CSS Flexbox Flexible, Robust Line-Based Layout

R. Scott Granneman & Jans Carton

© 2017 R. Scott Granneman Last updated 2023-10-23 You are free to use this work, with certain restrictions. For full licensing information, please see the last slide/page.

3.7

CSS Flexible Box Layout Module Level 1

W3C Candidate Recommendation, 19 November 2018

This version:

https://www.w3.org/TR/2018/CR-css-flexbox-1-20181119/

Latest published version:

https://www.w3.org/TR/css-flexbox-1/

Editor's Draft:

https://drafts.csswg.org/css-flexbox-1/

Previous Versions:

https://www.w3.org/TR/2018/CR-css-flexbox-1-20181108/ https://www.w3.org/TR/2017/CR-css-flexbox-1-20171019/ https://www.w3.org/TR/2016/CR-css-flexbox-1-20160526/ https://www.w3.org/TR/2016/CR-css-flexbox-1-20160301/ https://www.w3.org/TR/2015/WD-css-flexbox-1-20150514/ https://www.w3.org/TR/2014/WD-css-flexbox-1-20140925/ https://www.w3.org/TR/2014/WD-css-flexbox-1-20140325/ https://www.w3.org/TR/2012/CR-css3-flexbox-20120918/ https://www.w3.org/TR/2012/WD-css3-flexbox-20120612/ https://www.w3.org/TR/2012/WD-css3-flexbox-20120322/ https://www.w3.org/TR/2011/WD-css3-flexbox-20111129/ https://www.w3.org/TR/2011/WD-css3-flexbox-20110322/ https://www.w3.org/TR/2009/WD-css3-flexbox-20090723/

Tost Suito

 $\underline{\uparrow}$

 \rightarrow







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 \ddagger)





Assumes flex-direction: row



Cross End

Main Start

Cross Start







Flex Container Cross Size



Flex Item Main SizeFlex Item Cross Size

Flex Container Main Size



Main Axis: primary axis on which flex items are laid out

 $(\uparrow \text{or} \leftrightarrow)$



Provention and the second second second second



Main Start

Cross Start

Assumes flex-direction: column







Flex Container Main Size

Flex Item Main Size





Flex container Flex items Main axis





Cross axis \$

On the following slides, you'll see these symbols





display: flex display: inline-flex

flex-direction
flex-wrap
flex-flow

justify-content align-content place-content

align-items

column-gap row-gap gap



order

align-self

flex-grow
flex-shrink
flex-basis
flex





Triggering Flexbox Layout

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

Triggers flex layout for the container box:

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





* HTML

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

- J.J.

1. The odd mechanism of the hooked fastener

 \sim

V

V

\$	CSS (SCSS) Compiled
	.flex-container {
	display: flex;
	}
4	
*	IC

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.



display: inline-flex

the flexbox model:

- block
- » Immediate children become flex items » Changes the layout mode inside it

Generates an *inline box that behaves according to*

» Generates atomic inline box — an inline box that *does* not break across lines (just like display: inline-



* HTML

- The Shadow
- The odd mechanism of the hooked fastener
- vas perfectly well known to me
- and I snapped up the still rustless and workable lid and drew out the book within.
- The latter, as expected, was 6 • some twenty by fifteen inches in area
- and two inches thick; the thin metal covers opening at the top.
- Out of Time

CSS (SCSS) Compiled

```
.flex-container {
     display: inline-flex;
  }
   .flex-container > * {
     width: 70px;
6 }
* JS
```

\sim

×

×

The Shadow

1. т odd mech of the hook fastener

The
nanism
e
ed

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

display:		
flex	8*	12
inline-flex	111	12



* Uses -ms-flexbox +8-10 use -ms-inline-flexbox

Direction, Wrapping, Recepted



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

R. M. M. M. D. M. M. S. Songator



flex-direction

Values:

- \gg row (default) \rightarrow
- » row-reverse (
- » column 🗸
- » column-reverse \uparrow





flex-direction: row

Flex items are stacked in a row \rightarrow : from left-to-right (If the default for your locale is direction: rtl, then

(If the default for your locative it's the opposite)

R. M. M. M. D. M. M. S. Songator



* HTML

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

1. The odd mechanism of the hooked fastener

 \sim

 \sim

V



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.



flex-direction: row-reverse

it's the opposite)

R. M. M. M. D. M. M. S. Songator

Flex items are stacked in a row \leftarrow : from right-to-left (If the default for your locale is direction: rtl, then



* HTML

- The odd mechanism of the hooked fastener
- 3- was perfectly well known to me
- 4- and I snapped up the still rustless and workable lid and drew out the book within.
- The latter, as expected, was some twenty by fifteen inches in area
- and two inches thick; the 6. thin metal covers opening at the top.

CSS (SCSS) Compiled \mathbf{v} .flex-container { display: flex; flex-direction: row-reverse; * JS

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

 \sim

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

mechanism of the hooked fastener



flex-direction: column

Flex items are stacked \downarrow : in a column from top-tobottom







* HTML

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



 \sim

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.

1. The odd mechanism of the hooked fastener

flex-direction: column-reverse

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





* HTML

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



 \mathbf{v}

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

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

You can apply display: flex on flex items Add flex-direction to the mix & you can really have something





Card Title

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

Sig

Card Title

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



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'sll'ha ph'gel Chaugnar Faugn ch' goka stell'bsna 'fhalma ya sgn<mark>'</mark>wahl 'bthnk, 'fhalmayar nafkharanak fYoggoth oobc shu ehye lloig naNyarlathote flw'nafh, R'lyeh phlegeth kn'a Azathoth ee ooboshu grah'nnyth R'lyehyar.



1:11 🔊

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.

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.

Card Title

cdpn.io

. . . ? 🔳

Sign Up

Sign Up

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






flex-wrap

R P

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

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



flex-wrap

Values:

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





flex-wrap: nowrap

Flex items are displayed in one row





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

CSS (SCSS) Compiled

1 • 2	<pre>.flex-container { display: flex; display: flex;</pre>
3 4	<pre>tlex-wrap: nowrap; }</pre>
	.flex-container > * {
	width: 200px;

 $\mathbf{1.}$ The odd mechanism of the hooked fastener

 \sim

~

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.



1 The odd mechanism of the hooked fastener 3. vas perfectly well known to me 4- and I snapped up the still rustless and workable lid and drew out the book within. 5. The latter, as expected, was some twenty by fifteen inches in area 6. and two inches thick; the thin metal covers opening at the top.

CSS (SCSS) Compiled

- .flex-container { display: flex;
- flex-wrap: nowrap;
- .flex-container > * { width: 200px;

1. The odd mechanism o the hooked fastener

V

These are not 200px wide because flex items do *not* wrap by default

 \sim

2. was perfectly y known to	well 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 inches the thin covers of at the to
li Color Font 16px Background Padding	"Iowan Old	145.8 × 197 ■ #00 3 Style", Georgia, Senf ■ #D1E8FA 16px		
Contrast		Aa 16.63 🗸		



flex-wrap: wrap

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





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

1. The odd mechanism c hooked faste

~

4. The latter expected, wa twenty by fif inches in are

CSS (SCSS) Compiled

	<pre>.flex-container { display: flex; flex-wrap: wrap;</pre>
4	}
	.flex-container > * {
	width: 200px;
	}

~

of the ener	2. was perfectly well known to me	3. and I snapped up the still rustless and workable lid and drew out the book within.
er, as as some fteen ea	5. and two inches thick; the thin metal covers opening at the top.	

HTML

- 1
 2 Cli>The odd mechanism of the hooked
 fastener
- 3 < 3 me
- 4 and I snapped up the still
 rustless and workable lid and drew out
 the book within.
- 5- The latter, as expected, was some twenty by fifteen inches in area

CSS (SCSS) Compiled

1. .flex-container {
2 display: flex;
3 flex-wrap: wrap;
4 }
5. .flex-container > * {
6 width: 200px;
7 }
8

1. The odd mechanism of hooked faste

V

4. The latter expected, wa twenty by fit inches in are

These are 200px wide

 \sim

l of the ener	2. was perfectly well known to me	3. and I snapped up the still rustless and workable lid and drew out the book within.
er, as as some fteen ea	li Color Font 16px "Iowan Old Style", Background Padding	200 × 131 ■ #00 0 Georgia, sent ■ #D1E8FA 16px
	Contrast	Aa 16.63 🗸

flex-wrap: wrap-reverse

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





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

4. The latter expected, we twenty by fit inches in are

~

1. The odd mechanism hooked faste

CSS (SCSS) Compiled

```
1. flex-container {
2   display: flex;
3   flex-wrap: wrap-reverse;
4  }
5. flex-container > * {
6   width: 200px;
7  }
8
```

~

ter, as vas some ifteen rea	5. and two inches thick; the thin metal covers opening at the top.	
d of the ener	2. was perfectly well known to me	3. and I snapped up the still rustless and workable lid and drew out the book within.

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
                                     \sim
 1 
     The odd mechanism of the
   hooked fastener
     was perfectly well known to
   me
     and I snapped up the still
   ructlace and workahla lid and draw
CSS (SCSS) Compiled
                                     \sim
   .flex-container {
     display: flex;
   .flex-container > *:nth-child(1) {
     order: 2;
   .flex-container > *:nth-child(2) {
     order: 1;
   .flex-container > *:nth-child(3) {
     order: 0;
    .flex-container > *:nth-child(4) {
     order: 0;
   .flex-container > *:nth-child(5) {
     order: -1;
18 }
* JS
```

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

of the hooked fastener



flex-direction	11	12	28	9	9	29	4.4
flex-wrap	11*	12	28	9	9	29	4.4
flex-flow	11	12	28	9	9	29	4.4
order		12	28	9	9.2	29	4.4

* Very buggy; see chnsa.ws/1ik for more



Aligning Items & Lines



justify-content align-content

place-content

align-items align-self

safe

start end

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

	Flex items
Main axis	justify-cont
Cross axis	align-items align-self

This table shows the relationship between box alignment keywords in flexbox layout**

justify-self & justify-items do not apply with flex

* Requires flex-wrap: wrap ** Alas, these terms have different relationships in grid layout

	Flex lines*
tent	
	align-content



- » 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

PROTIP

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







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

container

R P

Default value for justify-content

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



- 1
 - The odd mechanism of the
 hooked fastener
- 3- was perfectly well known to me
- 4. and I snapped up the still
 rustless and workable lid and drew
 out the book within.

- 7

CSS (SCSS) Compiled

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

 \sim

1. The odd mechanism of the hooked fastener

2. was perfectly well known to me

 \sim

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.

justify-content: flex-end

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





- 1
- 3 3 me

- 7

CSS (SCSS) Compiled

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

* JS

 \sim

 \sim

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



justify-content: center

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





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

CSS (SCSS) Compiled

```
.flex-container {
     display: flex;
     justify-content: center;
   .flex-container > * {
     width: 80px;
7 }
```

odd

* JS

 \sim

 \sim

1. The 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.

justify-content: space-between

flex container



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



- 1
 - The odd mechanism of the hooked fastener
- vas perfectly well known to me
- 4- and I snapped up the still rustless and workable lid and drew out the book within.
- The latter, as expected, was some twenty by fifteen inches in area
- and two inches thick; the 6. thin metal covers opening at the top.

CSS (SCSS) Compiled

```
.flex-container {
 display: flex;
  justify-content: space-between;
.flex-container > * {
 width: 80px;
```

1. The odd mechanism of the hooked fastener

 \sim



V

 \sim

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

covers

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





Next \rightarrow



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



- 1
 - The odd mechanism of the
 hooked fastener
- >was perfectly well known to
 me
- and I snapped up the still rustless and workable lid and drew out the book within.
- <-li>The latter, as expected, was some twenty by fifteen inches in area

CSS (SCSS) Compiled

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

1. The odd mechanism of the hooked fastener

 \sim

 \sim



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 1/2 of space between flex items



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



- 1
 - The odd mechanism of the
 hooked fastener
- >was perfectly well known to
 me
- 4 and I snapped up the still
 rustless and workable lid and drew
 out the book within.
- <-li>The latter, as expected, was some twenty by fifteen inches in area

CSS (SCSS) Compiled

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

1. The odd mechanism of the hooked fastener



 \sim

 \sim

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





* Includes support for flex-start, flex-end, center, space-between, & space-around

			iOS		
2	29	9	9.2	29	4
'9	52	11	11	60	






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

start to cross end of the flex container

Obviously the flex item cannot have height set

Default value for align-items

TRANSPORT PROVINCE CONTRACT

Flex items fill the whole height (or width) from cross



* HTML

1 The odd mechanism of the hooked fastener vas perfectly well known to me and I snapped up the still rustless and workable lid and drew out the book within. The latter, as expected, was some twenty by fifteen inches in area

- and two inches thick; the thin metal covers 6 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.

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

inches covers

V



align-items: flex-start

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





* HTML

1 The odd mechanism of the hooked fastener was perfectly well known to me and I snapped up the still rustless and workable lid and drew out the book within. The latter, as expected, was some twenty by fifteen inches in area

- and two inches thick; the thin metal covers 6 opening at the top.



V

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

inches covers

V



align-items: flex-end

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





* HTML

1 The odd mechanism of the hooked fastener was perfectly well known to me 4. and I snapped up the still rustless and workable lid and drew out the book within. 5. The latter, as expected, was some twenty by fifteen inches in area

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



V

			14.25	
		2	4. The	-
-		3. and I	latter, as	5. a
I . The		snapped up the	expected,	inche
odd	2. was	still rustless	was some	thick
mechanism	perfectly	and workable	twenty by	thin
of the	well	lid and drew	fifteen	cove
hooked	known	out the book	inches in	open
fastener	to me	within.	area	the t



align-items: center

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





* HTML

1 The odd mechanism of the hooked fastener was perfectly well known to me 4. and I snapped up the still rustless and workable lid and drew out the book within. 5. The latter, as expected, was some twenty by fifteen inches in area

- 6. 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.	4. The latter, as expected, was some twenty by fifteen inches in area	5. inch thick thin cove oper the t
--	---	--	--	--



* HTML

1
2 Container">
3

\$	CSS (SCSS) Compiled	~
	.flex-container {	
	display: flex;	
	height: 300px;	
4	justify-content: center;	
	align-items: center;	
-		

 \sim

 $\mathbf{1}$. The odd mechanism of the hooked fastener

Horizontally and vertically centered!

align-items: baseline

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





* HTML 1 The odd mechanism of the hooked fastenervas perfectly well known to me and I snapped up the still rustless and workable lid and drew out the book within. The latter, as expected, was some twenty by fifteen inches in area and two inches thick; the thin metal covers 6 • opening at the top. 7

CSS (SCSS) Compiled

```
.flex-container {
     display: flex;
     align-items: baseline;
4 }
   .flex-container > *:nth-child(2) {
     font-size: 1.5em;
   .flex-container > *:nth-child(3) {
     font-size: .8em;
   .flex-container > *:nth-child(4) {
     line-height: 1.6;
13 }
```

1. The odd mechanism of the hooked fastener

 \sim

 \sim

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

covers



* HTML 1 The odd mechanism of the hooked fastener was perfectly well known to me 4- and I snapped up the still rustless and workable lid and drew out the book within. 5. The latter, as expected, was some twenty by fifteen inches in area 6. and two inches thick; the thin metal covers opening at the top. 7

CSS (SCSS) Compiled

```
.flex-container {
     display: flex;
     align-items: baseline;
4 }
   .flex-container > *:nth-child(2) {
     font-size: 1.5em;
   .flex-container > *:nth-child(3) {
     font-size: .8em;
   .flex-container > *:nth-child(4) {
     line-height: 1.6;
13 }
```

×

 \sim

1. The	2. was	3. and I	4. The	5.
odd mechanism of the hooked fastener	perfectly well known to me	snapped up the still rustless and workable lid and drew out the book within.	latter, as expected, was some twenty by fifteen inches in area	two thic thir cov ope the







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

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



Uses the value of align-items on the flex container



* HTML	
<pre>1 <ul class="flex-container"> 2 < >The odd mechanism of the booked fastener </pre>	
<pre>3 3 3 > me</pre>	0
4- and I snapped up the still rustless and workable lid and dr out the book within.	ew

```
\sim
```

The odd
 mechanism
 of the
 hooked
 fastener

```
CSS (SCSS) Compiled
```

area

6 -

 \sim

```
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  }
0
```

The latter, as expected, was

some twenty by fifteen inches in

and two inches thick; the

thin motal covanc anoning at the

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.



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





\$	НТ№	1L			
	<ul< th=""><th>class=</th><th>="fle</th><th>ex-containe</th><th>er"></th></ul<>	class=	="fle	ex-containe	er">
2•	<	li>The	odd	mechanism	of t

The odd mechanism of the
hooked fastener

- me
- 4. and I snapped up the still rustless and workable lid and drew out the book within.
 - The latter, as expected, was
 some twenty by fifteen inches in
 area
- 6. and two inches thick; 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: flex-start;
8 }
```

1. The odd mechanism of the hooked fastener

 \sim

 \sim

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.



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

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

and two inches thick; the 6 thin motal covanc anoning at the

out the book within.

```
CSS (SCSS) Compiled
```

```
.flex-container {
 display: flex;
 height: 300px;
 align-items: center;
.flex-container > *:nth-child(3) {
 align-self: flex-end;
```

 \sim

1. The odd mechanism of the hooked fastener

 \sim

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.



align-self: center

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





\$	нтм	1L					
	<ul< td=""><td>cla</td><td>ISS=</td><td>="fle</td><td>ex-containe</td><td>er"></td><td></td></ul<>	cla	ISS=	="fle	ex-containe	er">	
2 •	<]	li>T	he	odd	mechanism	of	the
	hool	ked	fas	stene	er		

- 4- and I snapped up the still rustless and workable lid and drew out the book within.
 - The latter, as expected, was
 some twenty by fifteen inches in
 area
- 6. and two inches thick; 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: center;
8  }
```

1. The odd mechanism of the hooked fastener

 \sim

 \sim

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.



align-self: baseline

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





\$ HTML	
	5

The odd mechanism of the
 hooked fastener

- me

 me

 me
- 4- and I snapped up the still rustless and workable lid and drew out the book within.
- 6. and two inches thick; 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: baseline;
8  }
```

1. The odd mechanism of the hooked fastener

 \sim

 \sim

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.



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





\$	НТ№	1L
	<ul< th=""><th><pre>class="flex-container"></pre></th></ul<>	<pre>class="flex-container"></pre>

The odd mechanism of the
hooked fastener

- >was perfectly well known to me
- 4 and I snapped up the still
 rustless and workable lid and drew
 out the book within.
 - The latter, as expected, was
 some twenty by fifteen inches in
 area
- 6. and two inches thick; 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  }
```

1. The odd mechanism of the hooked fastener

 \mathbf{v}

 \sim

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.



					iOS		
align-self:	11	12	27	9	9	36	4.2
baseline		79	45			57	Y
stretch	11	79	52			57	Y





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		~		
<pre>1 <ul class="flex-conto
2 Abhoth
3 Azathoth
4 Cthulhu</pre></th><th>ainer"></pre>		1. Abhoth	2. Aza	
<pre>5 Dagon 6 Deep One 7 Dunwich Horror</pre> 8 Formless Spawn 9 Hound of Tinda	 los		8. Hound of Tindalos	9. Hu Ho
<pre>10 Hunting Horror</pre> 11 Ali>Mi-go 12 Nightgaunt 13 Nyarlathotep<!--/li--> 14 Shoggoth 	li>		15. Spider of Leng	16 Tsa
<pre>15 Shub-Niggurath 16 Shub-Niggurath 16 CSS (SCSS) Compiled</pre>				
<pre>1flex-container { 2 height: 80vh; 3 flex-wrap: wrap; 4 display: flex; 5 align-content: flex; 6 }</pre>	<-start;			
<pre>7 .flex-container > * · 8 width: 100px; 9 } JS</pre>	{			

ithoth	3. Cthulhu	4. Dagon	5. Deep One	6. Dunwich Horror	7. Formless Spawn
nting rror	10. Мі- go	11. Nightgaunt	12. Nyarlathote	13. Shoggoth	14. Shub- Niggurath
. thoggua	17. Yig	18. Yog- Sothoth	19. Yuggoth		

What about all this extra space?



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

The section of the se

If there is only a single line of flex items, aligncontent has no effect on the layout



Values for align-content:

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




align-content: stretch

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

Default for align-content

and distance and the





* HTML	~		
<pre>1 <ul class="flex-container"> 2 Abhoth 3 Azathoth 4 Cthulhu 5 Dagon 6 Deep One 7 Dunwich Horror </pre>		1. Abhoth	2. Aza
<pre>8 Formless Spawn 9 Hound of Tindalos 10 Hunting Horror 11 Mi-go 12 Nightgaunt 13 Nyarlathotep 14 Shoggoth 15 Shub-Niggurath </pre>		8. Hound of Tindalos	9. Hui Hor
<pre>16. CSS (SCSS) Compiled 1 .flex-container { 2 height: 80vh; 3 flex-wrap: wrap; 4 display: flex; 5 align-content: stretch; 6 } 7 . flex container > * 5</pre>		15. Spider of Leng	16 Tsat
<pre>/* .flex-container > * {</pre>			

athoth	3. Cthulhu	4. Dagon	5. Deep One	6. Dunwich Horror	7. Formless Spawn
nting rror	10. Mi- go	11. Nightgaunt	12. Nyarlathote	13. Shoggoth	14. Shub- Niggurath
b . thoggua	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		
<pre>1 <ul class="flex-container"> 2 Abhoth 3 Azathoth 4 Cthulhu </pre>	1. Abhoth	2. Aza
<pre>5. Dagon 6. Deep One 7. Dunwich Horror 8. Formless Spawn 9. Hound of Tindalos </pre>	8. Hound of Tindalos	9. Hu Ho
<pre>10 li>Hunting Horror 11 li>Mi-go 12 Nightgaunt 13 Nyarlathotep 14 Shoggoth </pre>	15. Spider of Leng	1e Tsa
<pre>15 < 15 < 16 < _li>Snub-Niggurath <pre>16 < _li>Snider of Leng</pre> <pre>* CSS (SCSS) Compiled</pre></pre>		
<pre>1flex-container { 2 height: 80vh; 3 flex-wrap: wrap; 4 display: flex; 5 align-content: flex-start; 6 } 7flex-container > * { 8 width: 100px; </pre>		
9 } * * JS		

athoth	3. Cthulhu	4. Dagon	5. Deep One	6. Dunwich Horror	7. Formless Spawn
inting orror	10. Mi- go	11. Nightgaunt	12. Nyarlathote	13. Shoggoth	14. Shub- Niggurath
5. athoggua	17. Yig	18. Yog- Sothoth	19. Yuggoth		



align-content: flex-end

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





HTML	~							
<pre>1 <ul class="flex-container"> 2 < Abhoth 3 Azathoth 4 Cthulhu 5 Dagon 6 Deep One 7 Dunwich Horror 8 Formless Spawn 9 Hound of Tindalos </pre>								
<pre>10 Hunting Horror 11 Ali>Mi-go 12 Nightgaunt 13 Nyarlathotep 14 Shoggoth </pre>		1. Abhoth	2. Azathoth	3. Cthulhu	4. Dagon	5. Deep One	6. Dunwich Horror	7. Formless Spawn
<pre>14</pre>		8. Hound of Tindalos	9. Hunting Horror	10. Mi- go	11. Nightgaunt	12. Nyarlathote	13. Shoggoth	14. Shub- Niggurath
<pre>1flex-container { 2 height: 80vh; 3 flex-wrap: wrap; 4 display: flex; 5 align-content: flex-end; 6 }</pre>		15. Spider of Leng	16. Tsathoggua	17. Yig	18. Yog- Sothoth	19. Yuggoth		
<pre>7 .flex-container > * { 8 width: 100px; 9 } </pre>								



align-content: center

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





* HTML	\sim		
<pre>1 <ul class="flex-container"> 2 < Abhoth 3 < Azathoth 4 Cthulhu 5 < Dagon 6 Deep One 7 Dunwich Horror 8 Formless Spawn </pre>		1. Abhoth	2. Aza
<pre>10 Hunting Horror 10 Mi-go 12 Nightgaunt 13 Nyarlathotep </pre>		8. Hound of Tindalos	9. Hur Hor
<pre>14 <11>Snoggotn<!--11--> 15 <1i>Shub-Niggurath 16 Shub-Niggurath 16 CSS (scs) Compiled</pre>		15. Spider of	16 Tsat
<pre> CSS (SCSS) Compiled .flex-container { height: 80vh; flex-wrap: wrap; display: flex; align-content: center; } .flex-container > * { width: 100px; } }</pre>		Leng	
* JS			

thoth	3. Cthulhu	4. Dagon	5. Deep One	6. Dunwich Horror	7. Formless Spawn
nting rror	10. Mi- go	11. Nightgaunt	12. Nyarlathote	13. Shoggoth	14. Shub- Niggurath
. thoggua	17. Yig	18. Yog- Sothoth	19. Yuggoth		

align-content: space-between

of the flex container



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





thoth	3. Cthulhu	4. Dagon	5. Deep One	6. Dunwich Horror	7. Formless Spawn
nting ror	10. Mi- go	11. Nightgaunt	12. Nyarlathote	13. Shoggoth	14. Shub- Niggurath
thoggua	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

half of the space between adjacent rows

TRANSPORT PROVINCE CONTRACT

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

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





thoth	3. Cthulhu	4. Dagon	5. Deep One	6. Dunwich Horror	7. Formless Spawn
nting Tror	10. Mi- go	11. Nightgaunt	12. Nyarlathote	13. Shoggoth	14. Shub- Niggurath
• hoggua	17. Yig	18. Yog- Sothoth	19. Yuggoth		

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



align-content: space-evenly

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







thoth	3. Cthulhu	4. Dagon	5. Deep One	6. Dunwich Horror	7. Formless Spawn
nting rror	10. Mi- go	11. Nightgaunt	12. Nyarlathote	13. Shoggoth	14. Shub- Niggurath
. thoggua	17. Yig	18. Yog- Sothoth	19. Yuggoth		

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





place-content

(which aligns *items* on the *main* axis, or *inline* direction)

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

If only 1 value is provided, it applies to both

SIDE NOTE

Shorthand for align-content (which aligns *lines* on the cross axis, or block direction) & justify-content





Nice quick way to center flex items if & only if flexwrap: wrap is enabled:

place-content: center;

SIDE NOTE



					iOS		
align-content:	11	12	28	9	9.2	29	4.4
stretch		79	52	9	9	57	& z
space-evenly		79	52	11	11	60	Y
place-content		79	45	9	9.2	59	59









SCOBY-DOO & THE DANGERS OF DATA LOSS

Every property previously covered in this section justify-content, align-content, place-content, align-items, & align-self — also supports 2 values that work in conjunction with the other values: safe & unsafe, e.g.:

div {
 display: flex;
 align-items: safe center;
 justify-content: safe center;

٦

<div> </div>

The viewport is easily big enough for our centered image

CSS (SCSS) Compiled

```
div {
   div {
     display: flex;
     align-items: center;
```

```
justify-content: center;
```

5 }

 \sim



Now make the viewport smaller than the centered image

The image is still centered, therefore all 4 edges go outside the viewport

Let's see what happens...

* HTML

1 </i>

<img src="https://assets.codepen.io/122116/tsathaggua.jpg"</pre> alt="Cthugha the Sleeper of N'kai">

</div>

\$	CS	S (SCSS) Compiled
	di	∨ {
2		display: flex;
		align-items: center;
4		<pre>justify-content: center;</pre>
	}	
\$	JS	

so that we can't even scroll to it!







• HTML

\$	JS
	}
4	justify-content: center;
	align-items: center;
2	display: flex;
	div {
\$	CSS (SCSS) Compiled
6	
5	
4	
	alt="Cthugha the Sleeper of N'kai">
2	<pre><img src="https://assets.codepen.io/122116/tsathag</pre></pre>
	<dlv></dlv>

But we can scroll to the bottom & right without a problem

V ggua.jpg" $\mathbf{\sim}$ \sim



Why?

Since we were unable to scroll past the top or left of the viewport to see the top or left of the image, you experience what the W3C calls "data loss"

You can still scroll right & down, however, thanks to the left-to-right *direction* & top-to-bottom *writing mode* of English

Let's enable safe center for align-items & justifycontent

Now when scrolling is triggered, the image is no longer centered & is now aligned as though it was:

align-items: start; justify-content: start;



* HTML

1- <div>

<img src="https://assets.codepen.io/122116/tsathaggua.jpg"
alt="Cthugha the Sleeper of N'kai">

- 3 </div>
- CSS (SCSS) Compiled
 1 div {
- 2 display: flex;
- 3 align-items: center;
- 4 justify-content: center;
- 5 align-items: safe center;
 - justify-content: safe center;

* JS

Now we can see the top & left of the image (before, it was out of reach)...



* HTML

```
1      <div>
2      <img src="https://assets.codepen.io/122116/tsathaggua.jpg"
      alt="Cthugha the Sleeper of N'kai">
3      </div>
4
* CSS (scss) Compiled
* div {
2      display: flex;
3      align-items: center;
4      justify-content: center;
5      align-items: safe center;
```

```
6 justify-content: safe center;
```

* JS

& the bottom & right, like we always could



			iOS		
align-items: safe&unsafe	115	63		115	
justify-content: safe&unsafe	115	63		115	11

Since Safari doesn't support **safe** & **unsafe** for now, make sure your design handles the problem via other responsive behaviors such as image resizing & word wrapping





Marathon Man (1976)

Aligning via Writing Mode

Every property previously covered in this section justify-content, align-content, place-content, align-items, & align-self — also supports 2 values: start & end, e.g.:

div display: flex; justify-content: start;

start

Flex items are aligned at the logical start edge of the flex container relative to reading direction or writing mode

end

Flex items are *aligned* at the logical end edge of the flex container relative reading direction or writing-mode

Latin- & Han-based

Block start Inline end Block end

Arabic-based





Han-based



					iOS		
justify-content: start&end			45	9	9		
align-items: start&end			45				-
align-self: start&end		79	45			57	
align-content: start&end			45				-
place-content: start&end	?	2	2	2	?	?	







Gutters
column-gap

Defines minimum size of space (the gutter) between items

row-gap



Defines minimum size of gutter between wrapped lines



Defines minimum size of gutter between rows & columns

Shorthand for setting column-gap & row-gap





- 1
- The odd mechanism of the hooked fastener
- >was perfectly well known to me
- and I snapped up the still rustless and workable lid and drew out the book within.
- The latter, as expected, was some twenty by fifteen inches in area
- and two inches thick; the 6 • 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.

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 < <ul class="flex-container">
- The odd mechanism of the hooked fastener
- 3- >was perfectly well known to me
 - and I snapped up the still rustless and workable lid and drew out the book within.
 - The latter, as expected, was some twenty by fifteen inches in area
- 6. 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

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



- 1 < <ul class="flex-container">
- The odd mechanism of the hooked fastener
- 3- >was perfectly well known to me
- still rustless and workable lid and drew out the book within.
 - The latter, as expected, was some twenty by fifteen inches in area
- 6. and two inches thick; the thin metal covers opening at the top.
- 2/111×
- * CSS (SCSS)
- 1 .flex-container {
- display: flex;
- column-gap: 0.5rem;
- flex-wrap: wrap;
- row-gap: 1rem;



O JS

*	~

* ~

Compiled

* ~

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.



- 1 < <ul class="flex-container">
- The odd mechanism of the hooked fastener
- 3- >was perfectly well known to me
 - and I snapped up the still rustless and workable lid and drew out the book within.
 - The latter, as expected, was some twenty by fifteen inches in area
- 6. 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

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



		iOS		
63	14.1	14.6	84	84
63	14.1	14.6	84	84
63	14.1	14.6	84	84



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

PROTIP



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
                           \sim
1 
   Abhoth
   Azathoth
   Cthulhu
   Dagon
   class="flex-end">Deep
6.
  One
  >Dunwich Horror
10 -
   Formless Spawn
   class="space-around">Hound of
  Tindalos
   Mi-go
  CSS (SCSS) Compiled
                           \sim
  .flex-container {
   display: flex;
  .flex-container .flex-end {
   margin-left: auto;
6 }
  .flex-container .space-around {
   margin: auto;
* JS
```

1. Abhoth

1. Dunwich Horn

2. Az	athoth	3. Cthulhu	4. Dagon		5	5. De	
lorror	2. Fo	ormless Spawn		3. H	lound of Tindalos		4



* HTML

- 1
- Abhoth
- Azathoth
- Cthulhu
 - Dagon
- Deep
- One

CSS (SCSS) Compiled

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



* HTML			
<pre>2 <div class="listed-item"> 3 <i class="fa fa-heart-o"></i><th>> //</th><th>\heartsuit</th><th>My F</th></div></pre>	> //	\heartsuit	My F
<pre>Favorite Tee Favorite Tee <pre>5 < <pre>class="listed-licem_title"</pre></pre></pre>	>ıvıy	\heartsuit	My F
6			
CSS (SCSS) Compiled		\heartsuit	My F
<pre>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 } 37</pre>			
* JS	\sim		

avorite Tee

avorite Sweatshirt

avorite Skirt









Sizing Flex Items



flex-grow flex-shrink

flex-basis

flex





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





- » 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

PROTIP

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



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> Dagon 3 • Nyarlathotep Shaggai 5 -6 8 -9. 10.

CSS (SCSS) Compiled

```
ul {
      display: flex;
   }
    ul li {
      border: 20px solid mediumvioletred;
      background-color: deeppink;
 8- ul li span {
      background-color: darkmagenta;
10 }
* JS
                                          ×
```

Grow not at all

 \sim

 \sim





HTML

- 8 < <h2>Grow evenly</h2>
- 9.
- Dagon 10-
- Nyarlathotep
- Shaggai

- 16.
- 18.
- 19.
- 20





 $\boldsymbol{\sim}$







Grow not at all

Grow evenly





* HTML

- 15. <h2>Grow differently</h2>
- 16.
- Dagon
- Nyarlathotep 18-
- Shaggai 19-
- 20

CSS (SCSS) Compiled ul.grow-different *:nth-child(1) { flex-grow: 1; 18 } ul.grow-different *:nth-child(2) { flex-grow: 2; 21 22 ul.grow-different *:nth-child(3) { flex-grow: 3; 24 }

 $\boldsymbol{\sim}$

Da
Da

 $\boldsymbol{\sim}$

Grow not at all



Grow evenly



Grow differently







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





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



Any positive number for flex-shrink allows the flex

* HTML

	flex-shrink: 1;	
	ul.shrink-same li {	
\$	CSS (SCSS) Compiled	
12.		
10-		
9.		
8 -		
7		
6		
	The Blackness from the Stars	
	<pre>from the Stars</pre>	
4•	The Blackness	
	from the Stars	
2 · 3 •		
1.	<h2>Shrink evenly</h2>	
1	ch2 Shrink overlych2	

Shrink evenly

 $\mathbf{\sim}$

The Blackness from the Stars

The Blackness from the Stars

The Blackness from the Stars



HTML

8 -	<h2>Shrink</h2>	not	at	all
-----	-----------------	-----	----	-----

 \sim

×

- 9 all">
- 10 The Blackness
 from the Stars
- 11 <-li>The Blackness
 from the Stars
- 12 The Blackness
 from the Stars
- 13
- 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

Shrink not at all

The Blackness from the Stars

The Blackness from the Stars The Blackness from the Stars

The Blackness from the Stars







*	HTML
5 •	<h2>Shrink different</h2>
6•	<ul class="shrink-</th></tr><tr><th></th><th>different">
7•	The Blackness
	from the Stars
8 -	The Blackness
	from the Stars
9•	The Blackness
0	trom the Stars
0	
¥ (CSS (SCSS) Compiled
	<pre>ul.shrink-different *:nth</pre>
	<pre>child(1) {</pre>
	flex-shrink: 1;
	}
	ul.shrink-different *:nth
	$child(2)$ {
	tlex-snrink: Ζ;
	<pre>} ul chrink_different *.n+h</pre>
	child(3) {
	flex-shrink: 3:
	}

* JS

Shrink evenly

 \sim

×

The Blackness from the Stars

Shrink not at all

The Blackness from the Stars

Shrink different

The Blackness from the Stars

The Blackness from the Stars The Blackness from the Stars

The Blackness from the Stars

The Blackness from the Stars The Blackness from the Stars











Note that...

» the default for flex-grow is 0: do not grow



» the default for flex-shrink is 1: shrink as necessary





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

on flex items to limit their maximum growth or shrinking

PROTIP

You will, however, need to use max-width & min-width



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
- need to worry about flex-direction vs width/ height
- grow & flex-shrink
- W3C recommends you use

» flex-basis works along the main axis, so you don't

» flex-basis is designed to work together with flex-

» flex-basis is part of the flex shorthand, which the

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/





flex-basis: max-content

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

flex-basis: min-content

the longest word or image



Sizes flex items to narrowest the content can be, e.g.,

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





flex-basis: 0

Do not base initial size on content, so growing & shrinking occur without content in mind
flex-basis: 200px Nyarlathotep Shaggai Dagon

flex-basis: 125px

Dagon Nyarlathotep Shaggai

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

Dagon	Nyarlathotep

flex-basis: 0

Dagon	Nyarlathotep	Shaggai

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

Dagon Nyarlathotep









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 $\rightarrow \rightarrow$
- <number> \rightarrow
- » <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-grow>
initial	0 1 auto
auto	1 1 auto
none	0 0 auto
<number></number>	<number> 1</number>
<length></length>	1 1 <length< td=""></length<>
<pre><percentage></percentage></pre>	1 1 <percen< td=""></percen<>

<flex-shrink> <flex-basis>









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



ce } Fully flexible!



flex: none

Equivalent to 0 0 auto

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





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

same size



Typically this will results in all flex items having the





* JS

Console

Assets

* HTML 1 - The odd mechanism of the hooked fastener 3 - >was perfectly well known to me 4 - and I snapped up the still rustless and workable lid and drew out the book within. 5- The latter, as expected, was some twenty by fifteen inches in area CSS (SCSS) Compiled .flex-container { display: flex; 3 } 4 - .flex-container > * { 5 flex: 1 1 0;

Comments **#** Keys

1. The odd mechanism of the hooked fastener

 \sim

 \sim

V

Save

🗱 Settings

Change View

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.

Last saved LESS THAN A MINUTE AGO

Delete

 $\mathbf{\Gamma}$

Add to Collection **¬** Fork Embed Export Share





					iOS		
flex-grow	11	12	28	9	9.2	29	4.4
flex-shrink	111	12	28	9	9.2	29	4.4
flex-basis	11*	12	28	9	9.2	29	4.4
flex	151	12	28	9	9.2	29	4.4



* See note at chnsa.ws/1ii





flex-basis:					iOS		
auto	11	12	18	7	7	22	Y
content		12*	61				
max-content			66				
min-content			66				

* No longer supported as of 79!







xbugs	Watch ▼ 110	★ Star	1,637	ပို Fork	40							
of flexbox issues and cross-br	owser workarounds for the	em.										
P 2 branches	Solution Soluti Solution Solution Solution Solution Solution Solution S	a contributors	<	> Code								
branch: master - flexbugs / +						Issues						
Update flexbug 11's language and demos.												
days ago		latest commit cd3eb48e9a		🗈 Wiki								
LICENSE Initial commit				go								
README.md Update flexbug 11's language and demos.			19 days ago									
			[<u></u>	II Graphs								
			нтт	PS clone l	JRL							
			ht	tps://gi	:hub.com/p	Ê						
5			You or S	can clone ubversion.	with HTTPS, \$ ⑦	3SH,						
This repository is a community-curated list of flexbox issues and cross-browser workarounds for					Clone in Desktop							
them. The goal is that if you're building a website using flexbox and something isn't working as you'd expect, you can find the solution here.												
es to evolve and vendors nail o	lown their implementation	s, this repo will be										
	of flexbox issues and cross-bro 2 branches flexbugs / + and demos. Jays ago Initial commit Update flexbug 11's language and de community-curated list of flexb at if you're building a website u the solution here. es to evolve and vendors nail of discovered issues and remove	xbugs of flexbox issues and cross-browser workarounds for the \$2 branches © 0 releases flexbugs / + and demos. tays ago Initial commit Update flexbug 11's language and demos. S community-curated list of flexbox issues and cross-brow at if you're building a website using flexbox and something the solution here. es to evolve and vendors nail down their implementation discovered issues and remove old issues as they're fixed	Image: State in the second	xbugs	xbugs	xbugs						

updated with newly discovered issues and remove old issues as they relixed or become obsolete. If

' e	Gist	Blog	Help
------------	------	------	------













Guide

A Complete Guide to Flexbox

Last Updated Jun 12, 2020

Our comprehensive guide to CSS flexbox layout. This complete guide explains everything about flexbox, focusing on all the different possible properties for the parent element (the flex container) and the child elements (the flex items). It also includes history, demos, patterns, and a browser support chart.



Basics & Terminology









Get the poster!

Reference this guide a lot? Pin a copy up on the office wall.





Cleaner, hack-free CSS

View Project Source

CSS has been lacking proper layout mechanisms for far too long. Transitions, animations, filters, all of these are great and useful additions to the language, but they don't address the major problems that Web developers have been complaining about for what seems like an eternity.

Finally, thanks to Flexbox, we have a solution.

This site is not another CSS framework. Instead, its purpose is to showcase problems once hard or impossible to solve with CSS alone, now made trivially easy with Flexbox. And with the recent release of Internet Explorer 11 and Safari 6.1, the latest Flexbox spec is now supported in every modern browser.

Check out the demos below. View the styles in the Web inspector or dive into the source to see just how easy CSS layout will become once Flexbox becomes mainstream.





Introduction

Showcase

and the second s	Sent United sector encourt of encourt sector and a contract of the encourt contract of an encourt of encourt database interaction contra database interaction	n ga angerendina wai Interdenti Mangerendina wai Interdenti Gang rean ganda Id. kaancidak Gang rean ganda Id. kaancidak
		Your Intervent
	Afber 1	INP IC.
	sensort size in men	
	of the period of the termine	a shared
and a state of the	Full	/ Hailon
ALLE GRAD - MAN - ALLE - COLOR COLOR COLOR		
when sinds when and come directly		cut (Thirds
ADDER BRANCHY THE PARTY		Provide the
10-01140454		
	AL / Thirds	
10		
and the second s		using classes
Pol Maria	make without	out sizing cross
	colls. Colls.	
internal states and states and states	es to man ere	
want you can and share		otus
anona prant what prove as portrail		8000
The part results and and an or a second	auto	
13		
P2		1/3
BUBD		
a second s		
		1/3
auto		

Better, Simpler Grid Systems

Flexbox gives us most of the features we want from a grid system out of the box. And sizing and alignment are just one or two properties away.



Holy Grail Layout

This classic problem has been challenging CSS hackers for years, yet none of the historical solutions have fully solved it. With Flexbox, it's finally possible.



Media Object

Create media objects with fixed or varying figure sizes without worrying about overflow, clearfixing, or block formatting context hacks.



Sticky Footer

Getting your footer to stick to the bottom of sparsely contented pages has always been tricky. And if the footer's height is unknown, it's basically impossible. Not so anymore.

Showcase

[100000] [4]	Ge
	*
Appended and Prepended Add-ons	
Send Send Example One	L- manual
Atour	Encrypt
	Go
larkup	1.

Input Add-ons

Creating full-width, fluid input/button pairs has been impossible for most of the history of CSS. With Flexbox it couldn't be easier.



Vertical Centering

This classic problem has been challenging CSS hackers for years, yet none of the historical solutions have fully solved it. With Flexbox, it's finally possible.

Flexbox Playground



111 400/														
width: 12% ——					1	×	2	×	3	×	4	×	5	×
Parent Flex Pr	operties – flev	x container			0	0	0	0	0	0	0	0	0	<u>)</u>
					0	٢	0	٢	0	٢	0	<u></u>	0 flex-has	
flex-direction	1flex-wrap	justify-	align-items	align-content	1	٢	1	٢	1	٢	1	٢	1	0
🔘 row	🔘 nowrap	content	🔘 stretch	🔘 stretch	auto		auto		auto		auto		auto	
O reverse	O wrap	<pre> flex- start </pre>	O ^{flex-} start	O flex- start	auto	•	auto	•	auto	•	auto	-	auto	•
🔘 column	O reverse	<pre>O flex-end</pre>	🔘 flex-end	<pre>O flex-end</pre>										
column-		🔘 center	🔘 center	🔘 center										
• reverse		O space- between	🔘 baseline	O space- between										
		O_{around}		O space- around										
* The default properti	es are highlighed													
Children Flex	Properties – f	lex items												
The children flex p results below and the fields to see th	oroperties can be a change some of tl le property name.	applied at child level heir properties. Hov	l, separate for eac ver with the mous	ch child. See the e pointer or touch										

Result



container {	
<pre>display: flex; display: inline-flex;</pre>	
<pre>flex-direction: row; flex-direction: row-reverse; flex-direction: column; flex-direction: column-reverse;</pre>	
<pre>flex-wrap: nowrap; flex-wrap: wrap; flex-wrap: wrap-reverse;</pre>	
<pre>justify-content: flex-start; justify-content: center; justify-content: flex-end; justify-content: space-between; justify-content: space-around;</pre>	
<pre>align-items: flex-start; align-items: center; align-items: flex-end; align-items: stretch; align-items: baseline;</pre>	
<pre>align-content: flex-start; align-content: center; align-content: flex-end; align-content: space-between; align-content: space-around;</pre>	

000

i	t	e	m		{						
	0	r	d	e	r	:		0	;		
	f	ι	e	x	-	g	r	0	W	:	
	f	ι	e	x	-	g	r	0	w	:	
	f	ι	e	x	-	s	h	r	i	n	ŀ
	f	ι	e	x		s	h	r	i	n	ŀ
	f	ι	e	x	-	b	а	s	i	s	:
	а	ι	i	g	n	-	s	e	ι	f	
	а	ι	i	g	n	-	s	e	ι	f	
	а	ι	i	g	n	-	s	e	ι	f	:
	а	ι	i	g	n	t	s	e	ι	f	
	а	l	i	g	n	-	s	e	ι	f	
	а	ι	i	g	n	-	s	e	l	f	



Flexbox cheatsheet by <u>@sakamies</u>

Click a property or value to select its text. Makes it really fast to copy declarations.

The • means you selected something. (Click it to unselect.)

Click a selector to get the whole rule set with your selections. (Click outside the textarea or copy the text to clipboard to dismiss.)

Use tab, space, $\uparrow \downarrow \leftarrow \rightarrow$ keys and copy/paste for quick keyboard action.

CSS Cheats by @workflower









.container {

- display: flex; display: inline-f ex;
 - flex-direction: row; flex-direction: row-reverse,
 - flex-direction: column;
 - flex-direction: column-reverse;
 - flex-wrap: nowrap;
 - flex-wrap: wrap;
 - flex-wrap: wrap-reverse;
 - justify-content: flex-start; justify-content: center; justify-content: flex-end; justify-content: space-between; justify-content: space-around;
 - align-items: flex-start;
 - align-items: center;
 - align-items: flex-end;
 - align-items: stretch;
 - align-items: baseline;
 - align-content: flex-start; align-content: center; align-content: flex-end; align-content: space-between; align-content: space-around;

	000	
		ĺ
e , e		
	с С	
	٢	

}

.item { order: 0; flex-grow: 0; flex-grow: 1; flex-shrink: 0; flex-shrink: 1; flex-basis: auto; align-self: auto; align-self: center; align-self: baseline; align-self: stretch;





Flexbox cheatsheet by <u>@sakamies</u>

Click a property or value to select its text. Makes it really fast to copy declarations.

The • means you selected something. (Click it to unselect.)

Click a selector to get the whole rule set with your selections. (Click outside the textarea or copy the text to clipboard to dismiss.)

Use tab, space, $\uparrow\downarrow \leftarrow \rightarrow$ keys and copy/paste for quick keyboard action.

CSS Cheats by @workflower

Click on a declaration to select it





.container {	
• display: flex;	
display: inline-f.ex,	0
flex-direction: row;	
flex-direction: row-reverse;	•
<pre> flex-direction: column; </pre>	
flex-direction: column-reactse;	
flex-wrap: nowrap;	
flex-wrap: wrap;	
flex-wrap: wrap-reverse;	Ċ
justify-content: flex-start;	
justify-content: center;	
justify-content: flex-end;	
justify-content: space-between;	
justify-content: space-around;	
align-items: flex-start;	
align-items: center;	
<pre>o align-items: flex-end;</pre>	
align-items: stretch;	
align-items: baseline;	
align-content: flex-start;	
align-content: center;	
align-content: flex-end;	
align-content: space-between;	
align-content: space-around;	

.item { order: 0; flex-grow: flex-grow: flex-shrink flex-shrink flex-basis: align-self: align-self: align-self: align-self: align-self: align-self:

↓ ↑

	2
0;	 ↔]
1;	←)[]
: 0;	▶ ₩
: 1;	+ + ())
auto;	
auto;	
flex-start;	
center;	
flex-end;	
baseline;	
stretch;	

Flexbox cheatsheet by <u>@sakamies</u>

Click a property or value to select its text. Makes it really fast to copy declarations.

The • means you selected something. (Click it to unselect.)

Click a selector to get the whole rule set with your selections. (Click outside the textarea or copy the text to clipboard to dismiss.)

Use tab, space, $\uparrow \downarrow \leftarrow \rightarrow$ keys and copy/paste for quick keyboard action.

CSS Cheats by @workflower

Select multiple declarations









containor S
• display: flex.
display: inline-flex;
flex-direction: row;
flex-direction: row-reverse;
<pre> flex-direction: column; </pre>
flex-direction: column-reverse;
flex-wrap: nowrap; flex-wrap: wrap; flex-wrap: wrap-reverse;
<pre>justify-content: flex-start; justify-content: center; justify-content: flex-end; justify-content: space-between; justify-content: space-around;</pre>
align-items: flex-start; align-items: center;
align-items: flex-end;
align-items: stretch; align-items: baseline;
align-content: flex-start; align-content: center;
align-content: flex-end;
align-content: space-between;
align-content: space-around;

.item { order: 0; flex-grow: flex-grow: flex-shrink flex-shrink flex-basis: align-self: lign-self: align-self: align-self: align-self: align-elf: }

000

Î

\$

Ċ

.....

.....

-

And Street, or other



	₹ ₽ °
0;	I↔I]]
1;	
: 0;	▶ ₩
: 1;	▶ ◀]]]
auto;	
auto;	
flex-start;	
center;	
flex-end;	
baseline;	
stretch;	

Flexbox cheatsheet by <u>@sakamies</u>

Click a property or value to select its text. Makes it really fast to copy declarations.

The • means you selected something. (Click it to unselect.)

Click a selector to get the whole rule set with your selections. (Click outside the textarea or copy the text to clipboard to dismiss.)

Use tab, space, $\uparrow \downarrow \leftarrow \rightarrow$ keys and copy/paste for quick keyboard action.

CSS Cheats by @workflower

Now click on .container (or .item)











.container { display: flex; flex-direction: column; align-items: flex-end;

order: 0; flex-grow: 0; flex-grow: 1; flex-shrink: 0; flex-shrink: 1; flex-basis: auto; align-self: auto; align-self: center; align-self: flex-end; align-self: baseline; align-self: stretch;

.item {

}



Flexbox cheatsheet by <u>@sakamies</u>

Click a property or value to select its text. Makes it really fast to copy declarations.

The • means you selected something. (Click it to unselect.)

Click a selector to get the whole rule set with your selections. (Click outside the textarea or copy the text to clipboard to dismiss.)

Use tab, space, $\uparrow\downarrow \leftarrow \rightarrow$ keys and copy/paste for quick keyboard action.

CSS Cheats by @workflower

C/Ctrl+C (click anywhere to reset)







Techniques



Tag Cloud

HTML

- 1
- 2 Abhoth
- 3- Aiueb Gnshal
- 4- Aletheia
- 6 Azhorra-Tha
- 7 B'gnu-Thun
- 8 Bokrug

CSS (SCSS) Compiled

```
ul.tag-cloud {
      display: flex;
      flex-wrap: wrap;
   ul.tag-cloud li {
      display: block;
      flex-grow: 1;
      text-align: center;
    ul.tag-cloud::after {
11 content: "";
      flex-grow: 1000;
13 }
```



 $\mathbf{\sim}$

 \sim

hoth	Aiueb Gr	shal	Alethei	a Azath
horra-Th	a B'gn	u-Thun	Bokr	ug C'tha
hulhu	Cxaxuk	luth	D'endri	rah Dag
Daoloth	Γ	Darkness		Deep On
nwich Ho	orror Fo	ormless S	Spawn	Ghatanot
hroth	Gnoph	-Keh	Houn	d of Tindal
unting Ho	orror	Ialdago	rth	Kaajh'Kaal
thanid	Lu-Kt	hu l	Mh'ithrl	na Mi-
llandoth	and Mril	Thorion	Ν	lother Hyd
other of P	Pus I	Ngyr-Kor	ath	Nightgau
dens	Jy-Rakatl	h Nya	rlathote	ep Nycte
lkoth	Oryx	Oztal	un	Rhan-Tego



HTML

```
display: flex;
      flex-wrap: wrap;
   ul.tag-cloud li {
      display: block;
      flex-grow: 1;
      text-align: center;
    ul.tag-cloud::after {
11 content: "";
      flex-grow: 1000;
13 }
* JS
```



 \sim

 \sim

Aiueb Gnshal	Alet	heia	Azatho				
B'gnu-Thun Bo	okrug	C'thalp	a Cthu				
D'endrrah Da	gon	Daoloth	Darkn				
Dunwich Hor	ror	Form	less Spaw				
Ghroth Gnop	h-Keh	Hound	d of Tinda				
ror Ialdagorth	Kaaj	h'Kaalbh	Ktha				
h'ithrha Mi-go	Mland	oth and	Mril Tho				
Mother of Pus	Ngyr-	Korath	Nightga				
-Rakath Nyarlat	hotep	Nycteli	los Olk				
lun Rhan-Tegotł	n Sha	abbith-K	a Shag				
Shavalyoth Shub-Niggura							
g Star Mother	Star	Spawn	Suc'Na				



* HTML

- Abhoth
- class="obscure">Aiueb Gnshal 3 •
- Aletheia
- class="famous">Azathoth 5.
- Azhorra-Tha 6.
- B'gnu-Thun
- Bokrug 8 -

CSS (SCSS) Compiled

```
ul.tag-cloud {
  display: flex;
  flex-wrap: wrap;
ul.tag-cloud li {
  display: block;
  flex-grow: 1;
  text-align: center;
ul.tag-cloud::after {
 content: "";
  flex-grow: 1000;
```

 $\mathbf{\sim}$

 $\mathbf{\sim}$



 $\mathbf{Z}\mathbf{v}$

Oztalun	Rhan	-Tegoth	Shab	bith-K	Ka	Shag
Shathak		Shava	lyoth		Sho	ggot
Shub-Nigg	gurath	Spide	er of Le	eng	Star	· Mot
Star	Spaw	n		Suc'	Naa	th
The Black	ness fr	om the	Stars	The	Clou	ıd-Th
The Hydr	a 7	The Nam	neless I	Mist	T	hunn
Tru'nem	bra	Tsat	hoggu	a	Tu	lzsch
Ubbo-Satl	hla	Ulthar	Uvł	nash	Vo	orvad
Xa'ligha	Xexar	oth	ad-Tha	addag	Ya	ıggdy
Ycnàgnnis	SSZ	Yhound	eh N	libb-Ts	tll	Yid
Yig Yc	og-Sotł	noth	Yomag	gn'tho		Yuggo
Zvilpogght	ua					



• HTML

```
Tha
```

CSS (SCSS) Compiled

```
1. ul.tag-cloud {
2   display: flex;
3   flex-wrap: wrap;
4  }
5. ul.tag-cloud li {
6   display: block;
7   flex-grow: 1;
8   text-align: center;
9  }
10  ul.tag-cloud::after {
11   content: "";
12   flex-grow: 1000;
13  }
```



 $\mathbf{\sim}$

 $\mathbf{\sim}$

 $\mathbf{\sim}$

Nodens	Ny-Ral	kath	Nyarlatho
koth Or	yx Oz	talun	Rhan-Teg
Shagga	i Sha	thak	Shavaly
Shub-Ni	ggurath	SI	oider of Le
St	ar Spawı	n	Suc'Naat
from the	e Stars	The	Cloud-Th
The Na	ameless	Mist	Thunn'
[sathoggu	a Tulz	zscha	Ubbo-Sa
sh Vorv	adoss	Xa'ligh	a Xexar
Yag	gdytha	Y	cnàgnniss
bb-Tstll	Yidhra	Yig	Yog-Soth
luggoth	Zvilpog	gghua	
	Nodens Coth Or Shagga Shub-Ni Shub-Ni from the The Na Sathoggu Sh Vorv Yag bb-Tstll	Nodens Ny-Ral toth Oryx Oz Shaggai Sha Shub-Niggurath Star Spawn from the Stars The Nameless The Nameless Sathoggua Tulz sh Vorvadoss Yaggdytha bb-Tstll Yidhra Kuggoth Zvilpog	Nodens Ny-Rakath koth Oryx Oztalun Shaggai Shathak Shub-Niggurath Sp Star Spawn from the Stars The The Nameless Mist Sathoggua Tulzscha sh Vorvadoss Xa'ligh Yaggdytha Y bb-Tstll Yidhra Yig Yuggoth Zvilpogghua



HTML

- 1
- 3 Aiueb Gnshal
- 5 Azathoth
- 6 Azhorra-Tha
- 8 Bokrug

CSS (SCSS) Compiled

```
1 ul.tag-cloud {
2 display: flex;
3 flex-wrap: wrap;
4 }
5 ul.tag-cloud li {
6 display: block;
7 flex-grow: 1;
8 text-align: center;
9 }
10 ul.tag-cloud::after {
1 content: "";
12 flex-grow: 1000;
13 }
```

Spider of Le	ng	Star Mother Suc'Naath			
Star Spawn	n				
The Blac	kness from	n the Stars			
The Cloud-T	hing	The Hydra			
The Namele	ss Mist	Thunn'ha			
Tru'nembra	Tsathoggu	ggua Tulzsch			
Ubbo-Sathla	Ulthar		Uvhash		
Vorvadoss	Xa'ligha		Xexanoth		
Yad-Thaddag	Yaggdytha	a Ycnàgnnisssz Yidhra Yig			
Yhoundeh	Yibb-Tstll				
Yog-Sothoth	Yomagn'tl	10	Yuggoth		
Zvilpogghua					

 $\mathbf{\sim}$

V

By using Flexbox, items grow to fill the width of the container so you can avoid a ragged edge on the right

What if each of the terms is linked?

* HTML	
l - <ul class="tag-cloud">	Abhoth
<pre>2 < <a <br="" class="known">href="#_">Abhoth 3 < <a <br="" class="obscure">href="#_">Aiueb Gnshal</pre>	B'gnu-Thu
4- <a <="" class="obscure" th=""><th></th>	
CSS (SCSS) Compiled	
<pre>1 vul.tag-cloud { 2 display: flex; 3 flex-wrap: wrap; 4 } 5 ul.tag-cloud li { 6 display: contents; 7 } 8 vul.tag-cloud a { 9 display: block; 10 flex-grow: 1; 11 text-align: center; 12 } 13 ul.tag-cloud::after { 14 content: ""; 15 flex-grow: 1000; 16 } </pre>	We setbecaus <a>s

* JS



t display: contents on the se we only care about placing the



```
* HTML
                                                  \sim
1 < <ul class="tag-cloud">
       <a class="known" href="#_">Abhoth</a>
   <a class="obscure" href="#_">Aiueb</a>
3 🔻
   Gnshal</a>
       <a class="obscure"</p>
                                                  \mathbf{\sim}
CSS (SCSS) Compiled
1 - ul.tag-cloud {
     display: flex;
     flex-wrap: wrap;
4 }
  ul.tag-cloud li {
     display: contents;
   ul.tag-cloud a {
     display: block;
     flex-grow: 1;
     text-align: center;
 ul.tag-cloud::after {
     content: "";
     flex-grow: 1000;
```



* HTML

```
1 < <ul class="tag-cloud">
      <a class="known" href="#_">Abhoth</a>
      <a class="obscure" href="#_">Aiueb Gnshal</a>
      <a class="obscure" href="#_">Aletheia</a>
      <a class="famous" href="#_">Azathoth</a>
      <a class="obscure" href="#_">Azhorra-Tha</a>
CSS (SCSS) Compiled
1 - ul.tag-cloud {
    display: flex;
    flex-wrap: wrap;
4
  ul.tag-cloud li {
    display: contents;
8 - ul.tag-cloud a {
    display: block;
    flex-grow: 1;
    text-align: center;
 ul.tag-cloud::after {
    content: "";
    flex-grow: 1000;
```

 \sim

 \sim

Abhoth	Aiueb Gnshal	Alet
Azathoth	Azhorra-Tha	B'gnu-T
Bokrug	C'thalpa	Cthul
Cxaxuklutl	h	


Sticky Footer

HTML

* JS

- 1 <header>></header>
- 4 > <main>⇔</main>
- 12 <footer> /footer>



V

From The Case of Charles Dexter Ward

Charles Ward, as we have seen, first learned in 1918 of his descent from Joseph Curwen. That he at once took an intense interest in everything pertaining to the bygone mystery is not to be wondered at; for every vague rumour that he had heard of Curwen now became something vital to himself, in whom flowed Curwen's blood. No spirited and imaginative genealogist could have done otherwise than begin forthwith an avid and systematic collection of Curwen data.

In his first delvings there was not the slightest attempt at secrecy; so that even Dr. Lyman hesitates to date the youth's madness from any period before the close of 1919. He talked freely with his family though his mother was not particularly pleased to own an ancestor like Curwen—and with the officials of the various museums and libraries he visited. In applying to private families for records thought to be in their possession he made no concealment of his object, and shared the somewhat amused scepticism with which the accounts of the old diarists and letter-writers were regarded. He



* HTML

- <header>>>/header>
- <main> /main>
- <footer> / footer>



at; for every vague rumour that he had heard of Curwen now became something vital to himself, in whom flowed Curwen's blood. No spirited and imaginative genealogist could have done otherwise than begin forthwith an avid and systematic collection of Curwen data.

In his first delvings there was not the slightest attempt at secrecy; so that even Dr. Lyman hesitates to date the youth's madness from any period before the close of 1919. He talked freely with his family though his mother was not particularly pleased to own an ancestor like Curwen—and with the officials of the various museums and libraries he visited. In applying to private families for records thought to be in their possession he made no concealment of his object, and shared the somewhat amused scepticism with which the accounts of the old diarists and letter-writers were regarded. He often expressed a keen wonder as to what really had taken place a century and a half before at that Pawtuxet farmhouse whose site he vainly tried to find, and what Joseph Curwen really had been.

> The Case of Charles Dexter Ward is a short horror novel by American writer H. P. Lovecraft, written in early 1927.



* HTML

- <header>></header>
- <main>>></main>
- <footer> / footer>



Charles Ward, as we have seen, first learned in 1918 of his descent from Joseph Curwen. That he at once took an intense interest in everything pertaining to the bygone mystery is not to be wondered at; for every vague rumour that he had heard of Curwen now became something vital to himself, in whom flowed Curwen's blood. No spirited and imaginative genealogist could have done otherwise than begin forthwith an avid and systematic collection of Curwen data.



From The Case of Charles Dexter Ward

The Case of Charles Dexter Ward is a short horror novel by American writer H. P. Lovecraft, written in early 1927.



* HTML

- <header>>>/header>
- <footer> / footer>



V

From The Case of Charles Dexter Ward

Charles Ward, as we have seen, first learned in 1918 of his descent from Joseph Curwen. That he at once took an intense interest in everything pertaining to the bygone mystery is not to be wondered at; for every vague rumour that he had heard of Curwen now became something vital to himself, in whom flowed Curwen's blood. No spirited and imaginative genealogist could have done otherwise than begin forthwith an avid and systematic collection of Curwen data.

* JS

The Case of Charles Dexter Ward is a short horror novel by American writer H. P. Lovecraft, written in early 1927.





Media Objects

MadnessBook



Cthulhu is currently sleeping under the sea at R'yleh. Do not disturb. Or else.

It is a monster of vaguely anthropoid outline, but with an octopus-like head whose face was a mass of feelers, a scaly, rubbery-looking body, prodigious claws on hind and fore feet, and long, narrow wings behind.

We're going to make this in class if we have time



Azathoth, the boundless daemon sultan whose name no lips dare speak aloud, & who gnaws hungrily in inconceivable, unlighted chambers beyond time & space.



Abdul Alhazred, the "Mad Arab" who wrote the *Kitab al-Azif* (the *Necronomicon*). Devoured by an invisible beast in broad daylight in a public marketplace.



Nyarlathotep frequently walks the Earth in the guise of a human being, usually a tall, dark, slim man. He has a thousand other forms, most reputed to be maddeningly horrific.



Cards

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.

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



We're going to make this in class if we have time

Sign Up

Card Title

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

Sign Up



Thank you!

scott@granneman.com www.granneman.com ChainsawOnATireSwing.com @scottgranneman

jans@websanity.com websanity.com

CSS Flexbox Flexible, Robust Line-Based Layout

R. Scott Granneman & Jans Carton

© 2017 R. Scott Granneman Last updated 2023-10-23 You are free to use this work, with certain restrictions. For full licensing information, please see the last slide/page.

3.7

2023-10-23 3.7: Updated compatibility table for safe & unsafe

2022-10-14 3.6: Added Flexbox cheatsheet to References; added screenshots for column-gap, row-gap, & gap

2021-05-07 3.5: Updated W3C spec screenshot; in initial diagrams of terms, added diagrams assuming flexdirection: column; created *Terms* section; updated *Triggering Flexbox Layout* section; added example for flex: 1 1 0; few more details about place-content

fixed minor formatting issues

2020-07-23 3.3: Added dotted line to screenshot for justify-content: space-around; updated screenshots for Sticky Footer; converted to Granneman 1.6; fixed all formatting errors; added slides on using flexbox with items using <a> in Tag Cloud

2021-05-07 3.4: Converted theme to Granneman 1.7;

examples to Techniques

2020-07-13 3.1: Changed subtitle

2020-07-10 3.0: (con't. from \downarrow) added new section Scooby-Doo & The Dangers of Data Loss covering safe; added new section Aligning via Writing Mode covering start & end

2020-07-21 3.2: Moved some slides from *Examples* to References; changed Examples to Techniques; added new

2020-07-10 3.0: Moved Flexbox out of CSS Layout into its own slide deck; added diagram explaining basic Flexbox flow; completely re-did all screenshots; re-did almost all compatibility charts; added new section Direction, Wrapping, & Order; better examples for using multiple flex-direction; created new section Aligning Lines & Items; lots of warnings not to use width & height; renamed Sources to References; added CSS-Tricks' "A Complete Guide to Flexbox"; added 2 assignments to Examples; (con't. 1)

Licensing of this work

This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License.

To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/4.0/.

You are free to:

» Share — copy and redistribute the material in any medium or format » Adapt – remix, transform, and build upon the material for any purpose, even commercially

Under the following terms:

Attribution. You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. Give credit to:

Scott Granneman • www.granneman.com • scott@granneman.com

Share Alike. If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

No additional restrictions. You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Questions? Email scott@granneman.com