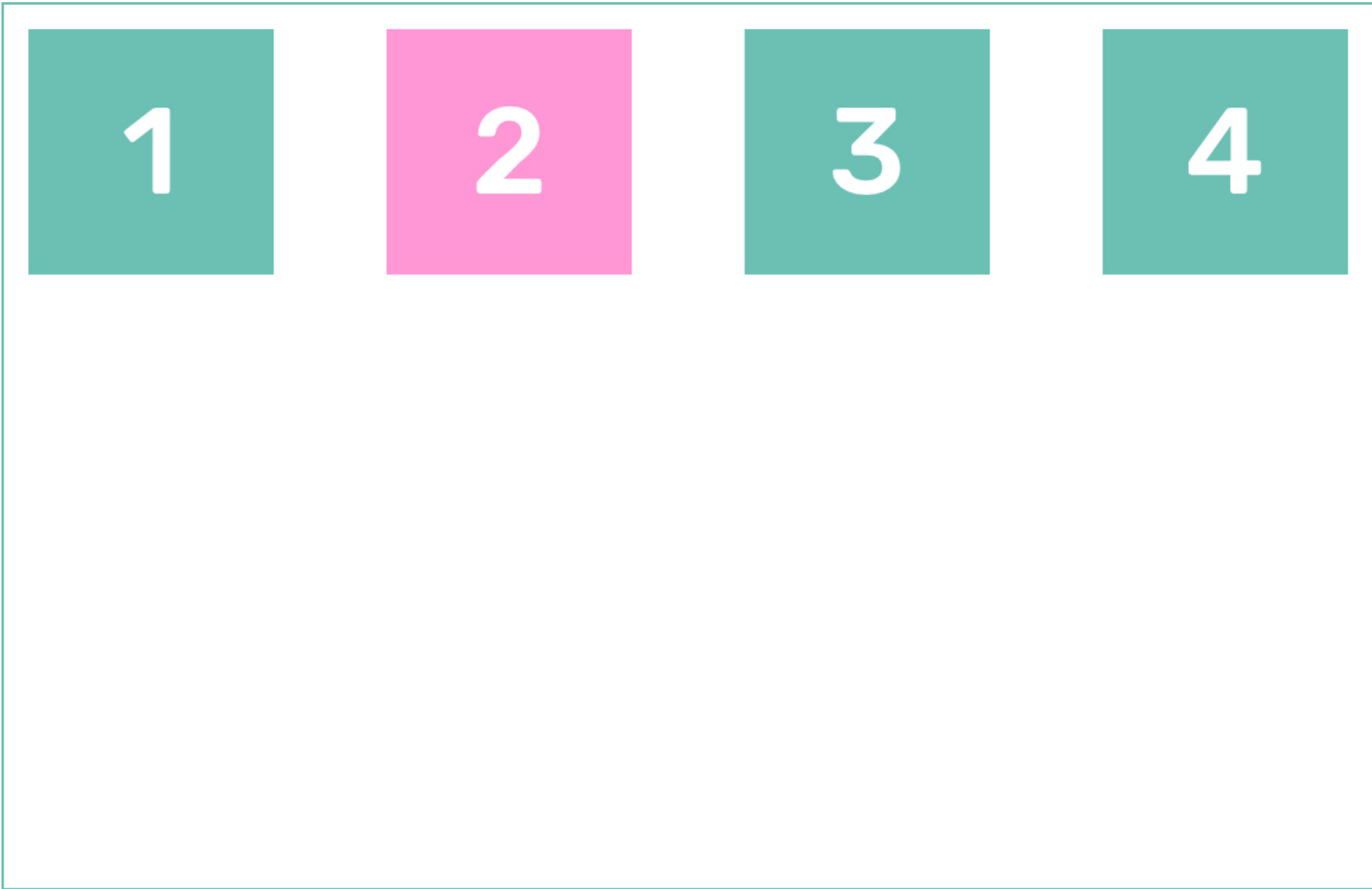
flex items :: align-self

```
/* align-self */
```

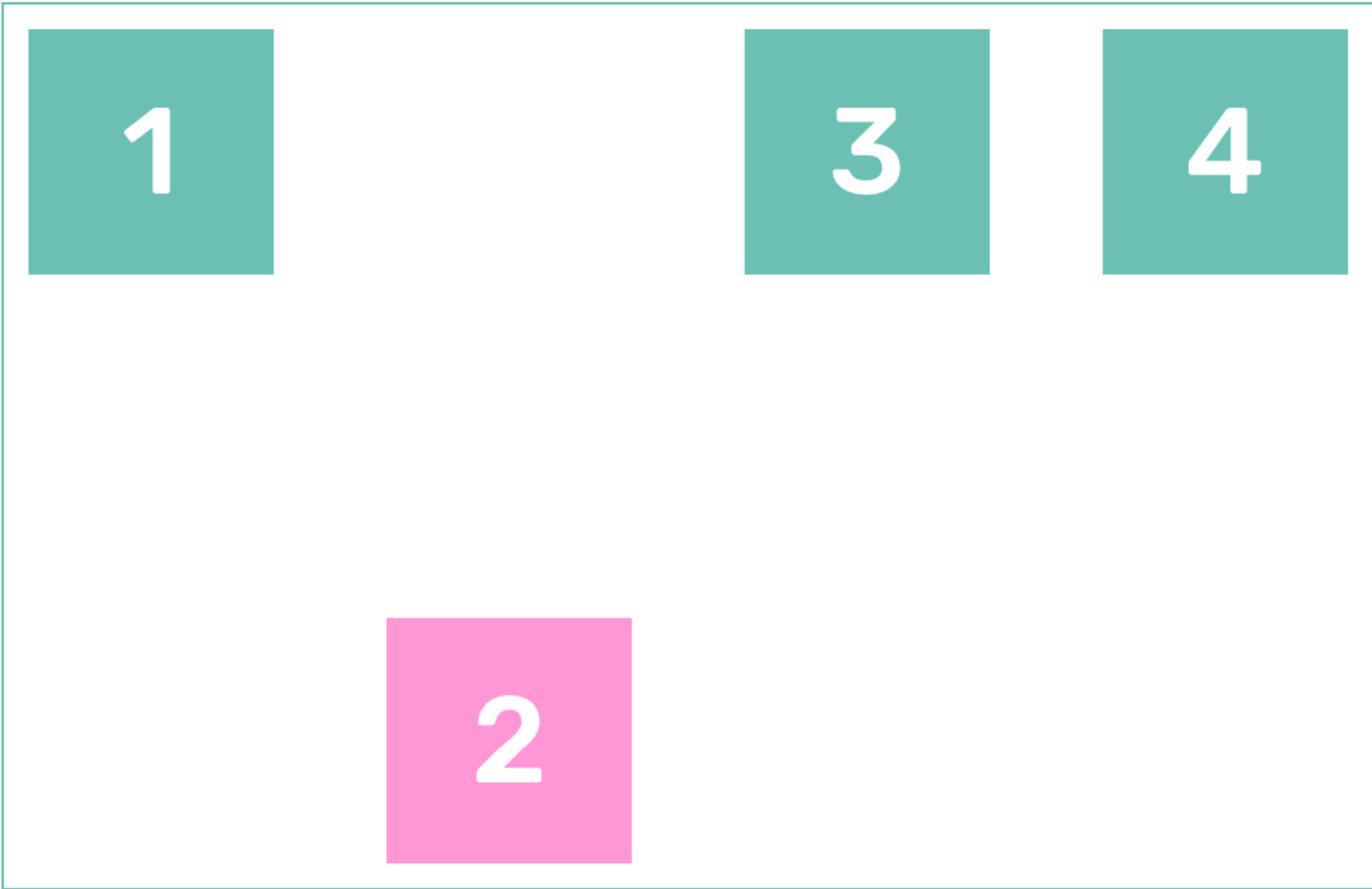
```
li{align-self: auto | flex-start | flex-end | center | baseline | stretch}
```

flex items :: align-self

align-self : flex-start;

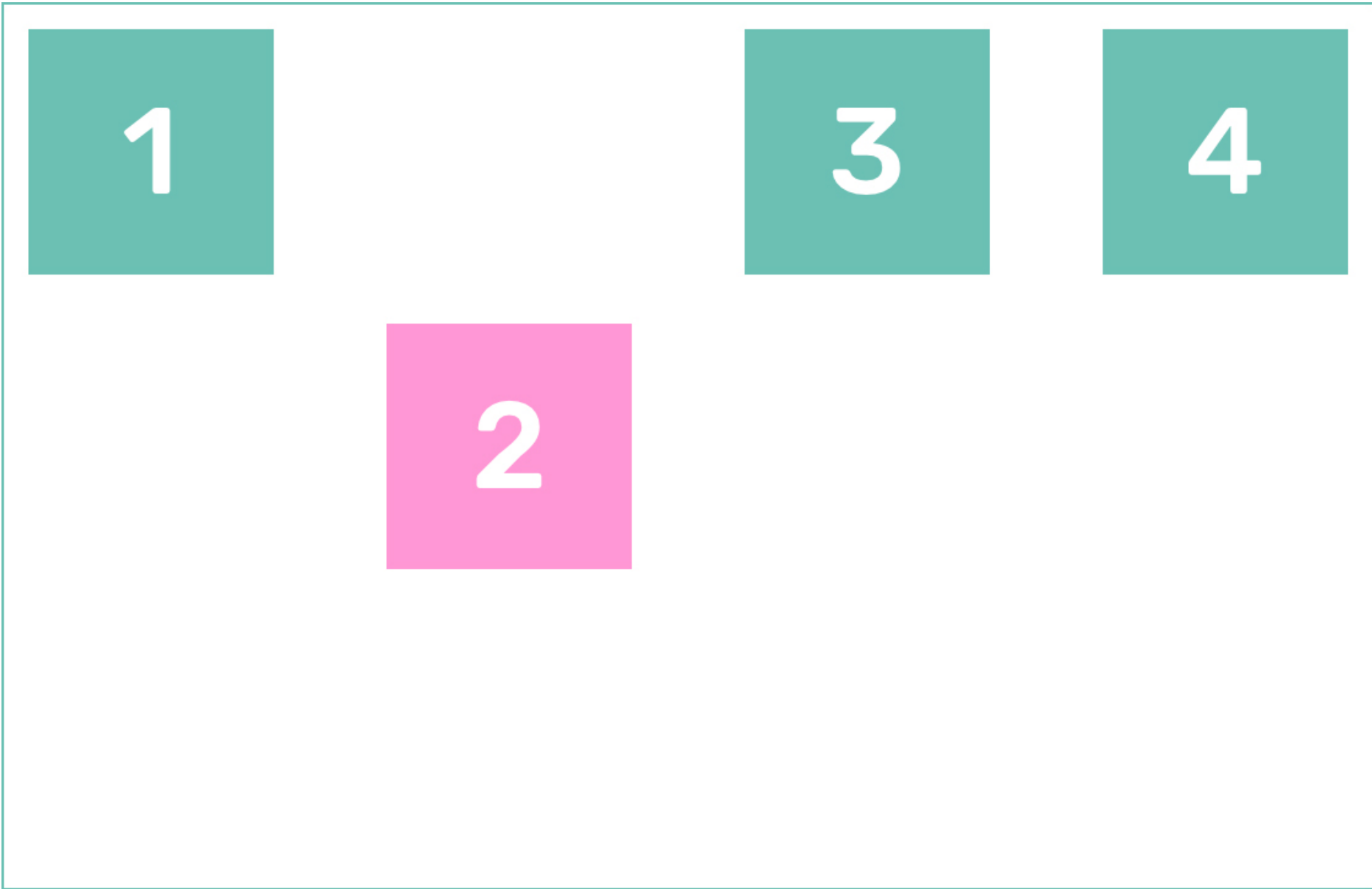


align-self : flex-end;

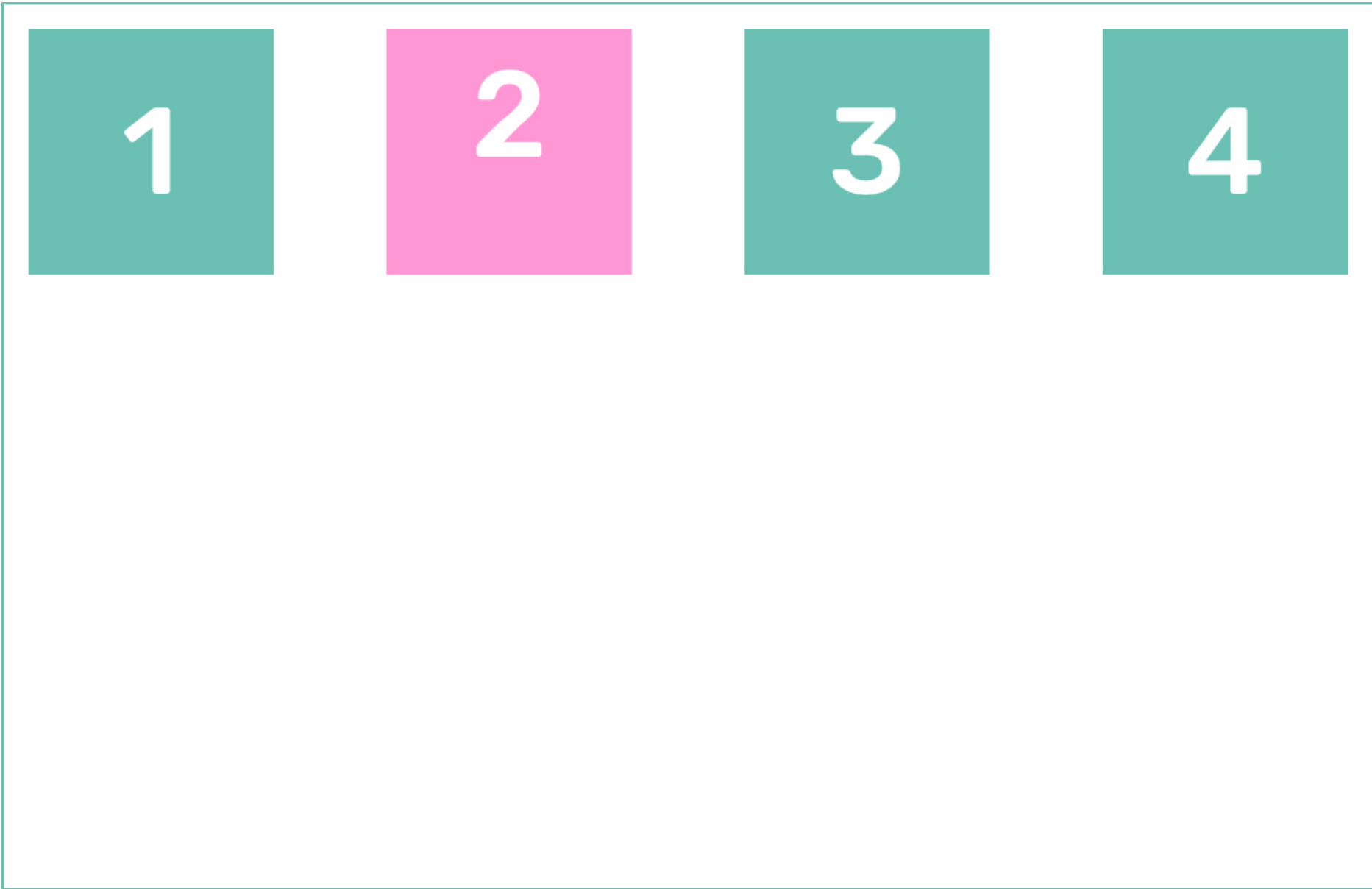


flex items :: align-self

align-self : center;



align-self : baseline;



flex items :: align-self

align-self : stretch;

