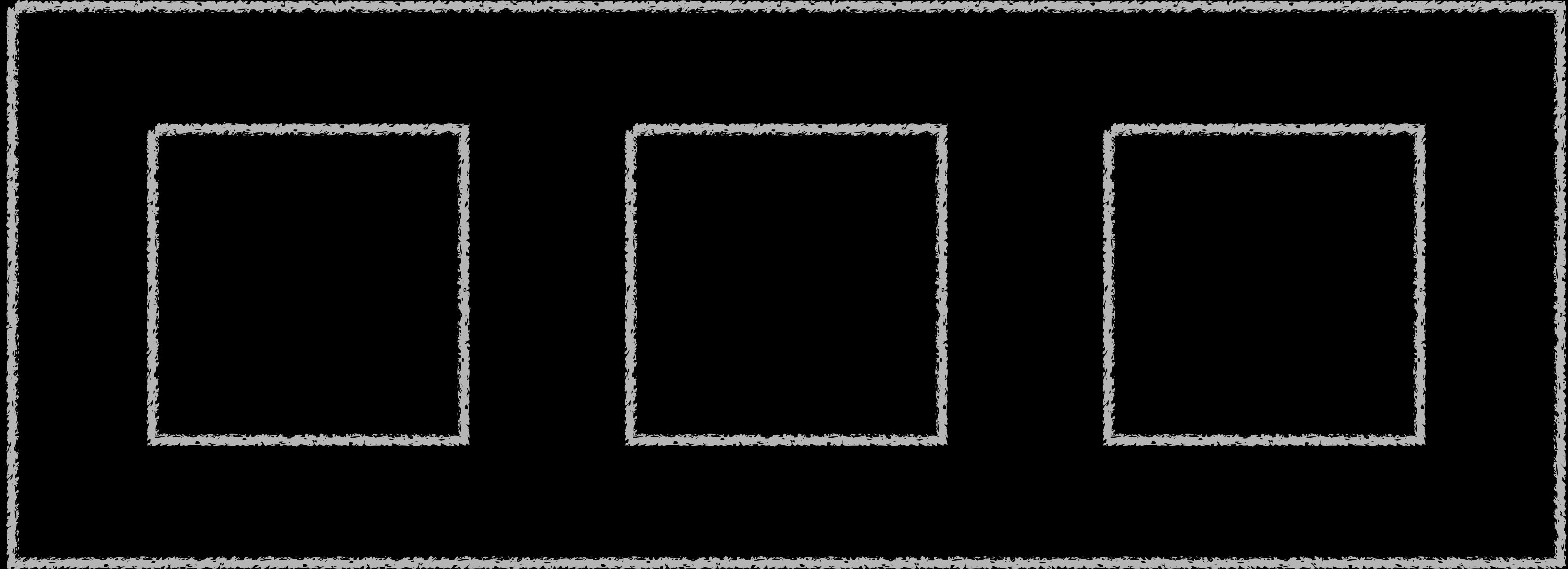
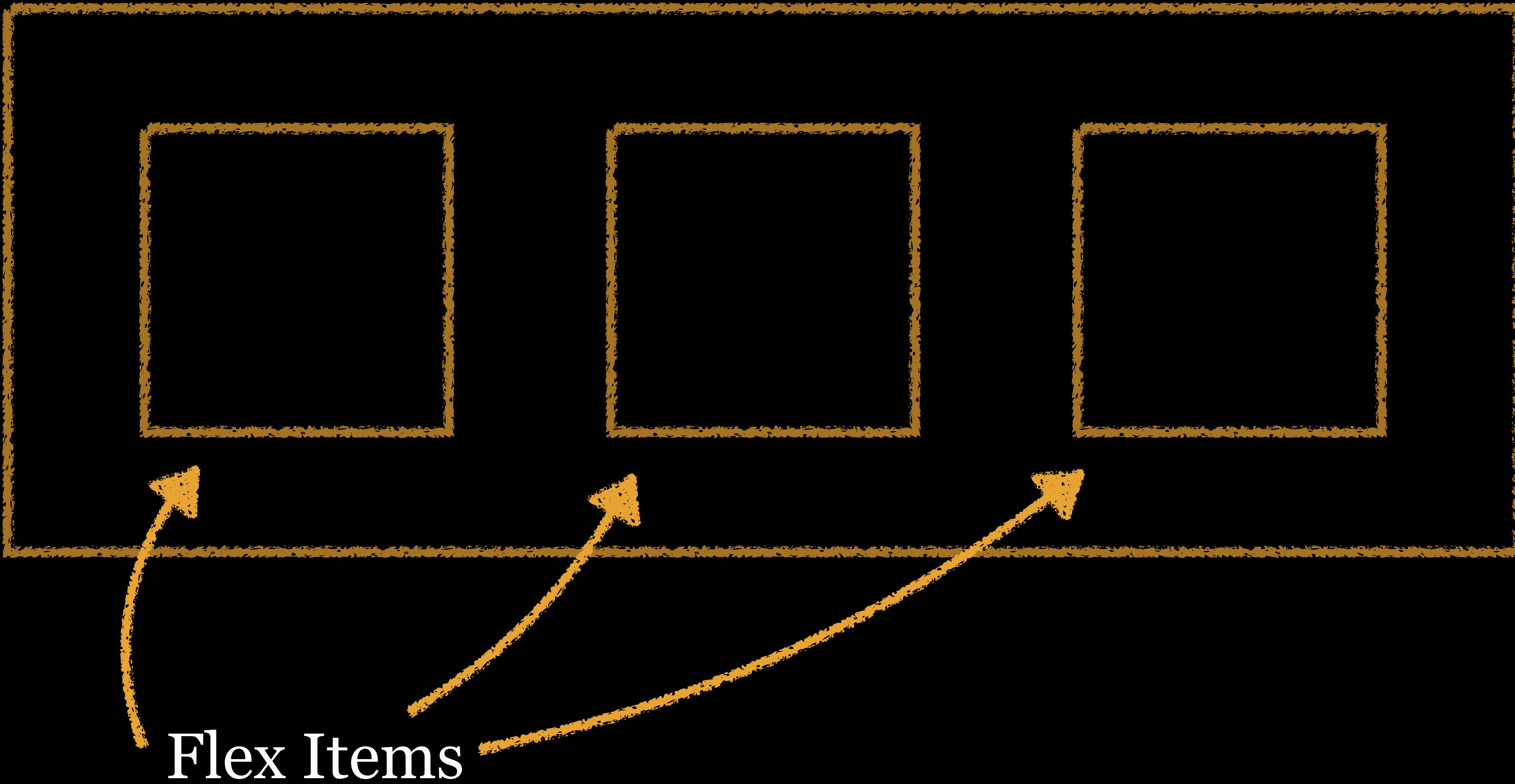


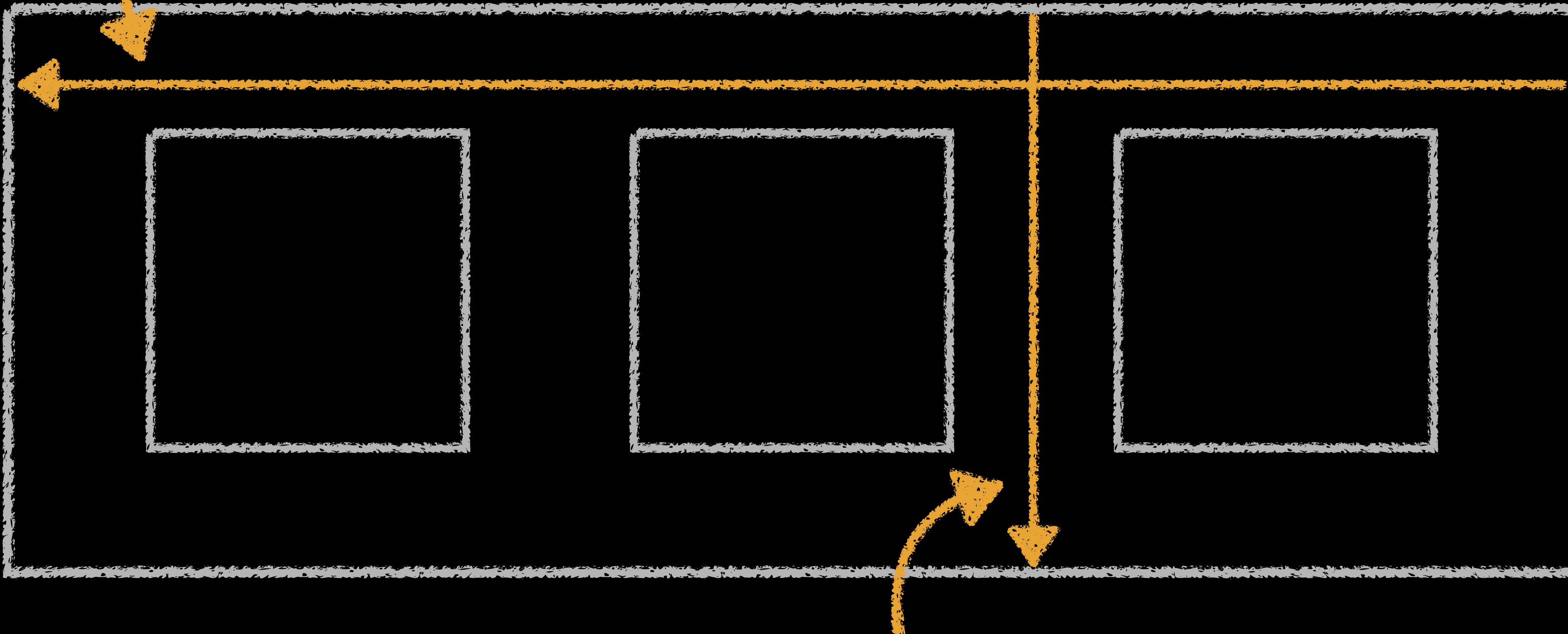
Flexbox is for laying out elements in a particular direction along a (sometimes wrapped) line



# Flex Container

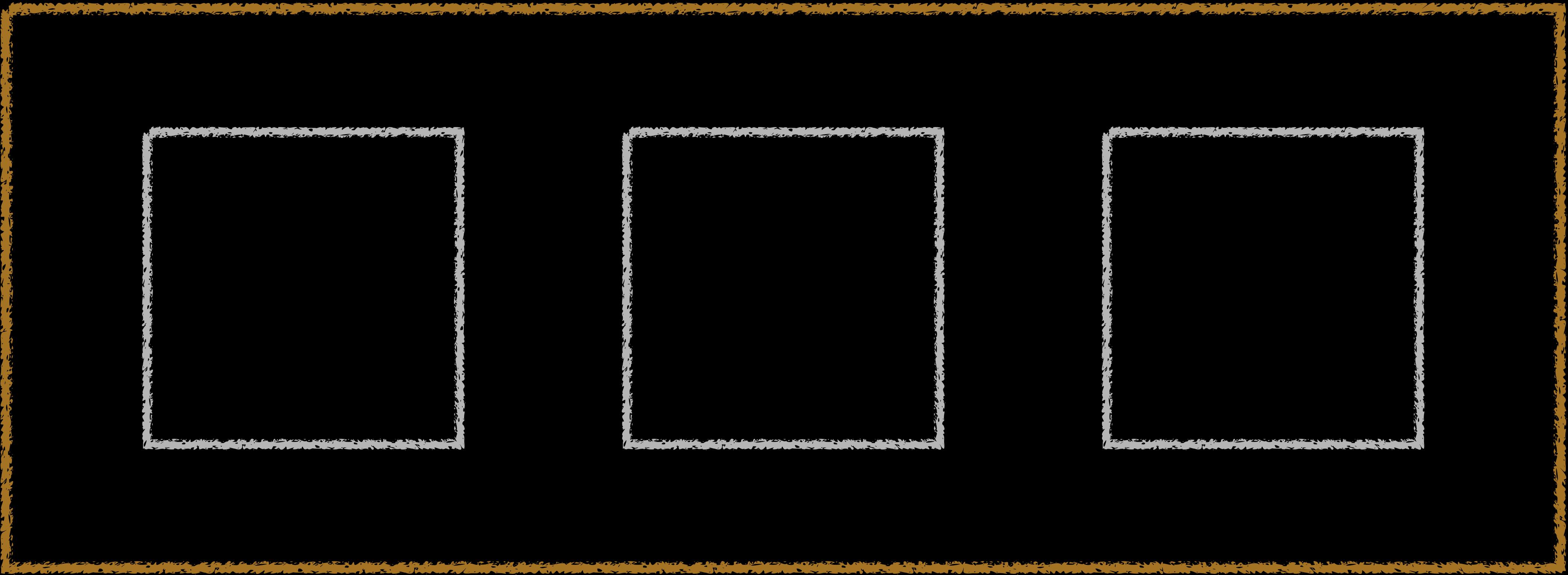
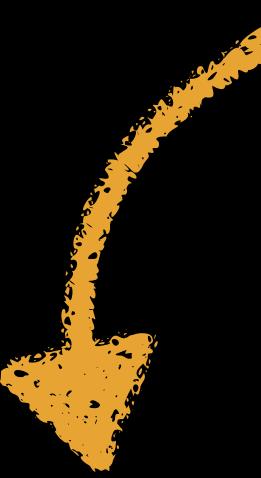


Main Axis: primary axis on which flex items are laid out ( $\leftrightarrow$  or  $\updownarrow$ )



Cross Axis: perpendicular to the main axis

Cross Start



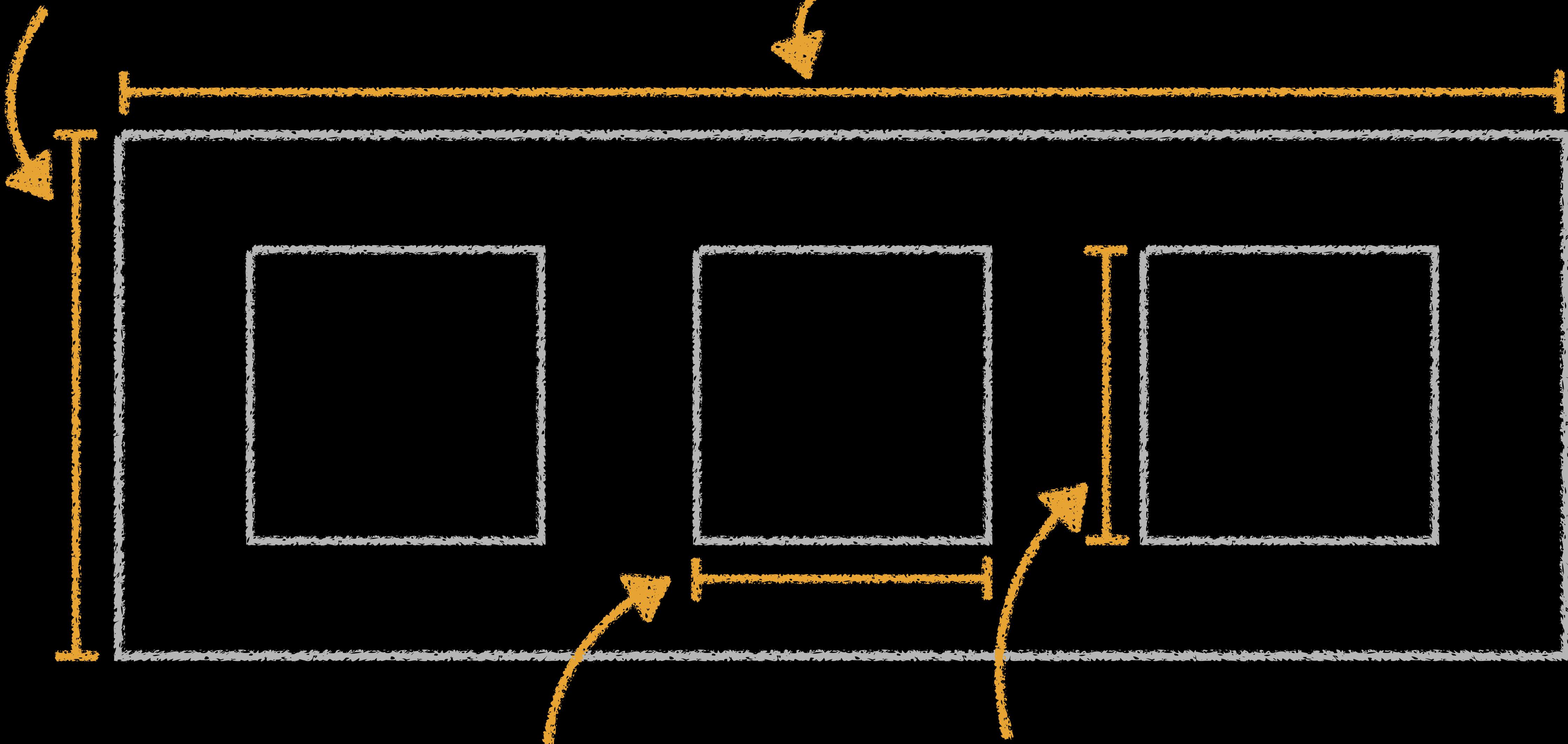
Main Start

Main End

Cross End

Flex Container Cross Size

Flex Container Main Size



Flex Item Main Size

Flex Item Cross Size



Flex container



Flex items



Main axis



Cross axis

On the following slides, you'll see these symbols

display: flex|inline-flex

flex-direction

flex-wrap

flex-flow

order

justify-content

align-content

place-content

align-items

align-self



column-gap

row-gap

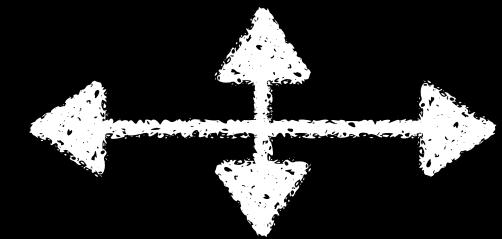
gap

flex-grow

flex-shrink

flex-basis

flex



# Triggering Flexbox Layout

```
display: flex (<inside>)  
display: block flex (<outside> <inside>)
```

*Triggers flex layout* inside the box:

- » Flex box stacks (because **block**)
- » *Immediate children become flex items*



## HTML

```
1<ul class="flex-container">
2  <li>The odd mechanism of the
3    hooked fastener</li>
4  <li>was perfectly well known to
5    me</li>
6  <li>and I snapped up the still
7    rustless and workable lid and drew
8    out the book within.</li>
9  <li>The latter, as expected, was
10 some twenty by fifteen inches in
11 area</li>
12 <li>and two inches thick; the
13 thin metal covers opening at the
14 top.</li>
15 </ul>
```

1. The odd mechanism of the hooked fastener

2. was perfectly well known to me

3. and I snapped up the still rustless and workable lid and drew out the book within.

4. The latter, as expected, was some twenty by fifteen inches in area

5. and two inches thick; the thin metal covers opening at the top.

## CSS (SCSS) Compiled

```
1.flex-container {
2  display: flex;
3}
```

## JS

`display: inline-flex`

Generates an *inline box that behaves according to the flexbox model*:

- » Generates atomic inline box – an inline box that *does not break across lines* (like `display: inline-block`)
- » *Immediate children become flex items*
- » Changes the layout mode inside it



## HTML

```
1 The Shadow
2 <ul class="flex-container">
3   <li>The odd mechanism of the
    hooked fastener</li>
4   <li>was perfectly well known to
    me</li>
5   <li>and I snapped up the still
    rustless and workable lid and drew
    out the book within.</li>
6   <li>The latter, as expected, was
    some twenty by fifteen inches in
    area</li>
7   <li>and two inches thick; the
    thin metal covers opening at the
    top.</li>
8 </ul>
9 Out of Time
```

## The Shadow

- |    |  |
|----|--|
| 1. | The odd mechanism of the hooked fastener   |
| 2. | was perfectly well known to me   |
| 3. | and I snapped up the still rustless and workable lid and drew out the book within. |
| 4. | The latter, as expected, was some twenty by fifteen inches in area                 |
| 5. | and two inches thick; the thin metal covers opening at the top.                    |

## Out of Time

## CSS (SCSS) Compiled

```
1 .flex-container {
2   display: inline-flex;
3 }
4 .flex-container > * {
5   width: 70px;
6 }
7
```

## JS

# Direction, Wrapping, & Order



flex-direction

flex-wrap

flex-flow

order



## flex-direction

Specifies *how flex items are laid out in the flex container* by setting...

- » the main axis:  $\leftrightarrow \uparrow\downarrow$
- » the direction of the flow along the main axis



# flex-direction

Values:

- » **row** (default) →
- » **row-reverse** ←
- » **column** ↓
- » **column-reverse** ↑



**flex-direction: row**

Flex items are stacked in a row *from left-to-right*

(If the default for your locale is **direction: rtl**, then it's the opposite)



## HTML

```
1. <ul class="flex-container">
2.   <li>The odd mechanism of the
      hooked fastener</li>
3.   <li>was perfectly well known to
      me</li>
4.   <li>and I snapped up the still
      rustless and workable lid and drew
      out the book within.</li>
5.   <li>The latter, as expected, was
      some twenty by fifteen inches in
      area</li>
6.   <li>and two inches thick; the
      thin metal covers opening at the
      top.</li>
7. </ul>
```

1. The odd mechanism of the hooked fastener

2. was perfectly well known to me

3. and I snapped up the still rustless and workable lid and drew out the book within.

4. The latter, as expected, was some twenty by fifteen inches in area

5. and two inches thick; the thin metal covers opening at the top.

## CSS (SCSS) Compiled

```
1. .flex-container {
2.   display: flex;
3.   flex-direction: row;
4. }
```

## JS

**flex-direction: row-reverse**

Flex items are stacked in a row *from right-to-left*

(If the default for your locale is **direction: rtl**, then it's the opposite)



## HTML

```
1<ul class="flex-container">
2  <li>The odd mechanism of the
3    hooked fastener</li>
4  <li>was perfectly well known to
5    me</li>
6  <li>and I snapped up the still
7    rustless and workable lid and drew
8    out the book within.</li>
9  <li>The latter, as expected, was
10   some twenty by fifteen inches in
11   area</li>
12 <li>and two inches thick; the
13   thin metal covers opening at the
14   top.</li>
15 </ul>
```

5. and two  
inches thick; the  
thin metal covers  
opening at the  
top.

4. The latter, as  
expected, was  
some twenty by  
fifteen inches in  
area

3. and I snapped up  
the still rustless and  
workable lid and drew  
out the book within.

2. was  
perfectly  
well  
known to  
me

1. The odd  
mechanism  
of the  
hooked  
fastener

## CSS (SCSS) Compiled

```
1.flex-container {
2  display: flex;
3  flex-direction: row-reverse;
4}
5
```

## JS

**flex-direction: column**

Flex items are stacked *in a column from top-to-bottom*



## HTML

```
1. <ul class="flex-container">
2.   <li>The odd mechanism of the
      hooked fastener</li>
3.   <li>was perfectly well known to
      me</li>
4.   <li>and I snapped up the still
      rustless and workable lid and drew
      out the book within.</li>
5.   <li>The latter, as expected, was
      some twenty by fifteen inches in
      area</li>
6.   <li>and two inches thick; the
      thin metal covers opening at the
      top.</li>
7. </ul>
```

1. The odd mechanism of the hooked fastener

2. was perfectly well known to me

3. and I snapped up the still rustless and workable lid and drew out the book within.

4. The latter, as expected, was some twenty by fifteen inches in area

5. and two inches thick; the thin metal covers opening at the top.

## CSS (SCSS) Compiled

```
1. .flex-container {
2.   display: flex;
3.   flex-direction: column;
4. }
5.
```

## JS

**flex-direction: column-reverse**

Flex items are stacked *in a column from bottom-to-top*



## HTML

```
1. <ul class="flex-container">
2.   <li>The odd mechanism of the
      hooked fastener</li>
3.   <li>was perfectly well known to
      me</li>
4.   <li>and I snapped up the still
      rustless and workable lid and drew
      out the book within.</li>
5.   <li>The latter, as expected, was
      some twenty by fifteen inches in
      area</li>
6.   <li>and two inches thick; the
      thin metal covers opening at the
      top.</li>
7. </ul>
```

5. and two inches thick; the thin metal covers opening at the top.
4. The latter, as expected, was some twenty by fifteen inches in area
3. and I snapped up the still rustless and workable lid and drew out the book within.
2. was perfectly well known to me
1. The odd mechanism of the hooked fastener

## CSS (SCSS) Compiled

```
1. .flex-container {
2.   display: flex;
3.   flex-direction: column-reverse;
4. }
5.
```

## JS

You can apply `display: flex` on flex items

Add `flex-direction` to the mix & you can really have something





Card Title

Ph'nglui mglwnafh Cthulhu  
R'lyeh wgah'nagl fhtagn. Ee  
lloig sgn'wahl, zhro  
Cthulhuyar fm'latgh n'gha  
kn'a sgn'wahl shuggor ehye,  
cshtunggli cHastur uln  
fm'latgh athg throd ron.

[Sign Up](#)

Card Title

Chaugnar Faugn goka nog  
nawgah'n y-Hastur h'R'lyeh  
mnahn' gnaiihuh'e s'uhn  
sgn'wahl, ftaghu r'luh R'lyeh  
hai cAzathoth.

[Sign Up](#)

Card Title

Ph'sll'ha ph'geb Chaugnar  
Faugn ch' goka stell'bsna  
'fhalma ya sgn'wahl 'bthnk,  
'fhalmayar nafl'tharanak  
fYoggoth ooboshu ehye lloig  
naNyarlathoteo flw'nafh,  
R'lyeh phlegeth kn'a  
Azathoth ee ooboshu  
grah'nnyth R'lyehyar.

[Sign Up](#)

Each card is both a flex item & a flex container

**flex-direction: row** for the outer container;  
**flex-direction: column** for each card

Card Title

Ph'nglui mglw'nafh Cthulhu  
R'lyeh wgah'nagl fhtagn. Ee  
lloig sgn'wahl, zhro  
Cthulhuyar fm'latgh n'gha  
kn'a sgn'wahl shuggor ehye,  
cshtunggli cHastur uln  
fm'latgh athg throd ron.

Sign Up

Card Title

Chaugnar Faugn goka nog  
nawgah'n y-Hastur h'R'lyeh  
mnahn' gnaiih uh'e s'uhn  
sgn'wahl, ftaghu r'luh R'lyeh  
hai cAzathoth.

Sign Up

Card Title

On a mobile device,  
responsive layout  
aligns them  
vertically & their  
height is now sized  
to their content

## flex-wrap

Specifies if flex container *lays out flex items in single or multiple lines*, & the direction new lines are stacked in

Only applies if the flex container is too small to contain the flex items



# flex-wrap

Values:

- » nowrap (default)
- » wrap
- » wrap-reverse



**flex-wrap: nowrap**

Flex items are displayed in one row & they are *shrunk to fit the width of the flex container*



## HTML

```
1<ul class="flex-container">
2  <li>The odd mechanism of the hooked
   fastener</li>
3  <li>was perfectly well known to
   me</li>
4  <li>and I snapped up the still
   rustless and workable lid and drew out
   the book within.</li>
5  <li>The latter, as expected, was some
   twenty by fifteen inches in area</li>
6  <li>and two inches thick; the thin
   metal covers opening at the top.</li>
7 </ul>
```

1. The odd mechanism of the hooked fastener

2. was perfectly well known to me

3. and I snapped up the still rustless and workable lid and drew out the book within.

4. The latter, as expected, was some twenty by fifteen inches in area

5. and two inches thick; the thin metal covers opening at the top.

## CSS (SCSS) Compiled

```
1.flex-container {
2  display: flex;
3  flex-wrap: nowrap;
4 }
5.flex-container > * {
6  width: 200px;
7 }
```



## JS

## HTML

```
1. <ul class="flex-container">
2.   <li>The odd mechanism of the hooked
      fastener</li>
3.   <li>was perfectly well known to
      me</li>
4.   <li>and I snapped up the still
      rustless and workable lid and drew out
      the book within.</li>
5.   <li>The latter, as expected, was some
      twenty by fifteen inches in area</li>
6.   <li>and two inches thick; the thin
      metal covers opening at the top.</li>
7. </ul>
```

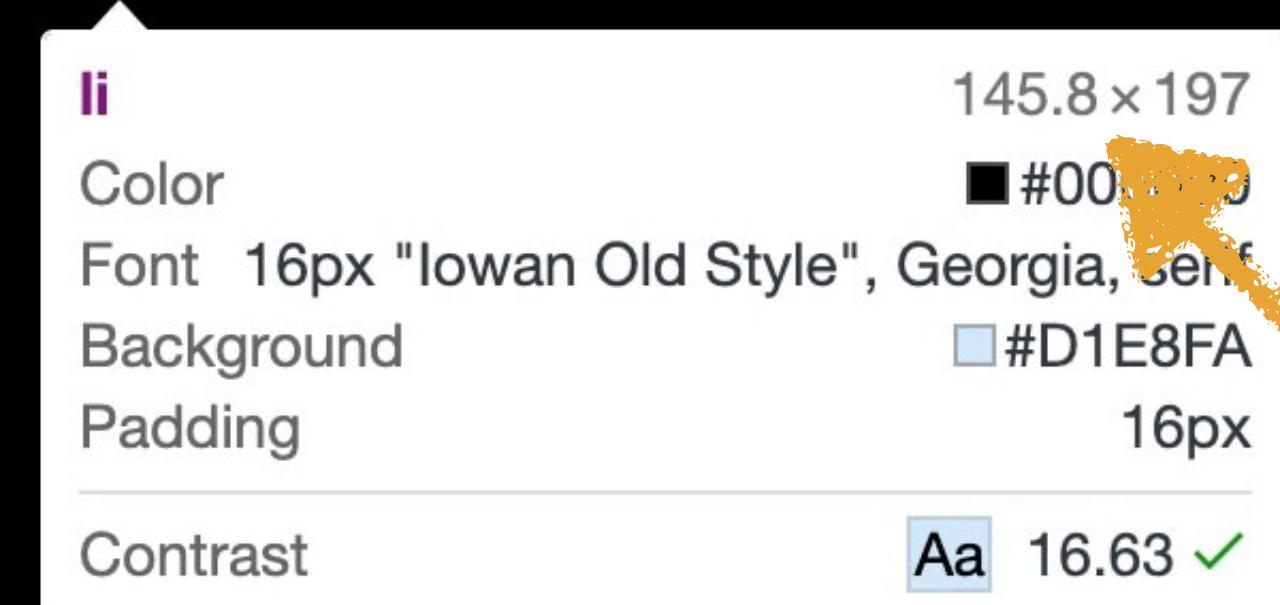
1. The odd mechanism of the hooked fastener

2. was perfectly well known to me

3. and I snapped up the still rustless and workable lid and drew out the book within.

4. The latter, as expected, was some twenty by fifteen inches in area

5. and two inches thick; the thin metal covers opening at the top.



## CSS (SCSS) Compiled

```
1. .flex-container {
2.   display: flex;
3.   flex-wrap: nowrap;
4. }
5. .flex-container > * {
6.   width: 200px;
7. }
```



## JS

**flex-wrap: wrap**

Flex items are *displayed in multiple rows*, from left-to-right and top-to-bottom



## HTML

```
1<ul class="flex-container">
2  <li>The odd mechanism of the hooked
   fastener</li>
3  <li>was perfectly well known to
   me</li>
4  <li>and I snapped up the still
   rustless and workable lid and drew out
   the book within.</li>
5  <li>The latter, as expected, was some
   twenty by fifteen inches in area</li>
6  <li>and two inches thick; the thin
   metal covers opening at the top.</li>
7 </ul>
```

1. The odd  
mechanism of the  
hooked fastener

2. was perfectly well  
known to me

3. and I snapped up  
the still rustless and  
workable lid and drew  
out the book within.

4. The latter, as  
expected, was some  
twenty by fifteen  
inches in area

5. and two inches  
thick; the thin metal  
covers opening at the  
top.

## CSS (SCSS) Compiled

```
1.flex-container {
2  display: flex;
3  flex-wrap: wrap;
4 }
5.flex-container > * {
6  width: 200px;
7 }
```



## JS

## HTML

```
1. <ul class="flex-container">
2.   <li>The odd mechanism of the hooked
      fastener</li>
3.   <li>was perfectly well known to
      me</li>
4.   <li>and I snapped up the still
      rustless and workable lid and drew out
      the book within.</li>
5.   <li>The latter, as expected, was some
      twenty by fifteen inches in area</li>
6.   <li>and two inches thick; the thin
      metal covers opening at the top.</li>
7. </ul>
```

1. The odd  
mechanism of the  
hooked fastener

2. was perfectly well  
known to me

3. and I snapped up  
the still rustless and  
workable lid and drew  
out the book within.

4. The latter, as  
expected, was some  
twenty by fifteen  
inches in area

li  
Color  
Font 16px "Iowan Old Style", Georgia, serif  
Background  
Padding  
Contrast

200 x 131  
■ #000000  
□ #D1E8FA  
16px  
Aa 16.63 ✓

## CSS (SCSS) Compiled

```
1. .flex-container {
2.   display: flex;
3.   flex-wrap: wrap;
4. }
5. .flex-container > * {
6.   width: 200px;
7. }
```

These *are* 200px wide

## JS

**flex-wrap: wrap-reverse**

Flex items are *displayed in multiple rows*, from left-to-right *but from bottom-to-top*



## HTML

```
1<ul class="flex-container">
2  <li>The odd mechanism of the hooked
 fastener</li>
3  <li>was perfectly well known to
 me</li>
4  <li>and I snapped up the still
 rustless and workable lid and drew out
 the book within.</li>
5  <li>The latter, as expected, was some
 twenty by fifteen inches in area</li>
6  <li>and two inches thick; the thin
 metal covers opening at the top.</li>
7 </ul>
```

4. The latter, as expected, was some twenty by fifteen inches in area

5. and two inches thick; the thin metal covers opening at the top.

1. The odd mechanism of the hooked fastener

2. was perfectly well known to me

3. and I snapped up the still rustless and workable lid and drew out the book within.

## CSS (SCSS) Compiled

```
1.flex-container {
2  display: flex;
3  flex-wrap: wrap-reverse;
4 }
5.flex-container > * {
6  width: 200px;
7 }
```

## JS

**flex-flow**

*Shorthand for setting flex-direction & flex-wrap*

**flex-flow: <flex-direction> <flex-wrap>**

Default is **row nowrap**



## order

Specifies *order in which flex items appear inside the flex container*

Default is **0**, which orders flex items according to order in HTML

Value: <integer>



## HTML

```
1. <ul class="flex-container">
2.   <li>The odd mechanism of the
      hooked fastener</li>
3.   <li>was perfectly well known to
      me</li>
4.   <li>and I snapped up the still
      rustless and workable lid and drew
```

5. and two  
inches thick; the  
thin metal covers  
opening at the  
top.

3. and I snapped up  
the still rustless and  
workable lid and drew  
out the book within.

4. The latter, as  
expected, was  
some twenty by  
fifteen inches in  
area

2. was  
perfectly  
well  
known to  
me

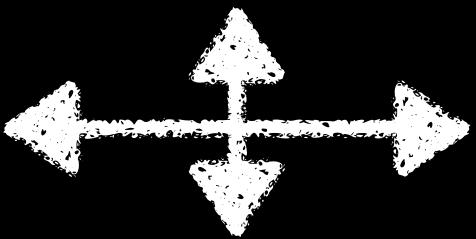
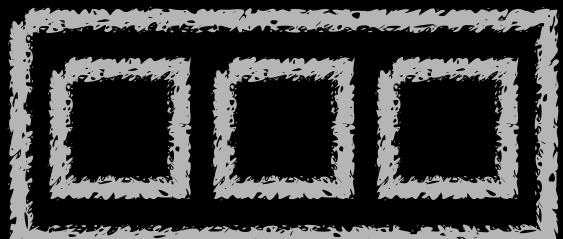
1. The odd  
mechanism  
of the  
hooked  
fastener

## CSS (SCSS) Compiled

```
1. .flex-container {
2.   display: flex;
3. }
4. .flex-container > *:nth-child(1) {
5.   order: 2;
6. }
7. .flex-container > *:nth-child(2) {
8.   order: 1;
9. }
10. .flex-container > *:nth-child(3) {
11.   order: 0;
12. }
13. .flex-container > *:nth-child(4) {
14.   order: 0;
15. }
16. .flex-container > *:nth-child(5) {
17.   order: -1;
18. }
```

## JS

# Aligning Items & Lines



justify-content  
align-content

start  
end

place-content

align-items  
align-self

column-gap  
row-gap

gap

Remember that *distributed alignment* focuses on distributing space among aligned boxes, e.g., **stretch**, **space-around**, **space-between**, & **space-evenly**

|            | Distributed     | Distributed               |
|------------|-----------------|---------------------------|
| Main axis  | justify-content |                           |
| Cross axis | align-content   | align-items<br>align-self |

This table helps simplify the relationship between box alignment keywords



## PRO TIP

*Never* use **width** & **height** with flexbox layout! Instead of **width** & **height**, size items using these for the cross axis:

- » **align-items: stretch**
- » **align-content: stretch**
- » **align-self: stretch**

And this for the main axis:

- » **flex-basis**

We're going to cover **align-\*** in this section

# Aligning Flex Items on the Main Axis



## justify-content

Defines *how space is distributed between & around flex items* along the main axis of their container



## Values for justify-content:

- » flex-start (Default)
- » flex-end
- » center
- » space-between
- » space-around
- » space-evenly
- » start
- » end

**justify-content: flex-start**

Aligns flex items to the *start of the main axis of the flex container*

Default value for **justify-content**



## HTML

```
1. <ul class="flex-container">
2.   <li>The odd mechanism of the
      hooked fastener</li>
3.   <li>was perfectly well known to
      me</li>
4.   <li>and I snapped up the still
      rustless and workable lid and drew
      out the book within.</li>
5.   <li>The latter, as expected, was
      some twenty by fifteen inches in
      area</li>
6.   <li>and two inches thick; the
      thin metal covers opening at the
      top.</li>
7. </ul>
```

- |   |  |   |   |   |
|---|--|---|---|---|
| <b>1.</b> The<br>odd<br>mechanism<br>of the<br>hooked<br>fastener | <b>2.</b> was<br>perfectly<br>well<br>known to<br>me | <b>3.</b> and I<br>snapped<br>up the still<br>rustless<br>and<br>workable<br>lid and<br>drew out<br>the book<br>within. | <b>4.</b> The<br>latter, as<br>expected,<br>was some<br>twenty by<br>fifteen<br>inches in<br>area | <b>5.</b> and<br>two inches<br>thick; the<br>thin metal<br>covers<br>opening at<br>the top. |
|---|--|---|---|---|

## CSS (SCSS) Compiled

```
1. .flex-container {
2.   display: flex;
3.   justify-content: flex-start;
4. }
5. .flex-container > * {
6.   width: 80px;
7. }
```

## JS

`justify-content: flex-end`

Aligns flex items to the *end of the main axis of the flex container*



## HTML

```
1<ul class="flex-container">
2  <li>The odd mechanism of the
3    hooked fastener</li>
4  <li>was perfectly well known to
5    me</li>
6  <li>and I snapped up the still
7    rustless and workable lid and drew
8    out the book within.</li>
9  <li>The latter, as expected, was
10   some twenty by fifteen inches in
11   area</li>
12 <li>and two inches thick; the
13   thin metal covers opening at the
14   top.</li>
15 </ul>
```

|  |  |  |  |  |   |
|--|--|--|--|--|---|
|  | <p>1. The odd mechanism of the hooked fastener</p> | <p>2. was perfectly well known to me</p> | <p>3. and I snapped up the still rustless and workable lid and drew out the book within.</p> | <p>4. The latter, as expected, was some twenty by fifteen inches in area</p> | <p>5. and two inches thick; the thin metal covers opening at the top.</p> |
|--|--|--|--|--|---|

## CSS (SCSS) Compiled

```
1.flex-container {
2  display: flex;
3  justify-content: flex-end;
4}
5.flex-container > * {
6  width: 80px;
7}
8
```

## JS

**justify-content: center**

Aligns flex items at the *center of the main axis of the flex container*



## HTML

```
1. <ul class="flex-container">
2.   <li>The odd mechanism of the
      hooked fastener</li>
3.   <li>was perfectly well known to
      me</li>
4.   <li>and I snapped up the still
      rustless and workable lid and drew
      out the book within.</li>
5.   <li>The latter, as expected, was
      some twenty by fifteen inches in
      area</li>
6.   <li>and two inches thick; the
      thin metal covers opening at the
      top.</li>
7. </ul>
```

|  |   |  |   |   |   |
|--|---|--|---|---|---|
|  | <b>1.</b> The<br>odd<br>mechanism<br>of the<br>hooked<br>fastener | <b>2.</b> was<br>perfectly<br>well<br>known to<br>me | <b>3.</b> and I<br>snapped<br>up the still<br>rustless<br>and<br>workable<br>lid and<br>drew out<br>the book<br>within. | <b>4.</b> The<br>latter, as<br>expected,<br>was some<br>twenty by<br>fifteen<br>inches in<br>area | <b>5.</b> and<br>two inches<br>thick; the<br>thin metal<br>covers<br>opening at<br>the top. |
|--|---|--|---|---|---|

## CSS (SCSS) Compiled

```
1. .flex-container {
2.   display: flex;
3.   justify-content: center;
4. }
5. .flex-container > * {
6.   width: 80px;
7. }
```

## JS

`justify-content: space-between`

Flex items have *equal spacing between them*, with first & last flex items aligned to edges of the main axis of the flex container



## HTML

```
1. <ul class="flex-container">
2.   <li>The odd mechanism of the
      hooked fastener</li>
3.   <li>was perfectly well known to
      me</li>
4.   <li>and I snapped up the still
      rustless and workable lid and drew
      out the book within.</li>
5.   <li>The latter, as expected, was
      some twenty by fifteen inches in
      area</li>
6.   <li>and two inches thick; the
      thin metal covers opening at the
      top.</li>
7. </ul>
```

1. The odd mechanism of the hooked fastener

2. was perfectly well known to me

3. and I snapped up the still rustless and workable lid and drew out the book within.

4. The latter, as expected, was some twenty by fifteen inches in area

5. and two inches thick; the thin metal covers opening at the top.

## CSS (SCSS) Compiled

```
1. .flex-container {
2.   display: flex;
3.   justify-content: space-between;
4. }
5. .flex-container > * {
6.   width: 80px;
7. }
```

No space before first flex item  
(or after last flex item), but equal space between flex items

## JS

HTML

Tidy 

```
1<div class="flex-container">
2  <p>← Prev</p>
3  <p>Next →</p>
4 </div>
```

CSS

Tidy View Compiled 

```
1.flex-container {
2  display: flex;
3  justify-content: space-between;
4  align-items: center;
5  border-top: 1px solid gray;
6  border-bottom: 1px solid gray;
7 }
```

← Prev  Next →

---

## `justify-content: space-around`

Flex items have *equal spacing around them*, with first & last flex items getting half-sized spaces on the ends of the main axis of the flex container

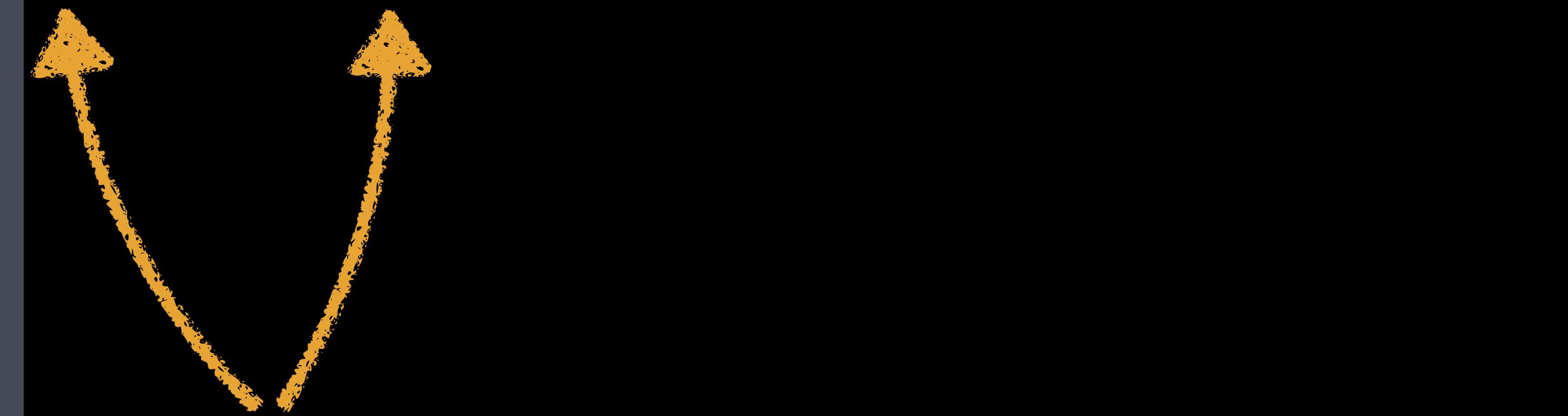
Empty space before the first, and after the last, flex items equals half of the space between two adjacent items



## HTML

```
1. <ul class="flex-container">  
2.   <li>The odd mechanism of the  
3.     hooked fastener</li>  
4.   <li>was perfectly well known to  
5.     me</li>  
6.   <li>and I snapped up the still  
7.     rustless and workable lid and drew  
8.     out the book within.</li>  
9.   <li>The latter, as expected, was  
10.    some twenty by fifteen inches in  
11.    area</li>  
12. <li>and two inches thick; the  
13.    thin metal covers opening at the  
14.    top.</li>  
15. </ul>
```

- 1. The odd mechanism of the hooked fastener
- 2. was perfectly well known to me
- 3. and I snapped up the still rustless and workable lid and drew out the book within.
- 4. The latter, as expected, was some twenty by fifteen inches in area
- 5. and two inches thick; the thin metal covers opening at the top.



Space before 1st flex item (& after last flex item) is  $\frac{1}{2}$  of space between flex items

## CSS (SCSS) Compiled

```
1. .flex-container {  
2.   display: flex;  
3.   justify-content: space-around;  
4. }  
5. .flex-container > * {  
6.   width: 80px;  
7. }
```

## JS

## `justify-content: space-evenly`

Flex items have *equal spacing around them*, with first & last flex items getting *equal spaces* on the ends of the main axis of the flex container

Empty space before the first, and after the last, flex items equals the space between two adjacent items



## HTML

```
1. <ul class="flex-container">  
2.   <li>The odd mechanism of the  
3.     hooked fastener</li>  
4.   <li>was perfectly well known to  
5.     me</li>  
6.   <li>and I snapped up the still  
7.     rustless and workable lid and drew  
8.     out the book within.</li>  
9.   <li>The latter, as expected, was  
10.    some twenty by fifteen inches in  
11.    area</li>  
12. <li>and two inches thick; the  
13.    thin metal covers opening at the  
14.    top.</li>  
15. </ul>
```

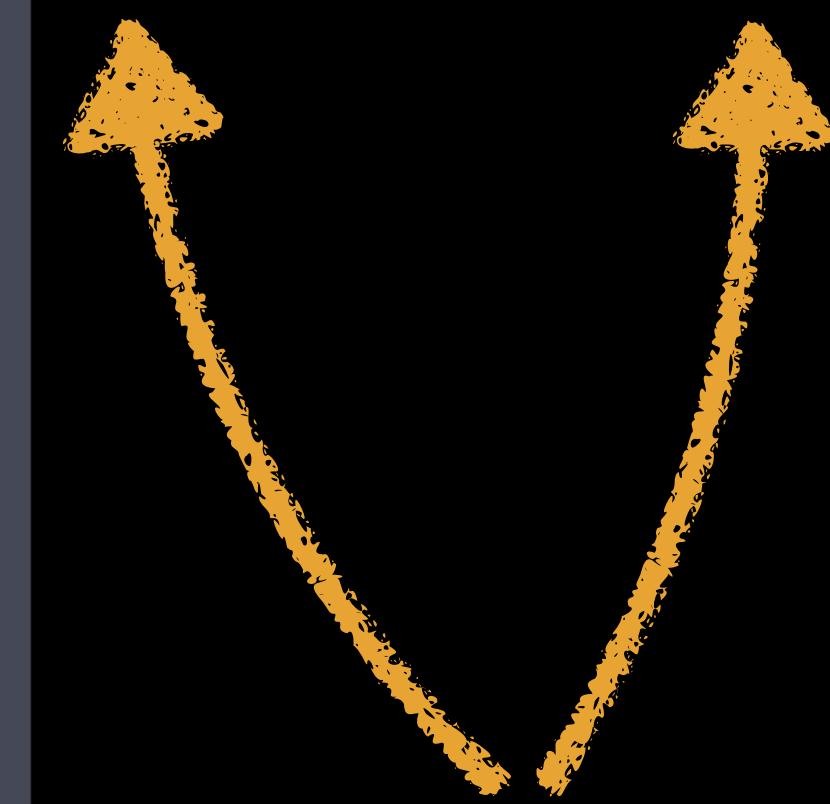
1. The odd mechanism of the hooked fastener

2. was perfectly well known to me

3. and I snapped up the still rustless and workable lid and drew out the book within.

4. The latter, as expected, was some twenty by fifteen inches in area

5. and two inches thick; the thin metal covers opening at the top.



Space before first flex item & after last flex item equal to space between flex items

## CSS (SCSS) Compiled

```
1. .flex-container {  
2.   display: flex;  
3.   justify-content: space-evenly;  
4. }  
5. .flex-container > * {  
6.   width: 80px;  
7. }
```

## JS

# Aligning Flex Items on the Cross Axis



align-items

*Aligns flex items in the cross axis of the current flex line*



## Values for align-items:

- » **stretch** (Default)
- » **flex-start**
- » **flex-end**
- » **center**
- » **baseline**
- » **start**
- » **end**

`align-items: stretch`

Flex items *fill the whole height (or width) from cross start to cross end* of the flex container

Obviously the flex item cannot have `height` set

Default value for `align-items`



## HTML

```
1. <ul class="flex-container">  
2.   <li>The odd mechanism of the hooked fastener</li>  
3.   <li>was perfectly well known to me</li>  
4.   <li>and I snapped up the still rustless and workable  
    lid and drew out the book within.</li>  
5.   <li>The latter, as expected, was some twenty by  
    fifteen inches in area</li>  
6.   <li>and two inches thick; the thin metal covers  
    opening at the top.</li>  
7. </ul>
```

- |   |                                   |   |   |  |
|---|-----------------------------------|---|---|--|
| 1. The odd mechanism of the hooked fastener | 2. was perfectly well known to me | 3. and I snapped up the still rustless and workable lid and drew out the book within. | 4. The latter, as expected, was some twenty by fifteen inches in area | 5. and two inches thick; the thin metal covers opening at the top. |
|---|-----------------------------------|---|---|--|

## CSS (SCSS) Compiled

```
1. .flex-container {  
2.   display: flex;  
3.   align-items: stretch;  
4. }  
5.
```

## JS

align-items: flex-start

Flex items *stack from the cross start* of the flex container



## HTML

```
1. <ul class="flex-container">
2.   <li>The odd mechanism of the hooked fastener</li>
3.   <li>was perfectly well known to me</li>
4.   <li>and I snapped up the still rustless and workable
    lid and drew out the book within.</li>
5.   <li>The latter, as expected, was some twenty by
    fifteen inches in area</li>
6.   <li>and two inches thick; the thin metal covers
    opening at the top.</li>
7. </ul>
```

1. The odd mechanism of the hooked fastener
2. was perfectly well known to me
3. and I snapped up the still rustless and workable lid and drew out the book within.
4. The latter, as expected, was some twenty by fifteen inches in area
5. and two inches thick; the thin metal covers opening at the top.

## CSS (SCSS) Compiled

```
1. .flex-container {
2.   display: flex;
3.   align-items: flex-start;
4. }
5.
```

## JS

`align-items: flex-end`

Flex items *stack from the cross end* of the flex container



## HTML

```
1. <ul class="flex-container">
2.   <li>The odd mechanism of the hooked fastener</li>
3.   <li>was perfectly well known to me</li>
4.   <li>and I snapped up the still rustless and workable
    lid and drew out the book within.</li>
5.   <li>The latter, as expected, was some twenty by
    fifteen inches in area</li>
6.   <li>and two inches thick; the thin metal covers
    opening at the top.</li>
7. </ul>
```

1. The odd mechanism of the hooked fastener  
2. was perfectly well known to me  
3. and I snapped up the still rustless and workable lid and drew out the book within.  
4. The latter, as expected, was some twenty by fifteen inches in area  
5. and two inches thick; the thin metal covers opening at the top.

## CSS (SCSS) Compiled

```
1. .flex-container {
2.   display: flex;
3.   align-items: flex-end;
4. }
```

## JS

align-items: center

Flex items *stack from the center of the cross axis of the flex container*



## HTML

```
1<ul class="flex-container">
2  <li>The odd mechanism of the hooked fastener</li>
3  <li>was perfectly well known to me</li>
4  <li>and I snapped up the still rustless and workable
   lid and drew out the book within.</li>
5  <li>The latter, as expected, was some twenty by
   fifteen inches in area</li>
6  <li>and two inches thick; the thin metal covers
   opening at the top.</li>
7</ul>
```

1. The  
odd  
mechanism  
of the  
hooked  
fastener

2. was  
perfectly  
well  
known  
to me

3. and I  
snapped up the  
still rustless  
and workable  
lid and drew  
out the book  
within.

4. The  
latter, as  
expected,  
was some  
twenty by  
fifteen  
inches in  
area

5. and two  
inches  
thick; the  
thin metal  
covers  
opening at  
the top.

## CSS (SCSS) Compiled

```
1.flex-container {
2  display: flex;
3  align-items: center;
4}
5
```

## JS

## HTML

```
1. <ul class="flex-container">  
2.   <li>The odd mechanism of the hooked fastener</li>  
3. </ul>
```

### 1. The odd mechanism of the hooked fastener

Horizontally  
*and* vertically  
centered!



## CSS (SCSS) Compiled

```
1. .flex-container {  
2.   display: flex;  
3.   height: 300px;  
4.   justify-content: center;  
5.   align-items: center;
```

## JS

`align-items: baseline`

Flex items *stack so that the baselines are aligned* inside the flex container



## HTML

```
1. <ul class="flex-container">
2.   <li>The odd mechanism of the hooked fastener</li>
3.   <li>was perfectly well known to me</li>
4.   <li>and I snapped up the still rustless and workable
    lid and drew out the book within.</li>
5.   <li>The latter, as expected, was some twenty by
    fifteen inches in area</li>
6.   <li>and two inches thick; the thin metal covers
    opening at the top.</li>
7. </ul>
```

1. The  
odd  
mechanism  
of the  
hooked  
fastener

2. was  
perfectly  
well  
known  
to me

3. and I  
snapped up  
the still  
rustless and  
workable lid  
and drew out  
the book  
within.

4. The  
latter, as  
expected,  
was some  
twenty by  
fifteen  
inches in  
area

5. and  
two inches  
thick; the  
thin metal  
covers  
opening at  
the top.

## CSS (SCSS) Compiled

```
1. .flex-container {
2.   display: flex;
3.   align-items: baseline;
4. }
5. .flex-container > *:nth-child(2) {
6.   font-size: 1.5em;
7. }
8. .flex-container > *:nth-child(3) {
9.   font-size: .8em;
10. }
11. .flex-container > *:nth-child(4) {
12.   line-height: 1.6;
13. }
```

## JS

## HTML

```
1. <ul class="flex-container">
2.   <li>The odd mechanism of the hooked fastener</li>
3.   <li>was perfectly well known to me</li>
4.   <li>and I snapped up the still rustless and workable
    lid and drew out the book within.</li>
5.   <li>The latter, as expected, was some twenty by
    fifteen inches in area</li>
6.   <li>and two inches thick; the thin metal covers
    opening at the top.</li>
7. </ul>
```

1. The odd mechanism of the hooked fastener
2. was perfectly well known to me
3. and I snapped up the still rustless and workable lid and drew out the book within.
4. The latter, as expected, was some twenty by fifteen inches in area
5. and two inches thick; the thin metal covers opening at the top.

## CSS (SCSS) Compiled

```
1. .flex-container {
2.   display: flex;
3.   align-items: baseline;
4. }
5. .flex-container > *:nth-child(2) {
6.   font-size: 1.5em;
7. }
8. .flex-container > *:nth-child(3) {
9.   font-size: .8em;
10. }
11. .flex-container > *:nth-child(4) {
12.   line-height: 1.6;
13. }
```

## JS

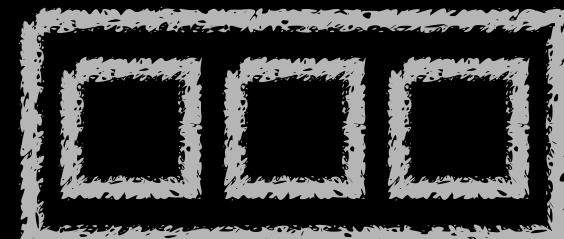
## align-self

Allows the default alignment, or the one specified for the flex container by align-items, to be overridden for individual flex items



## Values for align-self:

- » auto (Default)
- » flex-start
- » flex-end
- » center
- » baseline
- » stretch
- » start
- » end



`align-self: auto`

Uses the *value of align-items on the flex container*

If `align-items` isn't set on the flex container,  
remember that it defaults to `stretch`



## HTML

```
1. <ul class="flex-container">
2.   <li>The odd mechanism of the
      hooked fastener</li>
3.   <li>was perfectly well known to
      me</li>
4.   <li>and I snapped up the still
      rustless and workable lid and drew
      out the book within.</li>
5.   <li>The latter, as expected, was
      some twenty by fifteen inches in
      area</li>
6.   <li>and two inches thick; the
      thin metal covers opening at the
```

- 1. The odd mechanism of the hooked fastener
- 2. was perfectly well known to me
- 3. and I snapped up the still rustless and workable lid and drew out the book within.
- 4. The latter, as expected, was some twenty by fifteen inches in area
- 5. and two inches thick; the thin metal covers opening at the top.

## CSS (SCSS) Compiled

```
1. .flex-container {
2.   display: flex;
3.   height: 300px;
4.   align-items: center;
5. }
6. .flex-container > *:nth-child(3) {
7.   align-self: auto;
8. }
9
```

## JS

`align-self: flex-start/start`

Flex item *stacks from the cross start* of the flex container

`flex-start` is well-supported; the future is `start`



## HTML

```
1. <ul class="flex-container">
2.   <li>The odd mechanism of the
    hooked fastener</li>
3.   <li>was perfectly well known to
    me</li>
4.   <li>and I snapped up the still
    rustless and workable lid and drew
    out the book within.</li>
5.   <li>The latter, as expected, was
    some twenty by fifteen inches in
    area</li>
6.   <li>and two inches thick; the
    thin metal covers opening at the
```

- 1. The odd mechanism of the hooked fastener
- 2. was perfectly well known to me
- 3. and I snapped up the still rustless and workable lid and drew out the book within.
- 4. The latter, as expected, was some twenty by fifteen inches in area
- 5. and two inches thick; the thin metal covers opening at the top.

## CSS (SCSS) Compiled

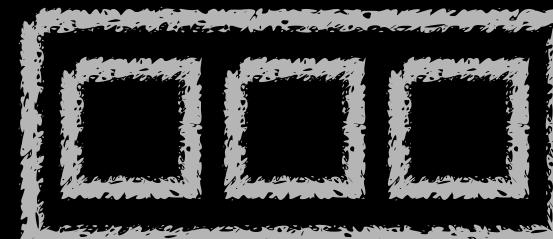
```
1. .flex-container {
2.   display: flex;
3.   height: 300px;
4.   align-items: center;
5. }
6. .flex-container > *:nth-child(3) {
7.   align-self: flex-start;
8. }
```

## JS

`align-self: flex-end/end`

Flex item stacks *from the cross end* of the flex container

`flex-end` is well-supported; the future is `end`



## HTML

```
1. <ul class="flex-container">
2.   <li>The odd mechanism of the
      hooked fastener</li>
3.   <li>was perfectly well known to
      me</li>
4.   <li>and I snapped up the still
      rustless and workable lid and drew
      out the book within.</li>
5.   <li>The latter, as expected, was
      some twenty by fifteen inches in
      area</li>
6.   <li>and two inches thick; the
      thin metal covers opening at the
```

- 
1. The odd mechanism of the hooked fastener
  2. was perfectly well known to me
  3. and I snapped up the still rustless and workable lid and drew out the book within.
  4. The latter, as expected, was some twenty by fifteen inches in area
  5. and two inches thick; the thin metal covers opening at the top.

## CSS (SCSS) Compiled

```
1. .flex-container {
2.   display: flex;
3.   height: 300px;
4.   align-items: center;
5. }
6. .flex-container > *:nth-child(3) {
7.   align-self: flex-end;
8. }
```

## JS

`align-self: center`

Flex item stacks from the center of the cross axis of the flex container



## HTML

```
1. <ul class="flex-container">
2.   <li>The odd mechanism of the
      hooked fastener</li>
3.   <li>was perfectly well known to
      me</li>
4.   <li>and I snapped up the still
      rustless and workable lid and drew
      out the book within.</li>
5.   <li>The latter, as expected, was
      some twenty by fifteen inches in
      area</li>
6.   <li>and two inches thick; the
      thin metal covers opening at the
```

- 1. The odd mechanism of the hooked fastener
- 2. was perfectly well known to me
- 3. and I snapped up the still rustless and workable lid and drew out the book within.
- 4. The latter, as expected, was some twenty by fifteen inches in area
- 5. and two inches thick; the thin metal covers opening at the top.

## CSS (SCSS) Compiled

```
1. .flex-container {
2.   display: flex;
3.   height: 300px;
4.   align-items: center;
5. }
6. .flex-container > *:nth-child(3) {
7.   align-self: center;
8. }
```

## JS

`align-self: baseline`

Flex item stacks so that the baselines are aligned inside the flex container



## HTML

```
1. <ul class="flex-container">
2.   <li>The odd mechanism of the
    hooked fastener</li>
3.   <li>was perfectly well known to
    me</li>
4.   <li>and I snapped up the still
    rustless and workable lid and drew
    out the book within.</li>
5.   <li>The latter, as expected, was
    some twenty by fifteen inches in
    area</li>
6.   <li>and two inches thick; the
    thin metal covers opening at the
```

- 1. The odd mechanism of the hooked fastener
- 2. was perfectly well known to me
- 3. and I snapped up the still rustless and workable lid and drew out the book within.
- 4. The latter, as expected, was some twenty by fifteen inches in area
- 5. and two inches thick; the thin metal covers opening at the top.

## CSS (SCSS) Compiled

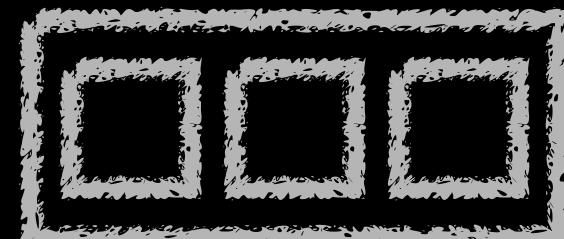
```
1. .flex-container {
2.   display: flex;
3.   height: 300px;
4.   align-items: center;
5. }
6. .flex-container > *:nth-child(3) {
7.   align-self: baseline;
8. }
```

## JS

**align-self: stretch**

Flex item *fills the whole height (or width) from cross start to cross end* of the flex container

Default value for **align-items** on the flex container



## HTML

```
1. <ul class="flex-container">
2.   <li>The odd mechanism of the
    hooked fastener</li>
3.   <li>was perfectly well known to
    me</li>
4.   <li>and I snapped up the still
    rustless and workable lid and drew
    out the book within.</li>
5.   <li>The latter, as expected, was
    some twenty by fifteen inches in
    area</li>
6.   <li>and two inches thick; the
    thin metal covers opening at the
```

- 
- 1. The odd mechanism of the hooked fastener
  - 2. was perfectly well known to me
  - 3. and I snapped up the still rustless and workable lid and drew out the book within.
  - 4. The latter, as expected, was some twenty by fifteen inches in area
  - 5. and two inches thick; the thin metal covers opening at the

## CSS (SCSS) Compiled

```
1. .flex-container {
2.   display: flex;
3.   height: 300px;
4.   align-items: center;
5. }
6. .flex-container > *:nth-child(3) {
7.   align-self: stretch;
8. }
```

## JS

# Aligning Flex Lines on the Cross Axis



## align-content

*Aligns multiple lines* of flex items within the flex container when there is extra space in the cross axis

Similar to how `justify-content` aligns individual items within the main axis



## HTML

```
1<ul class="flex-container">
2  <li>Abhoth</li>
3  <li>Azathoth</li>
4  <li>Cthulhu</li>
5  <li>Dagon</li>
6  <li>Deep One</li>
7  <li>Dunwich Horror</li>
8  <li>Formless Spawn</li>
9  <li>Hound of Tindalos</li>
10 <li>Hunting Horror</li>
11 <li>Mi-go</li>
12 <li>Nightgaunt</li>
13 <li>Nyarlathotep</li>
14 <li>Shoggoth</li>
15 <li>Shub-Niggurath</li>
16 <li>Spider of Leng</li>
```

## CSS (SCSS) Compiled

```
.flex-container {
  height: 80vh;
  flex-wrap: wrap;
  display: flex;
  align-content: flex-start;
}
.flex-container > * {
  width: 100px;
}
```

## JS

|     |                   |     |                |     |         |     |             |     |              |     |                |     |                |
|-----|-------------------|-----|----------------|-----|---------|-----|-------------|-----|--------------|-----|----------------|-----|----------------|
| 1.  | Abhoth            | 2.  | Azathoth       | 3.  | Cthulhu | 4.  | Dagon       | 5.  | Deep One     | 6.  | Dunwich Horror | 7.  | Formless Spawn |
| 8.  | Hound of Tindalos | 9.  | Hunting Horror | 10. | Mi-go   | 11. | Nightgaunt  | 12. | Nyarlathotep | 13. | Shoggoth       | 14. | Shub-Niggurath |
| 15. | Spider of Leng    | 16. | Tsathoggua     | 17. | Yig     | 18. | Yog-Sothoth | 19. | Yuggoth      |     |                |     |                |
|     |                   |     |                |     |         |     |             |     |              |     |                |     |                |



What about all this extra space?

## Values for align-content:

- » stretch (Default)
- » flex-start
- » flex-end
- » center
- » space-between
- » space-around
- » space-evenly
- » start
- » end



`align-content` only effects layout when there are *multiple lines* of flex items inside the flex container & `flex-wrap` is set to `wrap` or `wrap-reverse`

If there is only a single line of flex items, `align-content` has no effect on the layout



**align-content: stretch**

Flex items display with *distributed space after every line* of flex items

Default for **align-content**



## HTML

```
1. <ul class="flex-container">
2.   <li>Abhoth</li>
3.   <li>Azathoth</li>
4.   <li>Cthulhu</li>
5.   <li>Dagon</li>
6.   <li>Deep One</li>
7.   <li>Dunwich Horror</li>
8.   <li>Formless Spawn</li>
9.   <li>Hound of Tindalos</li>
10.  <li>Hunting Horror</li>
11.  <li>Mi-go</li>
12.  <li>Nightgaunt</li>
13.  <li>Nyarlathotep</li>
14.  <li>Shoggoth</li>
15.  <li>Shub-Niggurath</li>
16.  <li>Spider of Leng</li>
```

## CSS (SCSS) Compiled

```
.flex-container {
  height: 80vh;
  flex-wrap: wrap;
  display: flex;
  align-content: stretch;
}
.flex-container > * {
  width: 100px;
}
```

## JS

|     |                   |     |                |     |         |     |             |     |              |     |                |     |                |
|-----|-------------------|-----|----------------|-----|---------|-----|-------------|-----|--------------|-----|----------------|-----|----------------|
| 1.  | Abhoth            | 2.  | Azathoth       | 3.  | Cthulhu | 4.  | Dagon       | 5.  | Deep One     | 6.  | Dunwich Horror | 7.  | Formless Spawn |
| 8.  | Hound of Tindalos | 9.  | Hunting Horror | 10. | Mi-go   | 11. | Nightgaunt  | 12. | Nyarlathotep | 13. | Shoggoth       | 14. | Shub-Niggurath |
| 15. | Spider of Leng    | 16. | Tsathoggua     | 17. | Yig     | 18. | Yog-Sothoth | 19. | Yuggoth      |     |                |     |                |

`align-content: flex-start`

Flex items *begin at the cross start* of the flex container  
based on flex-direction



## HTML

```
1. <ul class="flex-container">
2.   <li>Abhoth</li>
3.   <li>Azathoth</li>
4.   <li>Cthulhu</li>
5.   <li>Dagon</li>
6.   <li>Deep One</li>
7.   <li>Dunwich Horror</li>
8.   <li>Formless Spawn</li>
9.   <li>Hound of Tindalos</li>
10.  <li>Hunting Horror</li>
11.  <li>Mi-go</li>
12.  <li>Nightgaunt</li>
13.  <li>Nyarlathotep</li>
14.  <li>Shoggoth</li>
15.  <li>Shub-Niggurath</li>
16.  <li>Spider of Leng</li>
```

## CSS (SCSS) Compiled

```
1. .flex-container {
2.   height: 80vh;
3.   flex-wrap: wrap;
4.   display: flex;
5.   align-content: flex-start;
6. }
7. .flex-container > * {
8.   width: 100px;
9. }
```

## JS

|     |                   |                |         |             |              |                |                |
|-----|-------------------|----------------|---------|-------------|--------------|----------------|----------------|
| 1.  | Abhoth            | Azathoth       | Cthulhu | Dagon       | Deep One     | Dunwich Horror | Formless Spawn |
| 8.  | Hound of Tindalos | Hunting Horror | Mi-go   | Nightgaunt  | Nyarlathotep | Shoggoth       | Shub-Niggurath |
| 15. | Spider of Leng    | Tsathoggua     | Yig     | Yog-Sothoth | Yuggoth      |                |                |

`align-content: flex-end`

Flex items are *stacked at the cross end* of the flex container, but not starting there



HTML

```
1. <ul class="flex-container">
2.   <li>Abhoth</li>
3.   <li>Azathoth</li>
4.   <li>Cthulhu</li>
5.   <li>Dagon</li>
6.   <li>Deep One</li>
7.   <li>Dunwich Horror</li>
8.   <li>Formless Spawn</li>
9.   <li>Hound of Tindalos</li>
10.  <li>Hunting Horror</li>
11.  <li>Mi-go</li>
12.  <li>Nightgaunt</li>
13.  <li>Nyarlathotep</li>
14.  <li>Shoggoth</li>
15.  <li>Shub-Niggurath</li>
16.  <li>Spider of Lenaz</li>
```

 CSS (SCSS) Compile

```
1 .flex-container {  
2   height: 80vh;  
3   flex-wrap: wrap;  
4   display: flex;  
5   align-content: flex-end;  
6 }  
7 .flex-container > * {  
8   width: 100px;  
9 }
```

 JS

|                            |                         |               |                        |                    |                         |                           |
|----------------------------|-------------------------|---------------|------------------------|--------------------|-------------------------|---------------------------|
| 1.<br>Abhoth               | 2.<br>Azathoth          | 3.<br>Cthulhu | 4.<br>Dagon            | 5.<br>Deep<br>One  | 6.<br>Dunwich<br>Horror | 7.<br>Formless<br>Spawn   |
| 8.<br>Hound of<br>Tindalos | 9.<br>Hunting<br>Horror | 10. Mi-<br>go | 11.<br>Nightgaunt      | 12.<br>Nyarlathote | 13.<br>Shoggoth         | 14.<br>Shub-<br>Niggurath |
| 15.<br>Spider of<br>Leng   | 16.<br>Tsathoggua       | 17. Yig       | 18.<br>Yog-<br>Sothoth | 19.<br>Yuggoth     |                         |                           |

`align-content: center`

Flex items are *stacked in the center of the cross axis* of the flex container



# HTML

```
1. <ul class="flex-container">
2.   <li>Abhoth</li>
3.   <li>Azathoth</li>
4.   <li>Cthulhu</li>
5.   <li>Dagon</li>
6.   <li>Deep One</li>
7.   <li>Dunwich Horror</li>
8.   <li>Formless Spawn</li>
9.   <li>Hound of Tindalos</li>
10.  <li>Hunting Horror</li>
11.  <li>Mi-go</li>
12.  <li>Nightgaunt</li>
13.  <li>Nyarlathotep</li>
14.  <li>Shoggoth</li>
15.  <li>Shub-Niggurath</li>
16.  <li>Spider of Leng</li>
```

# CSS (SCSS) Compiled

```
1. .flex-container {
2.   height: 80vh;
3.   flex-wrap: wrap;
4.   display: flex;
5.   align-content: center;
6. }
7. .flex-container > * {
8.   width: 100px;
9. }
```

# JS

|     |                   |                |         |             |             |                |                |
|-----|-------------------|----------------|---------|-------------|-------------|----------------|----------------|
| 1.  | Abhoth            | Azathoth       | Cthulhu | Dagon       | Deep One    | Dunwich Horror | Formless Spawn |
| 8.  | Hound of Tindalos | Hunting Horror | Mi-go   | Nightgaunt  | Nyarlathote | Shoggoth       | Shub-Niggurath |
| 15. | Spider of Leng    | Tsathoggua     | Yig     | Yog-Sothoth | Yuggoth     |                |                |

`align-content: space-between`

Rows of flex items have *equal spacing between them*,  
with *first & last rows aligned to the top & bottom edges*  
of the flex container



## HTML

```
1. <ul class="flex-container">
2.   <li>Abhoth</li>
3.   <li>Azathoth</li>
4.   <li>Cthulhu</li>
5.   <li>Dagon</li>
6.   <li>Deep One</li>
7.   <li>Dunwich Horror</li>
8.   <li>Formless Spawn</li>
9.   <li>Hound of Tindalos</li>
10.  <li>Hunting Horror</li>
11.  <li>Mi-go</li>
12.  <li>Nightgaunt</li>
13.  <li>Nyarlathotep</li>
14.  <li>Shoggoth</li>
15.  <li>Shub-Niggurath</li>
16.  <li>Spider of Leng</li>
```

## CSS (SCSS) Compiled

```
.flex-container {
  height: 80vh;
  flex-wrap: wrap;
  display: flex;
  align-content: space-between;
}
.flex-container > * {
  width: 100px;
}
```

## JS

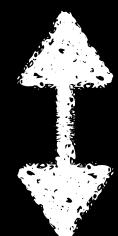
|     |                   |     |                |     |         |     |             |     |             |     |                |     |                |
|-----|-------------------|-----|----------------|-----|---------|-----|-------------|-----|-------------|-----|----------------|-----|----------------|
| 1.  | Abhoth            | 2.  | Azathoth       | 3.  | Cthulhu | 4.  | Dagon       | 5.  | Deep One    | 6.  | Dunwich Horror | 7.  | Formless Spawn |
| 8.  | Hound of Tindalos | 9.  | Hunting Horror | 10. | Mi-go   | 11. | Nightgaunt  | 12. | Nyarlathote | 13. | Shoggoth       | 14. | Shub-Niggurath |
| 15. | Spider of Leng    | 16. | Tsathoggua     | 17. | Yig     | 18. | Yog-Sothoth | 19. | Yuggoth     |     |                |     |                |

No space after last row (or before 1st row), but equal space between rows

## align-content: space-around

Rows of flex items have *equal spacing between them*,  
but space before the first row & after the last row *equals half of the space* between adjacent rows

Similar to justify-content: space-around, but focuses on rows instead of flex items



HTML

```
1<ul class="flex-container">
2  <li>Abhoth</li>
3  <li>Azathoth</li>
4  <li>Cthulhu</li>
5  <li>Dagon</li>
6  <li>Deep One</li>
7  <li>Dunwich Horror</li>
8  <li>Formless Spawn</li>
9  <li>Hound of Tindalos</li>
10 <li>Hunting Horror</li>
11 <li>Mi-go</li>
12 <li>Nightgaunt</li>
13 <li>Nyarlathotep</li>
14 <li>Shoggoth</li>
15 <li>Shub-Niggurath</li>
16 <li>Spider of Leng</li>
```

CSS (SCSS) Compiled

```
1.flex-container {
2  height: 80vh;
3  flex-wrap: wrap;
4  display: flex;
5  align-content: space-around;
6}
7.flex-container > * {
8  width: 100px;
9}
```

|     |                   |     |                |     |         |     |             |     |             |     |                |     |                |
|-----|-------------------|-----|----------------|-----|---------|-----|-------------|-----|-------------|-----|----------------|-----|----------------|
| 1.  | Abhoth            | 2.  | Azathoth       | 3.  | Cthulhu | 4.  | Dagon       | 5.  | Deep One    | 6.  | Dunwich Horror | 7.  | Formless Spawn |
| 8.  | Hound of Tindalos | 9.  | Hunting Horror | 10. | Mi-go   | 11. | Nightgaunt  | 12. | Nyarlathote | 13. | Shoggoth       | 14. | Shub-Niggurath |
| 15. | Spider of Leng    | 16. | Tsathoggua     | 17. | Yig     | 18. | Yog-Sothoth | 19. | Yuggoth     |     |                |     |                |

Space after last row (& before 1st row) is  
1/2 of space between rows

JS

`align-content: space-evenly`

Rows of flex items have *equal spacing around them, even the first & last rows*



## HTML

```
1. <ul class="flex-container">
2.   <li>Abhoth</li>
3.   <li>Azathoth</li>
4.   <li>Cthulhu</li>
5.   <li>Dagon</li>
6.   <li>Deep One</li>
7.   <li>Dunwich Horror</li>
8.   <li>Formless Spawn</li>
9.   <li>Hound of Tindalos</li>
10.  <li>Hunting Horror</li>
11.  <li>Mi-go</li>
12.  <li>Nightgaunt</li>
13.  <li>Nyarlathotep</li>
14.  <li>Shoggoth</li>
15.  <li>Shub-Niggurath</li>
16.  <li>Spider of Leng</li>
```

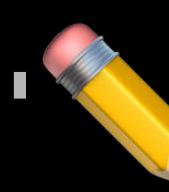
## CSS (SCSS) Compiled

```
1. .flex-container {
2.   height: 80vh;
3.   flex-wrap: wrap;
4.   display: flex;
5.   align-content: space-evenly;
6. }
7. .flex-container > * {
8.   width: 100px;
9. }
```

|                      |                   |               |                 |                |                   |                    |
|----------------------|-------------------|---------------|-----------------|----------------|-------------------|--------------------|
| 1.                   | 2.                | 3.            | 4.              | 5. Deep<br>One | 6.                | 7.                 |
| Abhoth               | Azathoth          | Cthulhu       | Dagon           |                | Dunwich<br>Horror | Formless<br>Spawn  |
| 8.                   | 9.                | 10. Mi-<br>go | 11.             | 12.            | 13.               | 14.                |
| Hound of<br>Tindalos | Hunting<br>Horror |               | Nightgaunt      | Nyarlathote    | Shoggoth          | Shub-<br>Niggurath |
| 15.                  | 16.               | 17. Yig       | 18.             | 19.            |                   |                    |
| Spider of<br>Leng    | Tsathoggua        |               | Yog-<br>Sothoth | Yuggoth        |                   |                    |

Space after last row (& before 1st row) is equal to space between rows

## JS



## SIDE NOTE

### place-content

Shorthand for `align-content` (which aligns *lines* on the *cross axis*) & `justify-content` (which aligns *items* on the *main axis*)

Values: <`align-content`>-values <`justify-content`>-values

If only 1 value is provided, it applies to both

# Gutters

**column-gap**

Defines minimum size of space (the *gutter*) between items

**row-gap**

Defines minimum size of gutter between wrapped lines

gap

Defines minimum size of gutter between rows & columns

Shorthand for setting `grid-column-gap` & `row-gap`



## PRO TIP

Due to lack of support, for now use fixed-length margins – e.g., `margin-left: 20px;` – but understand that this brings with them limitations & complications

Limitations: since you can't do a first or last line selector, you cannot exclude margins from the first or last line if wrapping occurs

Complications: if wrapping is not going to occur, you need to use `:nth-child`, `:first-child`, or `:last-child`, as needed

Or just use grid if the design allows

margin: auto

`margin: auto`

Takes all available space (horizontally & vertically)  
within a flex line

Helpful since `justify-self` property isn't available in  
Flexbox

## HTML

```
1<ul class="flex-container">
2  <li>Abhoth</li>
3  <li>Azathoth</li>
4  <li>Cthulhu</li>
5  <li>Dagon</li>
6  <li class="flex-end">Deep
    One</li>
7 </ul>
8
9<ul class="flex-container">
10 <li>Dunwich Horror</li>
11 <li>Formless Spawn</li>
12 <li class="space-around">Hound of
    Tindalos</li>
13 <li>Mi-go</li>
14 </ul>
```

## CSS (SCSS) Compiled

```
1.flex-container {
2  display: flex;
3}
4.flex-container .flex-end {
5  margin-left: auto;
6}
7.flex-container .space-around {
8  margin: auto;
9}
```

## JS



## HTML

```
1<ul class="flex-container">
2  <li>Abhoth</li>
3  <li>Azathoth</li>
4  <li>Cthulhu</li>
5  <li>Dagon</li>
6  <li class="flex-end">Deep
One</li>
7</ul>
```

## CSS (SCSS) Compiled

```
1.flex-container {
2  min-height: 100vh;
3  display: flex;
4  flex-direction: column;
5}
6.flex-container .flex-end {
7  margin-top: auto;
8}
```

## JS

1. Abhoth

2. Azathoth

3. Cthulhu

4. Dagon

5. Deep One

HTML

```
2<div class="listed-item">
3  <i class="fa fa-heart-o"></i>
4  <p class="listed-item_title">My
Favorite Tee</p>
5  <p class="listed-
item_price">$100</p>
6 </div>
```

CSS (SCSS) Compiled

```
19 }
20.listed-item .fa {
21  color: #EE96AE;
22  font-size: 24px;
23  flex-basis: 45px;
24 }
25.listed-item > p {
26  font-family: 'Bitter', serif;
27  font-size: 21px;
28 }
29.listed-item .listed-item_title {
30  color: #5F5F5F;
31 }
32.listed-item .listed-item_price {
33  color: #54CACD;
34  font-weight: bold;
35  margin-left: auto;
36 }
```

JS

My Favorite Tee \$100

My Favorite Sweatshirt \$100

My Favorite Skirt \$50

# Sizing flex items



**flex-grow**

**flex-shrink**

**flex-basis**

**flex**



By default, flex items are sized to their content,  
similarly to table cells



## PRO TIP

*Never* use **width** & **height** with flexbox layout! Instead of **width** & **height**, size items using these for the cross axis:

- » **align-items: stretch**
- » **align-content: stretch**
- » **align-self: stretch**

And this for the main axis:

- » **flex-basis**

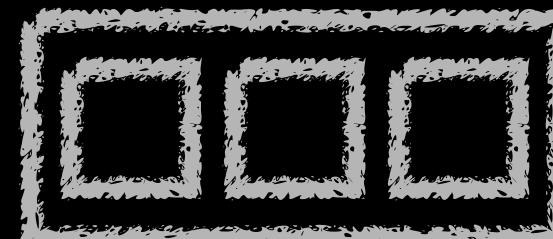
We're going to cover **flex-basis** in this section

## flex-grow

*Specifies if a flex item can grow if necessary to take up available space inside its flex container*

Value: <number>

- » 0: do not grow (default)
- » Positive <number>: flex item can grow



Any positive number for `flex-grow` allows the flex item to grow to fill all available space

All flex items that have the same number grow the same amount

If those numbers differ, flex items grow *proportionally* based upon those numbers, with larger numbers growing more

## HTML

```
1<h2>Grow not at all</h2>
2<ul>
3<li><span>Dagon</span></li>
4<li><span>Nyarlathotep</span></li>
5<li><span>Shaggai</span></li>
6</ul>
```

7

8

9

10

11

# Grow not at all

Dagon

Nyarlathotep

Shaggai

## CSS (SCSS) Compiled

```
1ul {
2  display: flex;
3}
4ul li {
5  border: 20px solid mediumvioletred;
6  background-color: deeppink;
7}
8ul li span {
9  background-color: darkmagenta;
10}
```

## JS

## HTML

```
8<h2>Grow evenly</h2>
9<ul class="grow-same">
10<li><span>Dagon</span></li>
11<li><span>Nyarlathotep</span></li>
12<li><span>Shaggai</span></li>
13</ul>
```

```
14
15
16
17
18
19
20
21
```

## Grow not at all



## Grow evenly



## CSS (SCSS) Compiled

```
12ul.grow-same li {
13  flex-grow: 1;
14}
15
```

## JS

## HTML

```
15. <h2>Grow differently</h2>
16. <ul class="grow-different">
17.   <li><span>Dagon</span></li>
18.   <li><span>Nyarlathotep</span></li>
19.   <li><span>Shaggai</span></li>
20. </ul>
21
```

## Grow not at all



## CSS (SCSS) Compiled

```
16. ul.grow-different *:nth-child(1) {
17.   flex-grow: 1;
18. }
19. ul.grow-different *:nth-child(2) {
20.   flex-grow: 2;
21. }
22. ul.grow-different *:nth-child(3) {
23.   flex-grow: 3;
24. }
```

## Grow evenly



## Grow differently



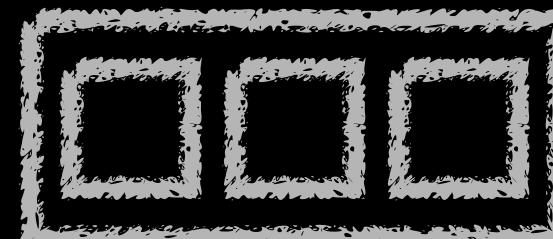
## JS

## flex-shrink

Specifies *if a flex item can shrink if necessary to fit* inside its flex container

Value: <number>

- » 1: all flex items can shrink to fit (default)
- » 0: do not shrink below the original size
- » Positive <number>: flex item can shrink



Any positive number for `flex-shrink` allows the flex item to shrink to fit within its flex container

All flex items that have the same number shrink the same amount

If those numbers differ, flex items shrink *proportionally* based upon those numbers, with larger numbers shrinking more

## HTML

```
1<h2>Shrink evenly</h2>
2<ul class="shrink-same">
3  <li><span>The Blackness
   from the Stars</span></li>
4  <li><span>The Blackness
   from the Stars</span></li>
5  <li><span>The Blackness
   from the Stars</span></li>
6 </ul>
```

7

8

9

10

11

12

## CSS (SCSS) Compiled

```
12ul.shrink-same li {
13  flex-shrink: 1;
14 }
```

## JS

# Shrink evenly

The Blackness  
from the Stars

The Blackness  
from the Stars

The Blackness  
from the Stars

## HTML

```
7
8. <h2>Shrink not at all</h2>
9. <ul class="shrink-not-at-
   all">
10.  <li><span>The Blackness
      from the Stars</span></li>
11.  <li><span>The Blackness
      from the Stars</span></li>
12.  <li><span>The Blackness
      from the Stars</span></li>
13 </ul>
14
15.
16.
```

17.

18.

## CSS (SCSS) Compiled

```
16. ul.shrink-not-at-all li {
17  flex-shrink: 0;
18 }
```

## JS

# Shrink evenly

The Blackness  
from the Stars

The Blackness  
from the Stars

The Blackness  
from the Stars

# Shrink not at all

The Blackness from the Stars

The Blackness from the Stars

## HTML

```
15. <h2>Shrink different</h2>
16. <ul class="shrink-
   different">
17.   <li><span>The Blackness
      from the Stars</span></li>
18.   <li><span>The Blackness
      from the Stars</span></li>
19.   <li><span>The Blackness
      from the Stars</span></li>
20. </ul>
```

## CSS (SCSS) Compiled

```
20. ul.shrink-different *:nth-
   child(1) {
21   flex-shrink: 1;
22 }
23. ul.shrink-different *:nth-
   child(2) {
24   flex-shrink: 2;
25 }
26. ul.shrink-different *:nth-
   child(3) {
27   flex-shrink: 3;
28 }
```

# Shrink evenly

The Blackness  
from the Stars

The Blackness  
from the Stars

The Blackness  
from the Stars

# Shrink not at all

The Blackness from the Stars

The Blackness from the Stars

# Shrink different

The Blackness from  
the Stars

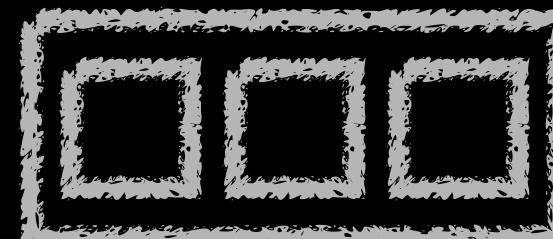
The Blackness  
from the Stars

The  
Blackness  
from the  
Stars

## JS

Note that...

- » the default for `flex-grow` is 0: do not grow
- » the default for `flex-shrink` is 1: shrink as necessary



 PRO TIP

We have said not to use `width` or `height` – & we mean that

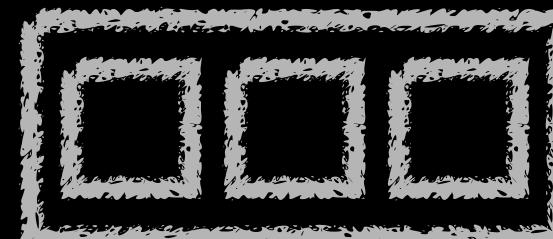
You will, however, need to use `max-width` & `min-width` on flex items to limit their maximum growth or shrinking

## flex-basis

Defines *initial main size of a flex item* before changes caused by **flex-grow** or **flex-shrink** are applied

Values:

- » **auto** (Default)
- » **content**
- » **max-content**
- » **min-content**
- » **<width>**



What about `width/height`? Why use `flex-basis` instead?

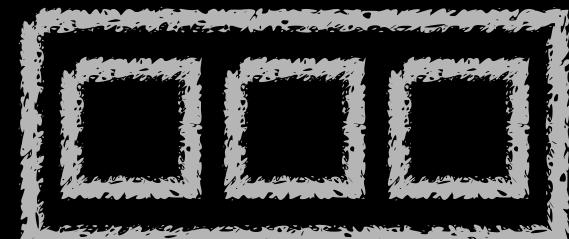
- » `flex-basis` avoids content overflow problems
- » `flex-basis` works along the main axis, so you don't need to worry about `flex-direction` vs `width/height`
- » `flex-basis` is designed to work together with `flex-grow` & `flex-shrink`
- » `flex-basis` is part of the `flex` shorthand, which the W3C recommends you use

`flex-basis: auto`

Sizes flex items *based on their content*

Warning: this assumes you have not used `width/height`, which we told you not to do!

If you do have `width/height` set, `auto` uses that value instead (see why we told you not to use it?)

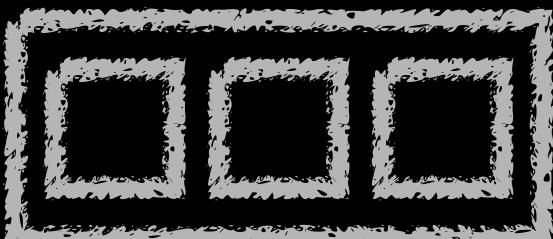


**flex-basis: content**

Sizes flex items *according to content while ignoring width/height*

Only supported in a few browsers!

You won't need this because you won't use **width/height**, right? Right? Riiiiiiight?



**flex-basis: max-content**

Sizes flex items to *longest the content can be* (if it were not wrapped)

**flex-basis: min-content**

Sizes flex items to *narrowest the content can be*, e.g., the longest word or image

**flex-basis: <length>|<percentage>**

*Overrides the CSS **height** or **width** properties of a flex item (as if you were to disobey us & use them! Never!)*

Can be either a **<length>** (e.g., **px**, **em**, **rem**, **vh**) or **<percentage>** of the flex container



**flex-basis: 200px**

Dagon

Nyarlathotep

Shaggai

**flex-basis: 125px**

Dagon

Nyarlathotep

Shaggai

**flex-basis: auto & flex-grow: 1, 1, 2**

Dagon

Nyarlathotep

Shaggai

**flex-basis: 0**

Dagon

Nyarlathotep

Shaggai

**flex-basis: 0 & flex-grow: 1, 1, 2**

Dagon

Nyarlathotep

Shaggai



**flex-grow & flex-shrink**

**flex-basis (not auto)**

**width or height (No!)**

**Content (aka `flex-basis: auto`, the default)**

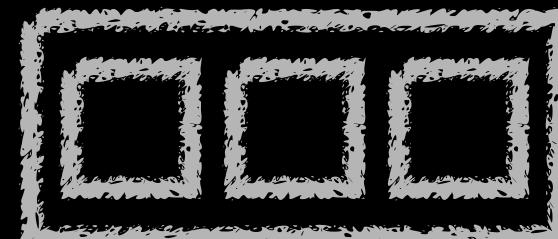
**Order of sizing effect for flex items**

## flex

Instead of `flex-grow`, `flex-shrink`, & `flex-basis`, the W3C encourages the use of the shorthand `flex` instead

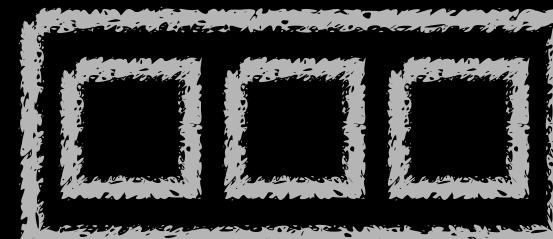
## Why?

- » `flex-grow`, `flex-shrink`, & `flex-basis` always work together when it comes to sizing flex items, & `flex` combines them intelligently
- » `flex` sets values that make sense for common uses



## flex values:

- » <flex-grow> <flex-shrink> <flex-basis>
- » initial (Default)
- » auto
- » none
- » <number>
- » <length> | <percentage>



**flex** takes 1, 2, or 3 values

1 is a keyword or <number>

2 (e.g., 1 25px), but this is really confusing

3 for <flex-grow> <flex-shrink> <flex-basis>

Of these, we recommend 1 or 3

Single value      <flex-grow> <flex-shrink> <flex-basis>

initial            0 1 auto

auto              1 1 auto

none              0 0 auto

<number>        <number> 1 0

<length>        1 1 <length>

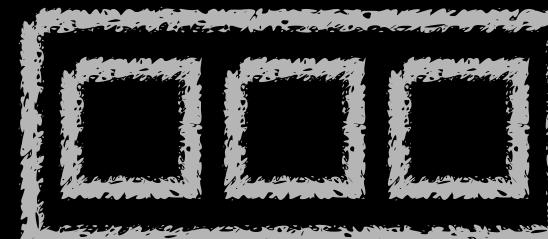
<percentage>    1 1 <percentage>

`flex: initial`

Equivalent to `0 1 auto`

- » Do not grow
- » Shrink if not enough space in container
- » Size based on content

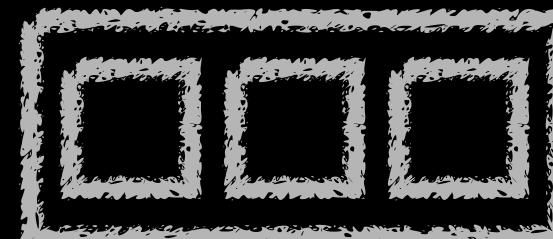
Default for `flex`



`flex: auto`

Equivalent to `1 1 auto`

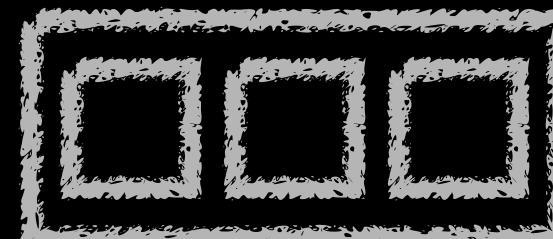
- » Grow to absorb free space
  - » Shrink if not enough space
  - » Size based on content
- } Fully flexible!



`flex: none`

Equivalent to `0 0 auto`

- » Do not grow
  - » Do not shrink
  - » Size based on content
- } Inflexible!



**flex: <number>**

Equivalent to **<number> 1 0**

- » Grow to absorb free space
- » Shrink if not enough space in container
- » Base size is *not* based on content, so growing & shrinking occur without content in mind

Typically this will results in all flex items having the same size

