Property: `flex-direction`

`flex-direction: row;`

| 1 | 2 | 3 | 4 |

`flex-direction: row-reverse;`

| 4 | 3 | 2 | 1 |

`flex-direction: column;`

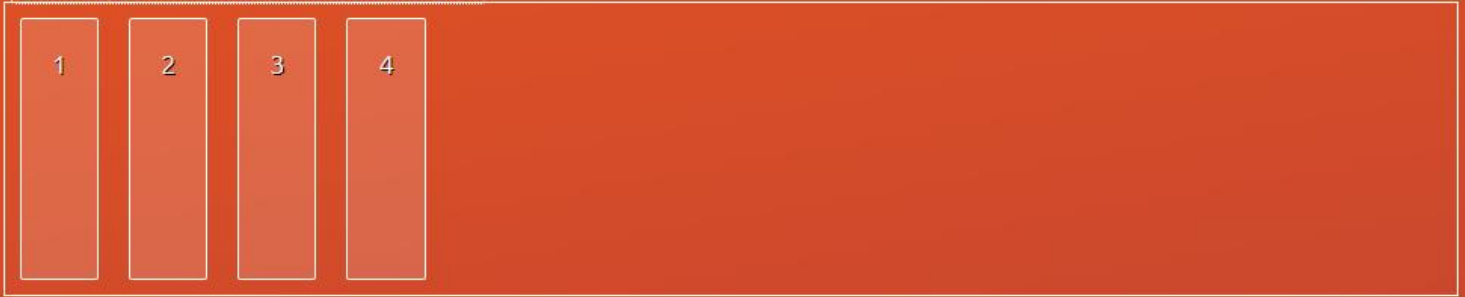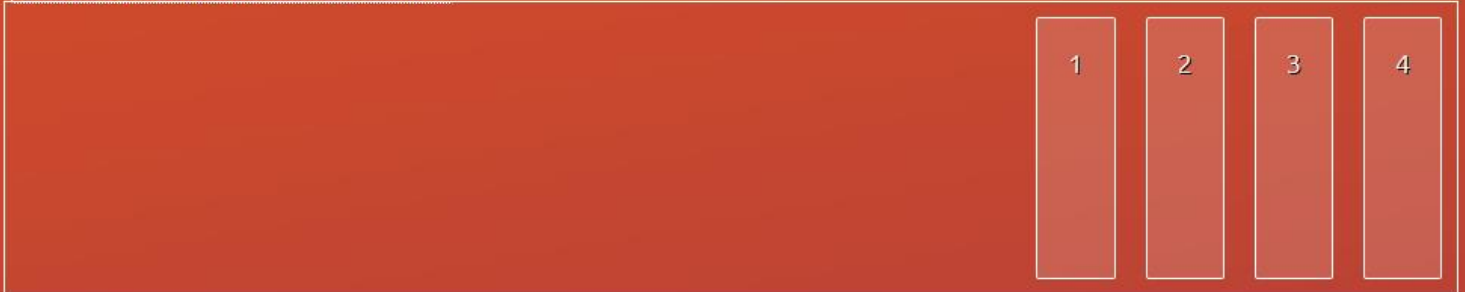1

2

3

4

`flex-direction: column-reverse;`
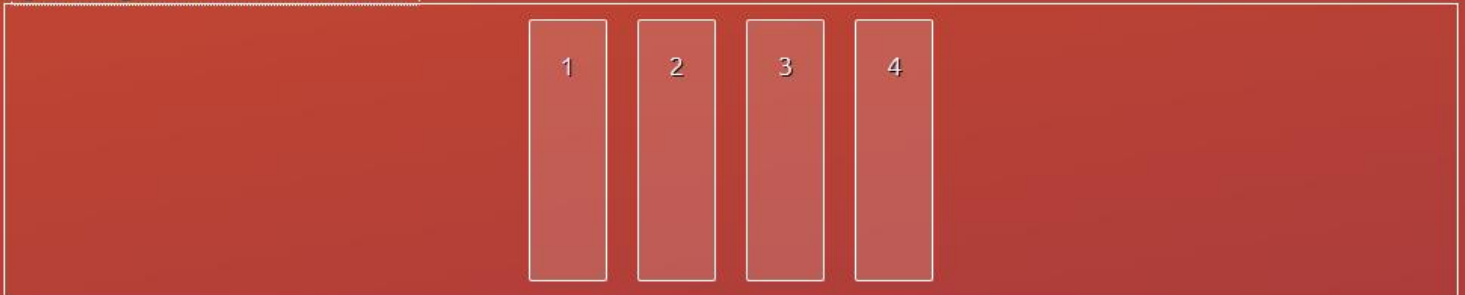
4

3

2

1

Property: `justify-content`

justify-content: flex-start;

| 1 | 2 | 3 | 4 |

justify-content: flex-end;

| 1 | 2 | 3 | 4 |

justify-content: center;

| 1 | 2 | 3 | 4 |

justify-content: space-around;

| 1 | 2 | 3 | 4 |

justify-content: space-between;

| 1 | 2 | 3 | 4 |

Property: `align-items`

`align-items: flex-start;`

| 1 | 2 | 3 | 4 |

`align-items: flex-end;`

| 1 | 2 | 3 | 4 |

`align-items: center;`

| 1 | 2 | 3 | 4 |

`align-items: stretch;`

| 1 | 2 | 3 | 4 |
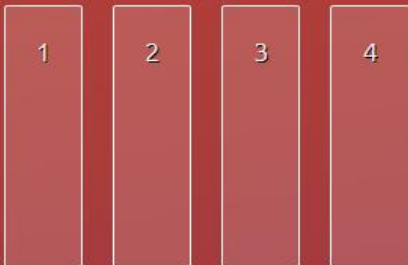
`align-items: baseline;`

| 1 | 2 | 3 | 4 |

Property: `flex-wrap`

flex-wrap: nowrap;

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 1 |

flex-wrap: wrap;

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

| 9 | 10 | 11 | 12 |
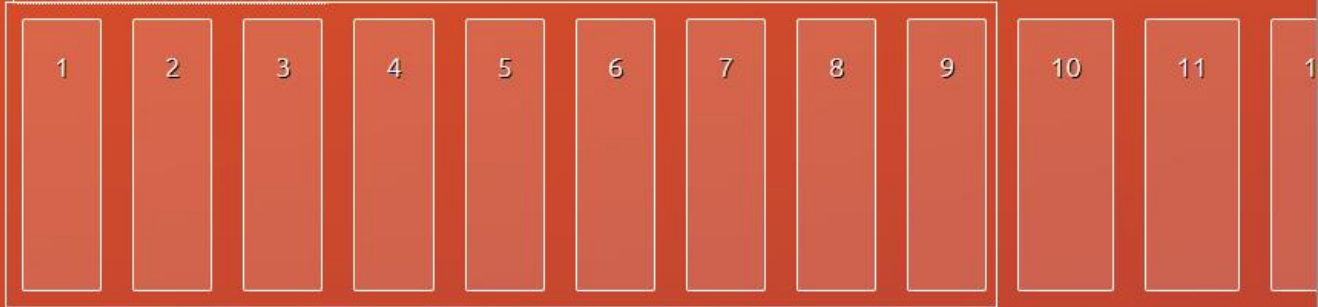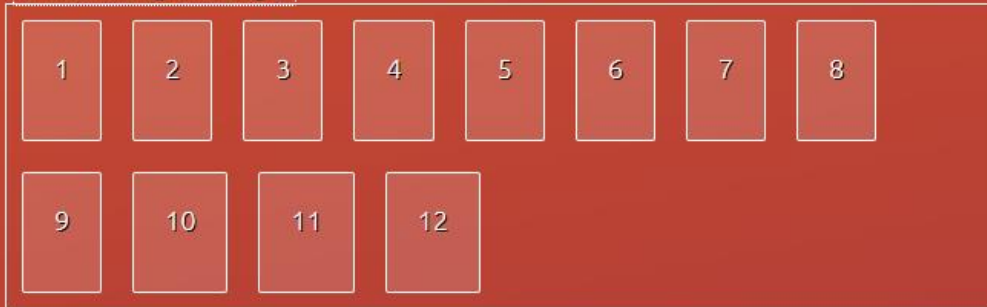
flex-wrap: wrap-reverse;

| 9 | 10 | 11 | 12 |

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
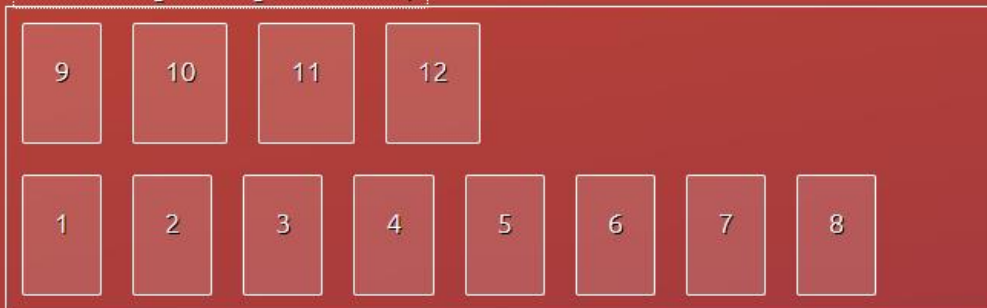
Property: `align-content`

align-content: flex-start;

1	2	3	4	5	6	7	8	9
10	11	12						
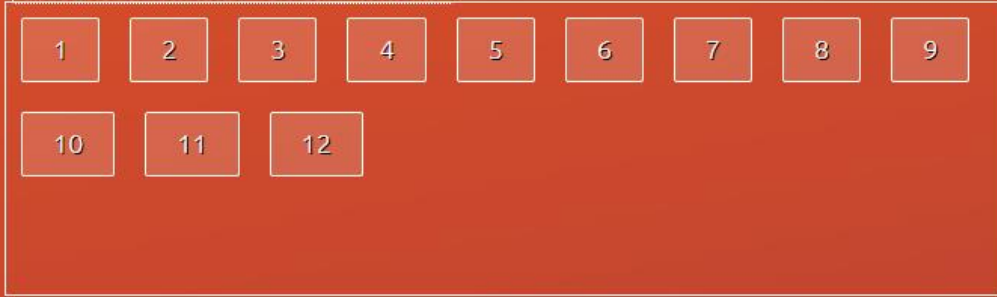
align-content: flex-end;

1	2	3	4	5	6	7	8	9
10	11	12						

align-content: center;

1	2	3	4	5	6	7	8	9
10	11	12						

align-content: space-around;

1	2	3	4	5	6	7	8	9
10	11	12						

align-content: space-between;

1	2	3	4	5	6	7	8	9
10	11	12						
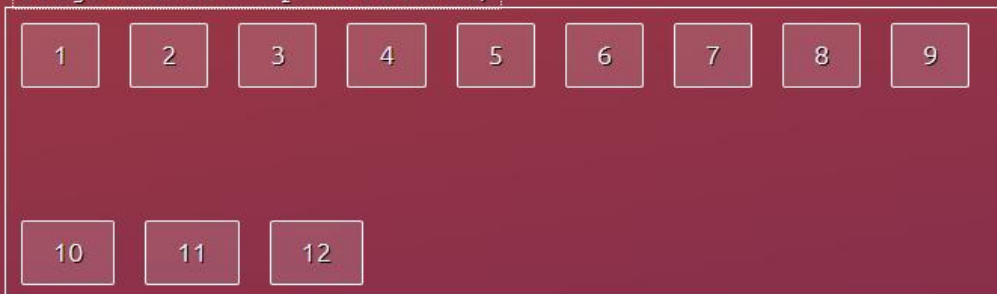
Property: `order`

Default order

1 order: 0	2 order: 0	3 order: 0	4 order: 0

2 and 3 with custom order

2 order: -1	1 order: 0	4 order: 0	3 order: 1

All with custom order

3 order: 1	2 order: 2	4 order: 2	1 order: 3

Property: `align-self`

`align-self: flex-start;`

| 1 | align-self: flex-start | 3 | 4 |

`align-self: flex-end;`

| 1 | | 3 | 4 |
| | align-self: flex-end | | |

`align-self: center;`

| 1 | align-self: center | 3 | 4 |

`align-self: stretch;`

| 1 | align-self: stretch | 3 | 4 |
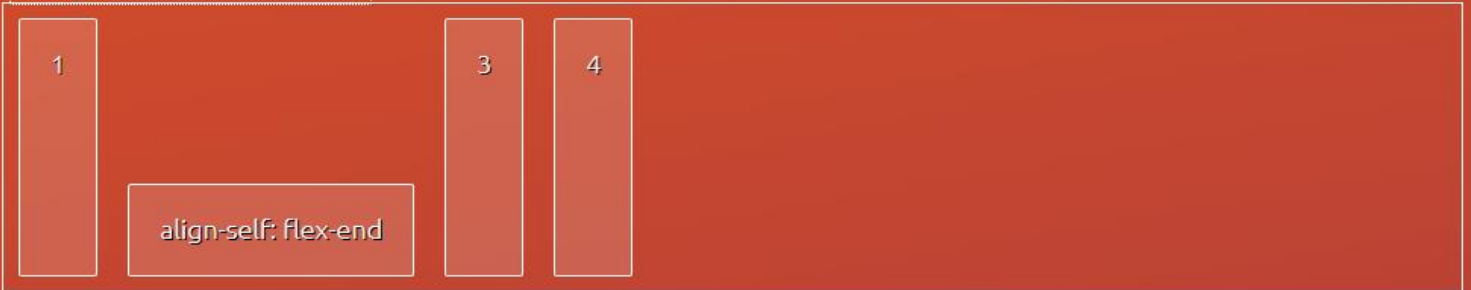
`align-self: baseline;`

| 1 | align-self: baseline | align-self: baseline | 4 | 5 | 6 |
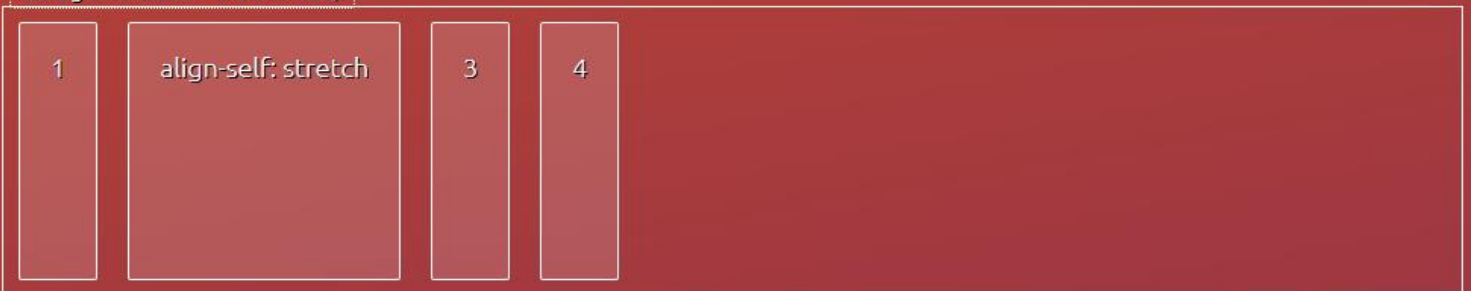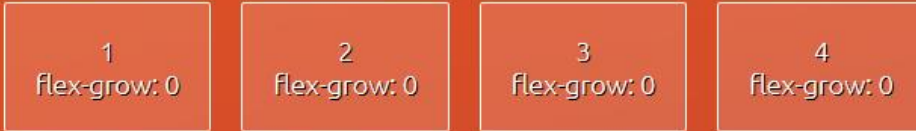
Property: `flex-grow`

Default: `flex-grow: 0` (remaining space not filled)

1 flex-grow: 0	2 flex-grow: 0	3 flex-grow: 0	4 flex-grow: 0

3 allowed to take remaining space (`flex-grow: 1`)

1 flex-grow: 0	2 flex-grow: 0	3 flex-grow: 1	4 flex-grow: 0

Remaining space distributed equally (all have same `flex-grow`)

1 flex-grow: 1	2 flex-grow: 1	3 flex-grow: 1	4 flex-grow: 1

3 gets 2/5 (40%) of remaining space (does not mean that 3 will be twice as wide)

1 flex-grow: 1	2 flex-grow: 1	3 flex-grow: 2	4 flex-grow: 1

3 gets 4/7 of remaining space

1 flex-grow: 1	2 flex-grow: 1	3 flex-grow: 4	4 flex-grow: 1

Property: `flex-shrink`

Default: `flex-shrink: 1` (flex items are allowed to shrink to some degree)

1 flex-shrink: 1	2 flex-shrink: 1	3 flex-shrink: 1	4 flex-shrink: 1	5 flex-shrink: 1	6 flex-shrink: 1

3, 4, and 5 are no longer allowed to shrink (`flex-shrink: 0`)

1 flex-shrink: 1	2 flex-shrink: 1	3 flex-shrink: 0	4 flex-shrink: 0	5 flex-shrink: 0	6 flex-shrink: 1

3 and 4 are allowed to shrink more compared to other flex items

1 flex-shrink: 1	2 flex-shrink: 1	3 flex-shrink: 5	4 flex-shrink: 5	5 flex-shrink: 1	6 flex-shrink: 1

Property: `flex-basis`

Default: `flex-basis: auto`

1 flex-basis: auto	2 flex-basis: auto	3 flex-basis: auto	4 flex-basis: auto

Different `flex-basis` values (act like width on the main axis)

1 flex-basis: 200px	2 flex-basis: 300px	3 flex-basis: auto	4 flex-basis: 100px

Percent values (note that it's necessary to combine with `flex-shrink`)

1 flex-basis: 25%	2 flex-basis: auto	3 flex-basis: auto

4 flex-basis: 50%

Property: `flex`

Default: `flex: 0 1 auto` (grow, shrink, basis)

1 flex: 0 1 auto	2 flex: 0 1 auto	3 flex: 0 1 auto	4 flex: 0 1 auto

`flex: 1` is the same as `flex: 1 1 0%`

1 flex: 1	2 flex: 1	3 flex: 1	4 flex: 1	5 flex: 1	6 flex: 1	7 flex: 1	8 flex: 1

Flex items stick to 25% width (works due to `flex-wrap: nowrap`, thus only for one row)

1 flex: 1 1 25%	2 flex: 1 1 25%	3 flex: 1 1 25%	4 flex: 1 1 25%

Example from `flex-basis` section, now combined with other properties

1 flex: 0 1 25%	2 flex: 1 1 auto	3 flex: 1 1 auto	4 flex: 0 1 50%