

flex-container properties

display

flex · inline-flex

flex-wrap

nowrap · wrap · wrap-reverse

flex-direction

row · row-reverse · column · column-reverse

justify-content

flex-start · flex-end · center · space-between · space-around

align-items

flex-start · flex-end · center · *stretch* · baseline

align-content

flex-start · flex-end · center · *stretch* · space-between · space-around

flex-item properties

flex-grow

0 · 1 · 2 · ...

flex-shrink

0 · *1* · 2 · ...

flex-basis

auto · value · fill · min-content · ...

flex

(= flex-grow flex-shrink flex-basis)
flex: initial; → *flex: 0 1 auto;*
flex: auto; → flex: 1 1 auto;
flex: none; → flex: 0 0 auto;
flex: value; → flex: value 1 0;

order

0 · 1 · 2 · ...

align-self

auto · flex-start · flex-end · center · stretch · baseline

internet explorer bugs

don't

.parent { flex: 1;}

.parent { flex: 1 1 0;}

.inline-child { ... }

flex-basis: value;
box-sizing: border-box;

flex-basis: calc(value);

do

.parent { flex: 1 1 0%;}

.parent { flex: 1 1 0%;}

.inline-child { display: block;}

width|height: value;
box-sizing: border-box

width|height: calc(value);

IE10-11

IE10

IE10-11

IE10-11

IE10-11

<https://github.com/philipwalton/flexbugs>

browsers support

Chrome

20- (old)
21+ (new)

Safari

3.1+ (old)
6.1+ (new)

Firefox

2-21 (old)
22+ (new)

Internet Explorer

10 (tweener)
11+ (new)

Android

2.1+ (old)
4.4+ (new)

iOS

3.2+ (old)
7.1+ (new)

Opera

12.1+ (new)

priority prevalence

1

min-width | min-height

2

max-width | max-height

3

flex-wrap

4

flex-grow | flex-shrink

5

flex-basis: value;

6

width: value;

7

flex-basis: auto;

mono-line grid

option 1 : flexible children

without gutters:

.parent { display: flex; }

.children { flex: 1 1 0%; }

with gutters:

.parent { display: flex; }

.children { flex: 1 1 0%; }

.parent > *:not(:first-child) { margin-left: gutter; }

option 2 : space-between

without gutters:

.parent { display: flex; }

.children { width: calc(100% / 3); }

with gutters:

.parent { display: flex; justify-content: space-between; }

.children { width: calc(100% / 3 - gutter / 2); }

patterns

media object

Lorem Elsass Ipsum mitt picon biere munster du ftomil Ponchour bisame. Bibbeleskaas jetz rossbolla sech choucroute un schwanz.

.parent { display: flex; align-items: flex-start; }

.content { flex: 1 1 0%; }

modifiers:

.parent--reverse { flex-direction: row-reverse; }

.figure--center { align-self: center; }

center everything

.parent { display: flex; justify-content: center; align-items: center; }

alternative:

.parent { display: flex; }

.children { margin: auto; }

resources

english

• Awesome Flexbox: <https://github.com/afonsopacifer/awesome-flexbox>

• A complete guide to Flexbox: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

• Flexbox Patterns: <http://www.flexboxpatterns.com/>

francophone

• Flexbox Layout : <http://kiwi.gg/flexbox>

• Flexbox, guide complet : <https://la-cascade.io/flexbox-guide-complet/>

• Livre « Flexbox, plongez dans les CSS modernes » : <http://goetterfr/livres/flexbox/>

unapplied properties

flex-container

• column-*

flex-item

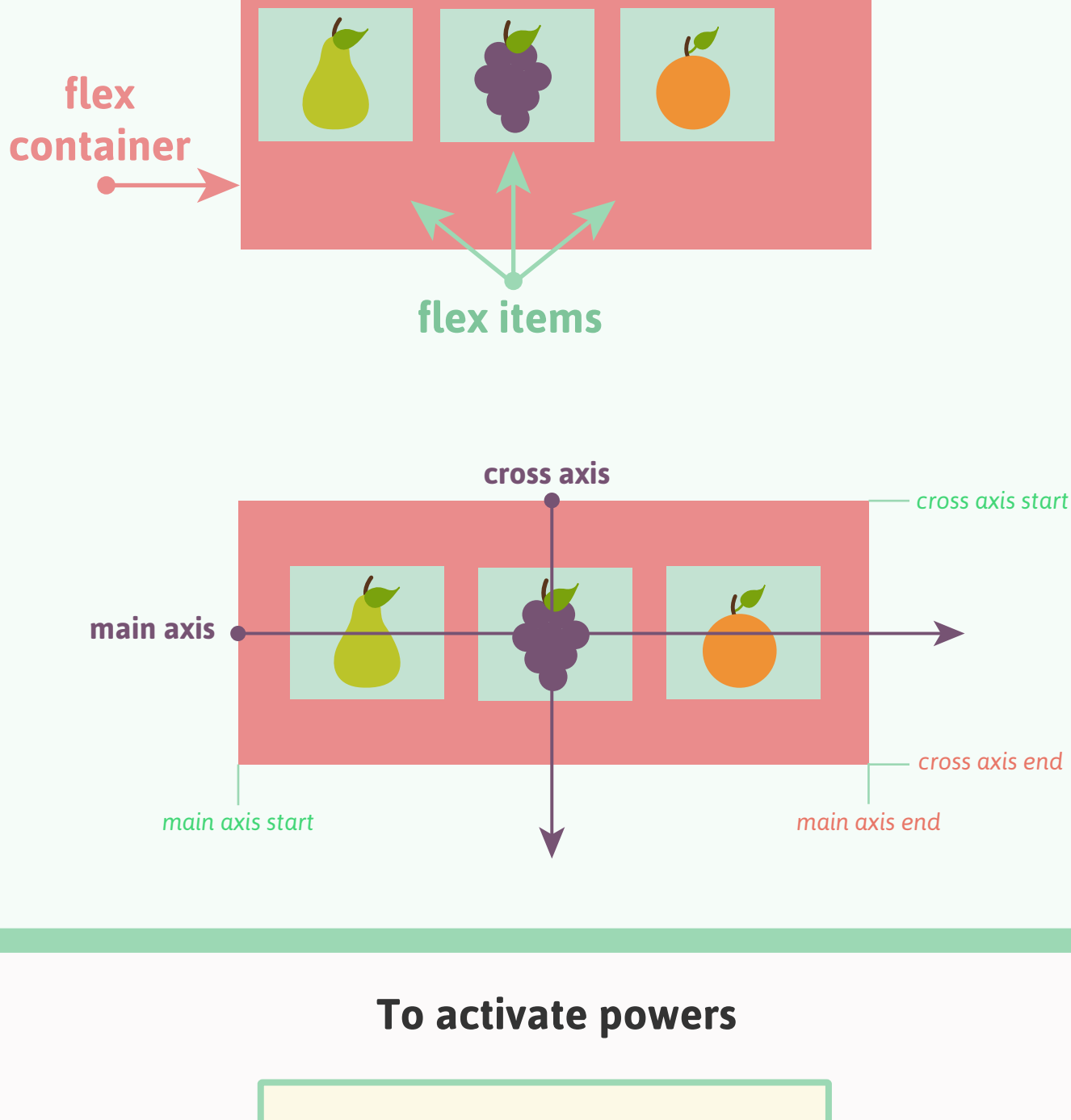
• float

• clear

• vertical-align

• ::first-line · ::first-letter

Flexbox Cheatsheet



To activate powers

```
display: flex; on flex container
```

Do you want rows or columns?

rows

```
flex-direction: row; on container
```

OR

```
flex-direction: row-reverse; on container
```



columns

```
flex-direction: column; on container
```

OR

```
flex-direction: column-reverse; on container
```



Do you want these items located at the beginning of the main axis?

YES, beginning

OK, default value has you covered.

```
justify-content: flex-start; on container
```



NO, other

OK, use one of these:

```
justify-content: flex-end; on container
```



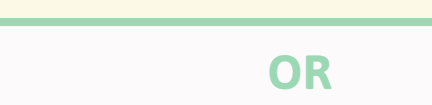
OR

```
justify-content: center; on container
```



OR

```
justify-content: space-between; on container
```



OR

```
justify-content: space-around; on container
```



Do you want the items on one line or to move to another when adjusting viewport?

one line

OK, default value has you covered.

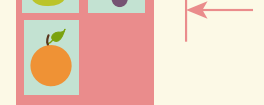
```
flex-wrap: nowrap; on container
```



move to another

OK, use one of these:

```
flex-wrap: wrap; on container
```



OR

```
flex-wrap: wrap-reverse; on container
```

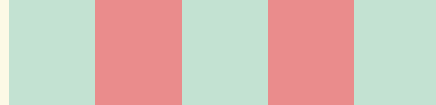


How do you want these items laid out on the cross axis?

stretched

OK, default value has you covered.

```
align-items: stretch; on container
```



not stretched

OK, use one of these:

```
align-items: flex-start; on container
```



OR

```
align-items: flex-end; on container
```



OR

```
align-items: center; on container
```



OR

```
align-items: baseline; on container
```



If you have multiple lines of content, how do you want this aligned?

stretched

OK, default value has you covered.

```
align-content: stretch; on container
```



not stretched

OK, use one of these:

```
align-content: flex-start; on container
```



OR

```
align-content: flex-end; on container
```



OR

```
align-content: center; on container
```



OR

```
align-content: space-between; on container
```



OR

```
align-content: space-around; on container
```



Do you want to change the order of the items?

NO

Awesome, don't do anything. That was easy.

YES

OK, use this:

```
order: <whole number>; on item
```

Do you need some items to grow if necessary?

NO

Awesome, don't do anything. That was easy.

YES

OK, use this:

```
flex-grow: <whole number>; on item
```

Do any of these items need to be aligned differently than the others?

NO

OK, you're done. That was super easy.

OK, use these:

```
align-self: flex-start; on item
```



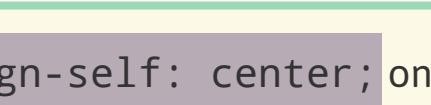
OR

```
align-self: flex-end; on item
```



OR

```
align-self: center; on item
```



OR

```
align-self: baseline; on item
```



OR

```
align-self: stretch; on item
```

