

FLEXBOX

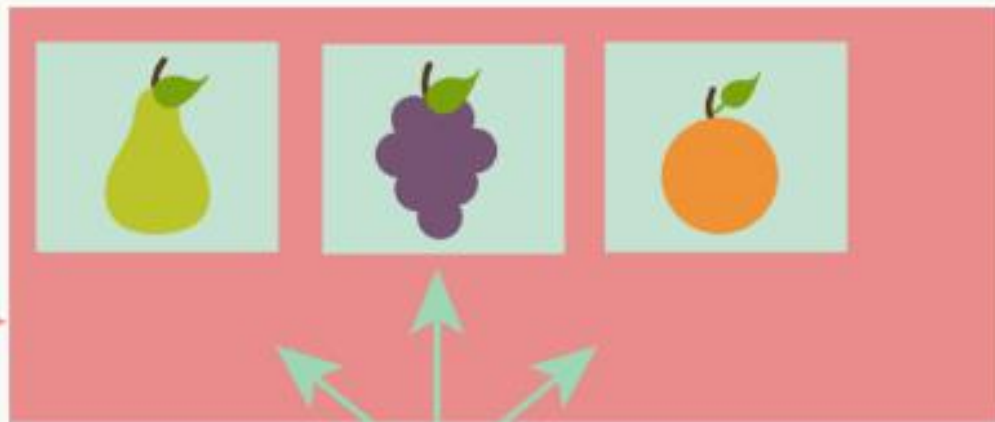
- Co to jest Flexbox, zalety
- Wsparcie dla przeglądarek, wady
- Dostępność a Flexbox
- Przykłady, zastosowanie
- Test

FLEXBOX

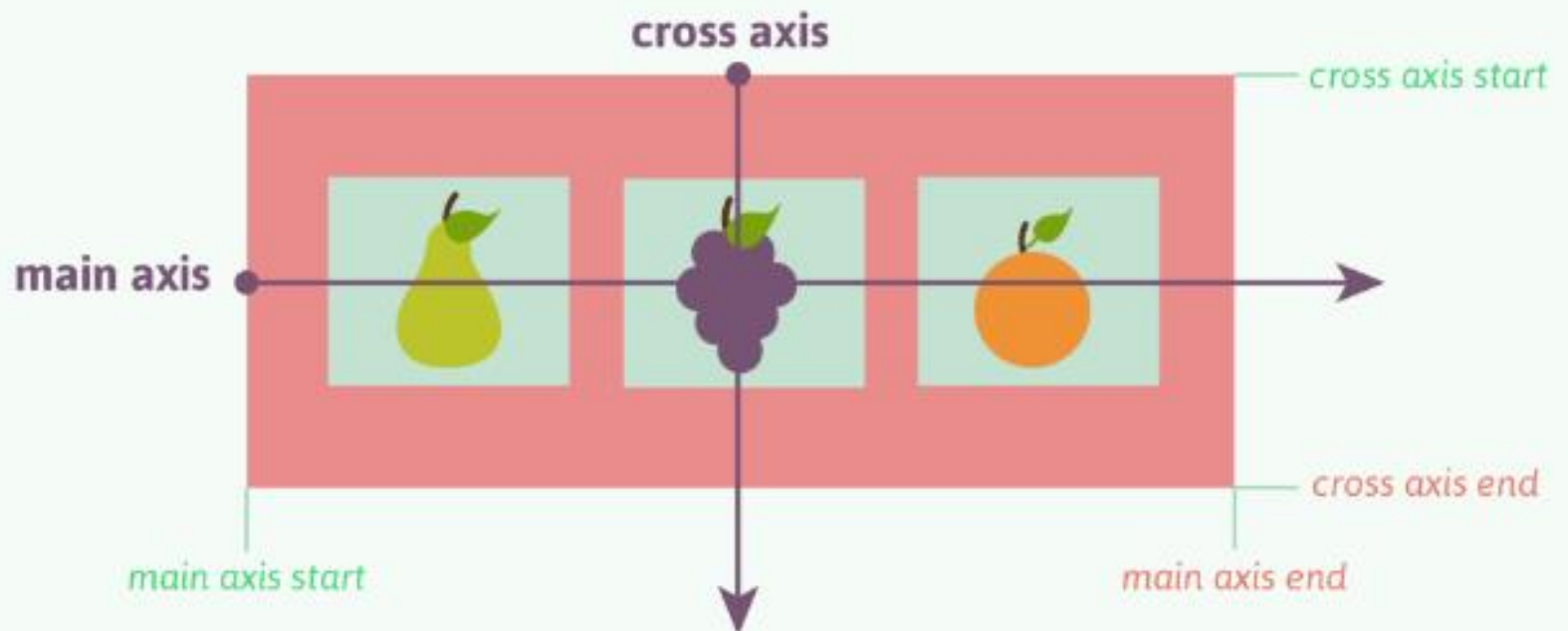
W3C Last Call Working Draft

- model pozycjonowania elementów blokowych
- kolejna wartość parametru display (display: block, display: inline, display: flex etc.)
- opisuje zachowanie kontenera i jego elementów

flex container



flex items



ZALETY FLEXBOXA

- ustalanie kolejności elementu
- ustalanie kierunku ułożenia elementów
- powiększanie lub zmniejszanie elementów
- ułożenie względem kontenera lub siebie nawzajem
- odcięcie się od jednostek
- kolumny równej wysokości
- założenie nieznanego lub dynamicznego rozmiaru elementów
- wydajność
- -> responsywność

WSPARCIE PRZEGLĄDAREK PROBLEM 1

BROWSERS THAT DON'T SUPPORT FLEXBOX

Desktop: Old versions of IE (9-) and Opera (12-)

Touch: Opera Mini (7-)

BROWSERS THAT SUPPORT THE OLD SYNTAX

All of these desktop and touch browsers require the -webkit- vendor prefix (except for Firefox, which needs the -moz- prefix).

Desktop: Firefox 2 – 21, Chrome 4 – 20, Safari 3.1 – 6

Touch: Android 2.1 – 4.3, iOS 3.2 – 6.1, UC browser 9.9 on Android, BlackBerry 7

BROWSERS THAT SUPPORT THE TWEENER SYNTAX

Desktop and touch: IE 10 (with -ms- vendor prefix)

WSPARCIE PRZEGLĄDAREK PROBLEM 1

BROWSERS THAT SUPPORT THE NEW SYNTAX

Desktop:

Unprefixed: Chrome 29+, Firefox 28+, IE 11+, Opera 17+

Prefixed: -webkit- prefix for Chrome 21+, Safari 6.1+, Opera 15+

Note: Old versions of Firefox (22 to 27) support the new syntax minus the flex-wrap and flex-flow properties. Opera (12.1+ and 17+) supports flexbox without vendor prefixes, but intermediate versions 15 and 16 require vendor prefixes.

Touch:

Unprefixed: Android 4.4+, Opera mobile 12.1+, BlackBerry 10+, Chrome for Android 39+, Firefox for Android 33+, IE 11+ mobile

Prefixed: -webkit- prefix for iOS 7.1+

```
.flex-container {
```

```
  display: -webkit-box;
```

```
  /* OLD - Safari 3.1-6, Chrome 20-, iOS, Android */
```

```
  display: -moz-box;
```

```
  /* OLD - Firefox 2-27 (buggy but mostly works) */
```

```
  display: -ms-flexbox;
```

```
  /* TWEENER - IE 10 */
```

```
  display: -webkit-flex;
```

```
  /* NEW - Safari 6.1-8, Chrome 21+, Blackberry 10+, Android 4.3-, */
```

```
  display: flex;
```

```
  /* NEW - Safari 9+, Edge 12+, Opera 12.1+, Firefox 28+, Opera Mini 8+ */
```

```
}
```



```
@mixin flexbox() {  
  display: -webkit-box;  
  display: -moz-box;  
  display: -ms-flexbox;  
  display: -webkit-flex;  
  display: flex;  
}
```

```
@mixin flex($values) {  
  -webkit-box-flex: $values;  
  -moz-box-flex: $values;  
  -webkit-flex: $values;  
  -ms-flex: $values;  
  flex: $values;  
}
```

```
@mixin order($val) {  
  -webkit-box-ordinal-group: $v;  
  -moz-box-ordinal-group: $v;  
  -ms-flex-order: $v;  
  -webkit-order: $v;  
  order: $v;  
}
```

```
.container {  
  @include flexbox();  
}  
  
.item {  
  @include flex(1 200px);  
  @include order(2);  
}
```

WSPARCIE PRZEGLĄDAREK

- **Sass mixins**

<http://smarchal.com/sass-flexbox/>

<https://gist.github.com/cimmanon/4461470>

<https://github.com/mastastealth/sass-flex-mixin>

https://github.com/lgosuki/compass-mixins/blob/master/lib/compass/css3/_flexbox.scss

- **Plugin**

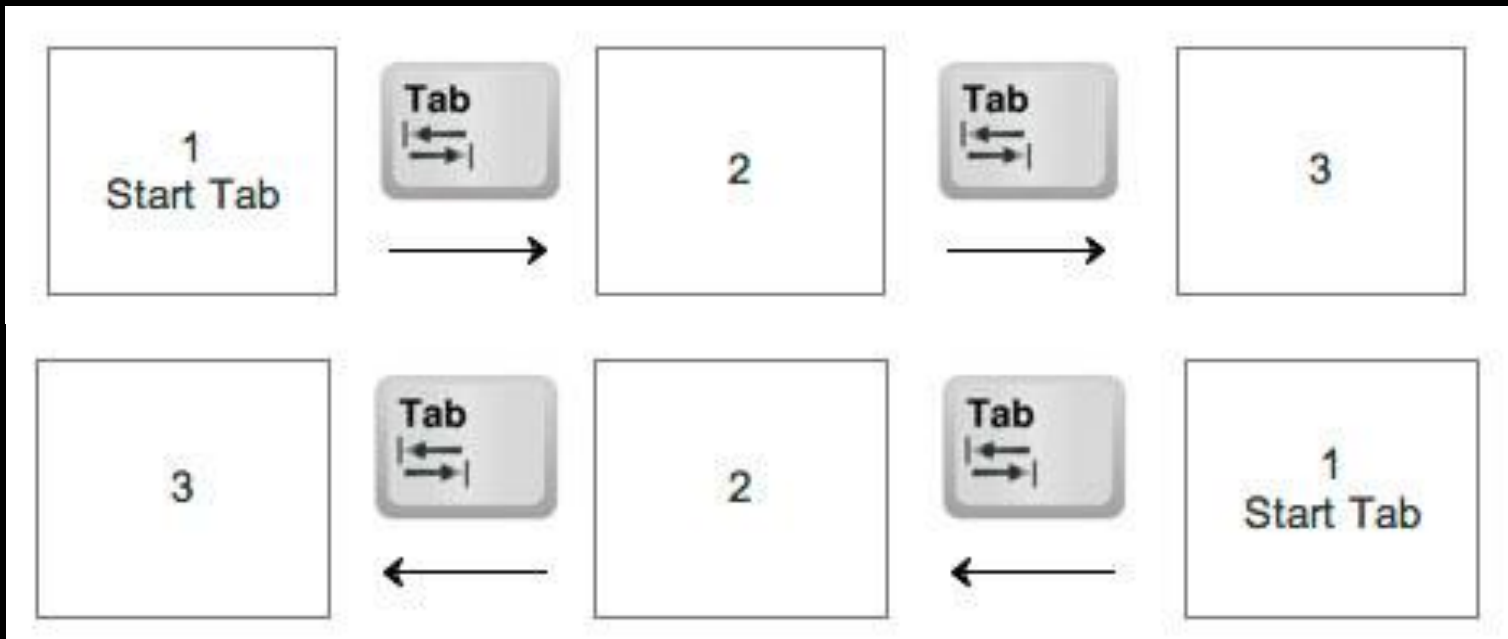
<https://github.com/postcss/autoprefixer>

WSPARCIE PRZEGLĄDAREK PROBLEM 2

- IE11
 - przy zdefiniowanej `min-height` kontenera, pozycjonowanie pionowe dla `items` nie działa
 - `flexbox` nie zadziała dla pseudoelementów (np.: `::before`, `::after`)
 - długie paragrafy nie są zawijane
 - wymaga jednostki dla `flex-basis`
- IE10
 - `-ms-flex-flow: row wrap` zadziała tylko wtedy, gdy `items` będą miały `display: inline-block`
 - domyślna wartość dla `flex` to „0 0 auto”, zamiast „0 1 auto”

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

Flexbox vs. dostępność



```
<div style="display: flex;">  
  <a href="/" style="order: 3;">One</a>  
  <a href="/" style="order: 2;">Two</a>  
  <a href="/" style="order: 1;">Three</a>  
</div>
```

aria-flowto (attribute)

- + wprowadza alternatywną nawigację dla czytników
- nie wspierana przez wszystkie czytniki
- komplikacja kodu

... aria-flowto="i3" ...

```
<div style="display: flex;">
```

```
  <a href="/" style="order: 3;" id="i1">One</a>
```

```
  <a href="/" style="order: 2;" id="i2" aria-flowto="i1">Two</a>
```

```
  <a href="/" style="order: 1;" id="i3" aria-flowto="i2">Three</a>
```

```
</div>
```

tabindex (attribute)

- + ustala pozycję danego elementu przy nawigacji klawiaturą
- zakresem jest cały dokument!
(element z `tabindex=„1”` będzie pierwszym odczytanym elementem niezależnie od położenia na stronie)

```
<div style="display: flex;">  
  <a href="/" style="order: 3;" tabindex="3">One</a>  
  <a href="/" style="order: 2;" tabindex="2">Two</a>  
  <a href="/" style="order: 1;" tabindex="1">Three</a>  
</div>
```

przeładowarka

Firefox –

mechanizm nawigujący w kolejności wyświetlającej się na stronie -> zgłoszony błąd w implementacji, przeciwnej specyfikacji flexboxa



Example One

Send

- **HTML:**

```
<div class="inputContainer">  
  <span class="inputItem">...</span>  
  <input class="inputItem-field"></input>  
  <button class="inputItem">...</button>  
</div>
```

- **CSS:**

```
.inputContainer { display: flex; }  
.inputItem-field { flex: 1; }  
.inputItem { /* item styles */ }
```



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

- **flex-grow** - proporcja względem wszystkich elementów, jak bardzo item urośnie względem innych elementów, w przypadku wolnej powierzchni
- **flex-shrink** - za stopień "kompresji" elementu, czyli o ile razy zmniejszyć dany element
- **flex-basis** - bazowa szerokość elementu, inaczej mówiąc minimalna szerokość elementu

flex-grow

Item 1

Item 2

```
.flex-container{ width: 600px; }  
.flex-item-1{ flex-basis: 200px; flex-grow: 3; }  
.flex-item-2{ flex-basis: 200px; flex-grow: 1; }
```

Available space = (container size - basis-total)
container size(600) - basis total (400) = 200

Grow unit = (Available space / grow-total)
200 / 4 = 50

*Flex item size = (Flex basis + (Grow Unit * num))*
Item 1: 200px + (50*3) = 350px
Item 2: 200px + (50*1) = 250px

flex-shrink

Item
1

Item 2

Item 2

```
.flex-container {width: 600px;}  
.flex-item-1 {flex-basis: 100px; flex-shrink: 1;}  
.flex-item-2 {flex-basis: 400px; flex-shrink: 1;}  
.flex-item-3 {flex-basis: 400px; flex-shrink: 1;}
```

Space remaining = (container size - basis-total)
container size(600) - basis total (900) = -300

Flex item size = (Flex basis + (Flex basis num/ basis-total))*

Item 1: 100px + (100*1/900) which is 66px

Item 2: 400px + (400*1/900) which is 266px

Item 3: 400px + (400*1/900) which is 266px

Text Input:

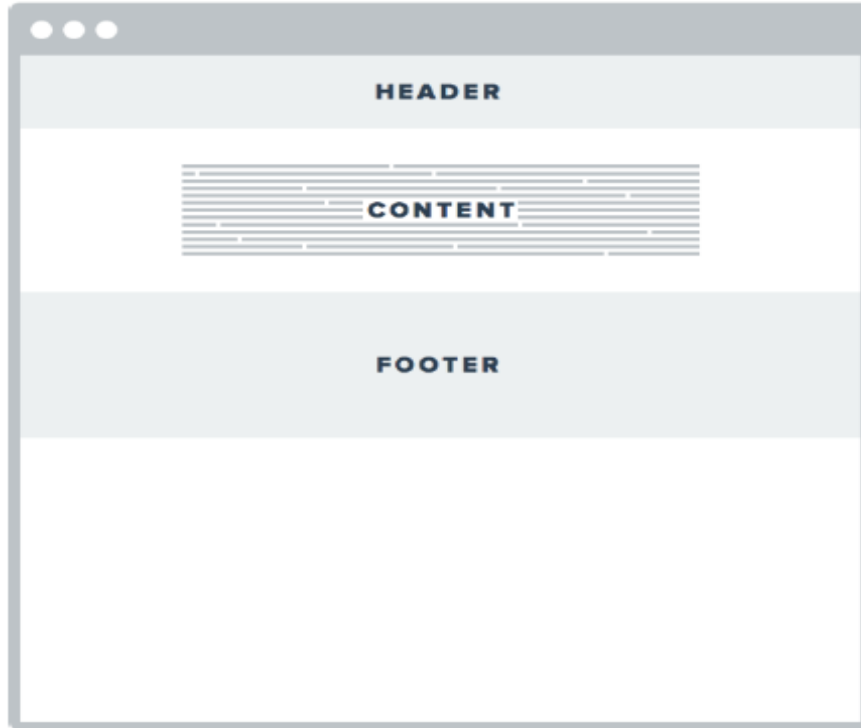
Another Text Input:

What about a select?:

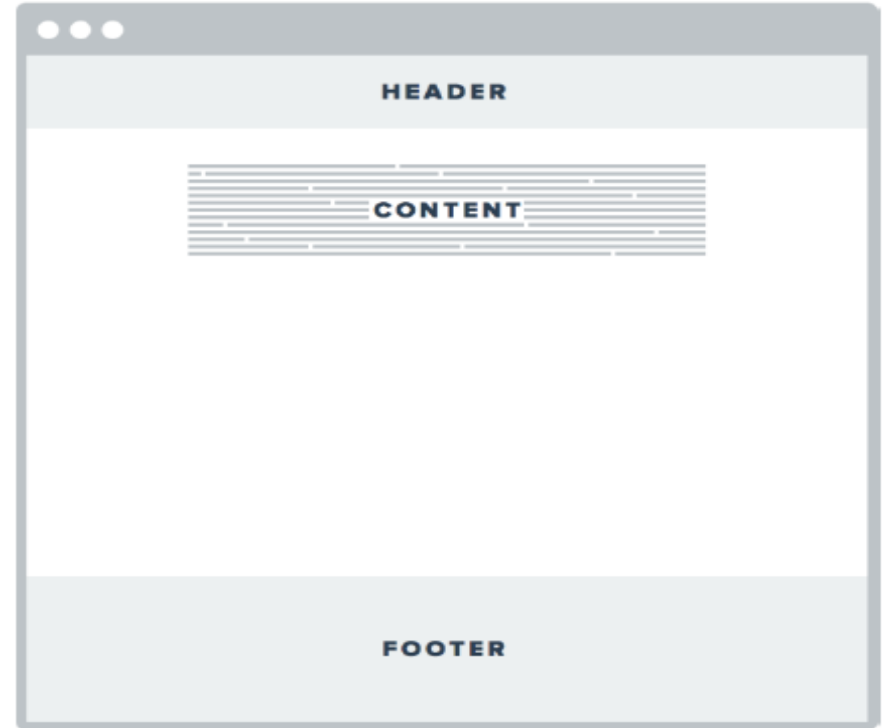
```
p {  
  display: flex;  
}
```

```
input, select {  
  flex-grow: 1;  
}
```

Default Behavior

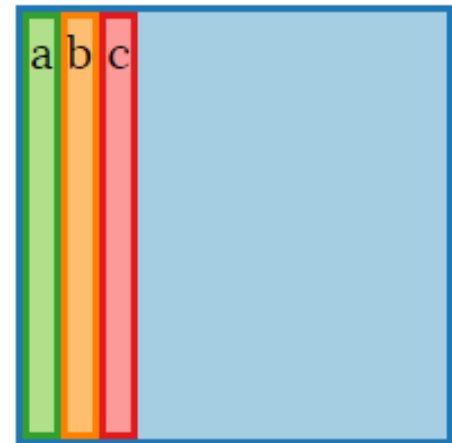


Desired Effect

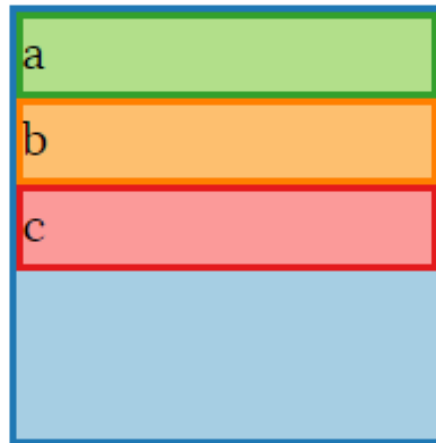


```
body {  
  display: flex;  
  min-height: 100vh;  
  flex-direction: column;  
}  
  
.content {  
  flex: 1;  
}
```

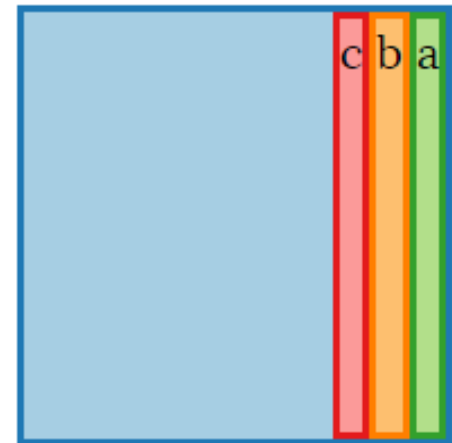
flex-direction



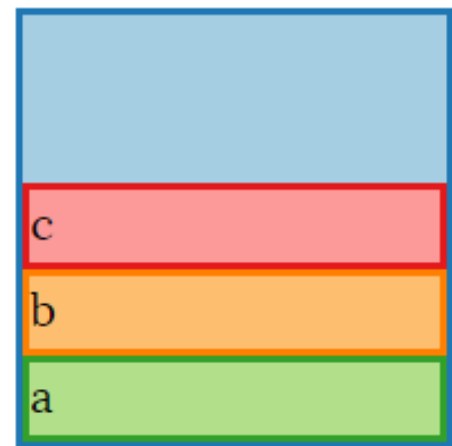
flex-direction: row



flex-direction: column



flex-direction: row-reverse



flex-direction: column-reverse

I'm Centered!

This box is both vertically and horizontally centered. Even if the text in this box changes to make it wider or taller, the box will still be centered. Go ahead, give it a try. Just click to edit the text.

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

justify-
content

MAIN AXIS

flex-start



flex-end



center



space-between



space-around



align-items

flex-start



flex-end



CROSS AXIS

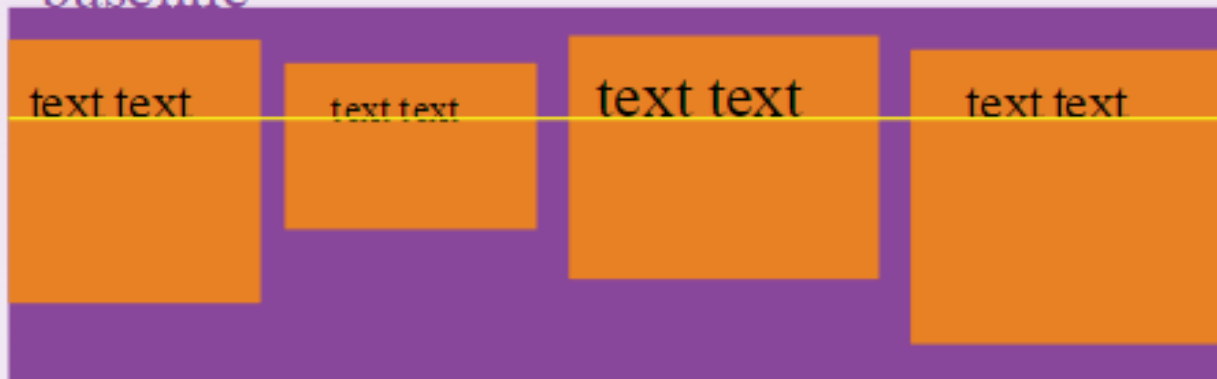
center



stretch



baseline



justify-content

Main axis alignment

(horizontal when row,
vertical when column)

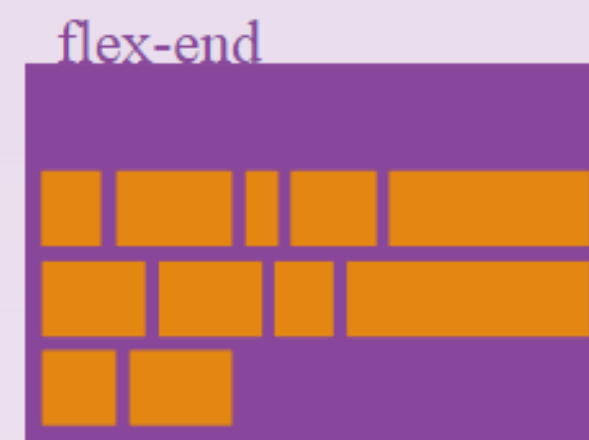
align-items

Cross axis alignment

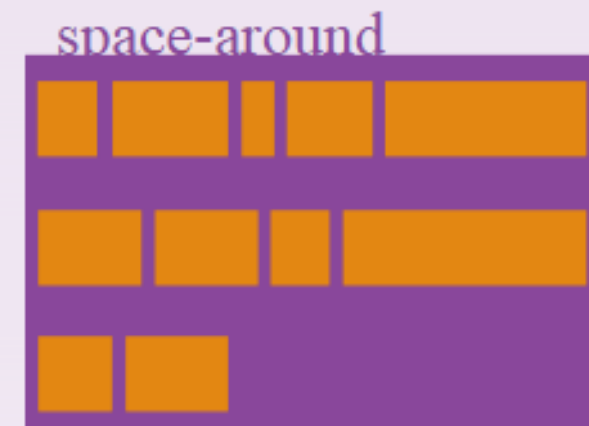
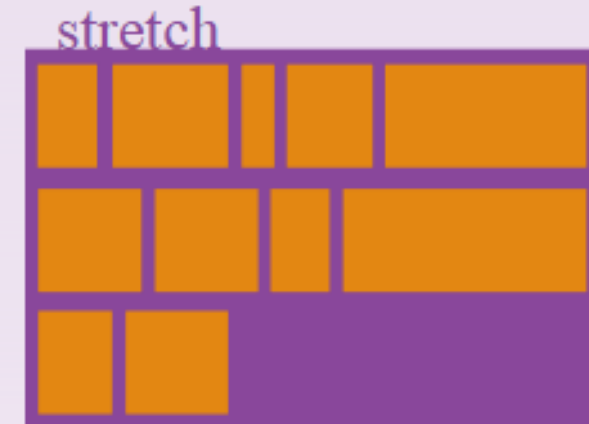
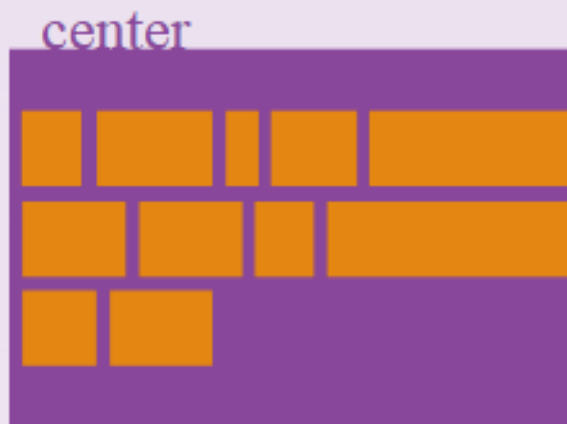
(vertical when row,
horizontal when column)

(P.S. Also responsible for
equal-height columns)

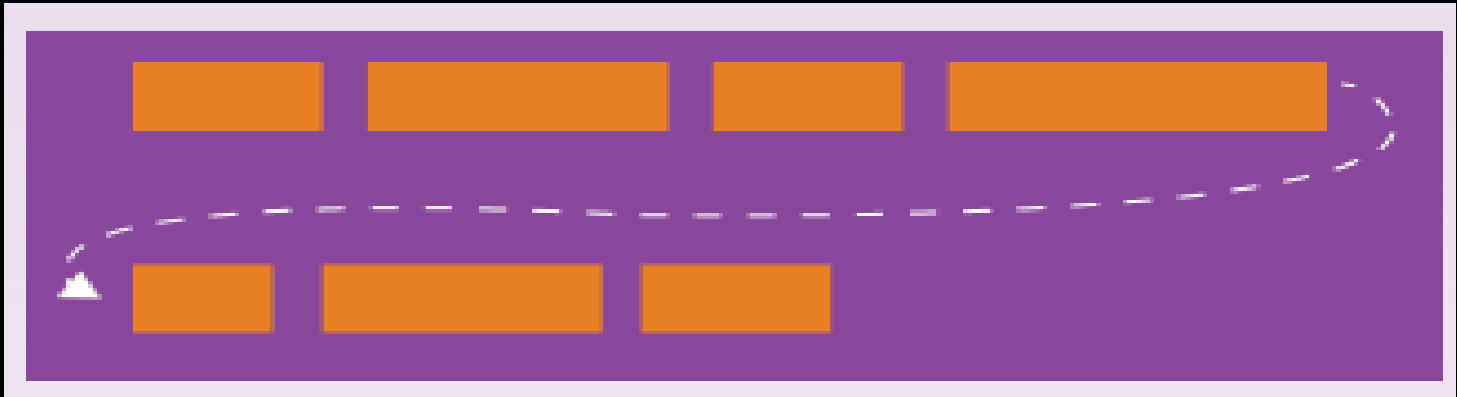
align-
content



CROSS AXIS

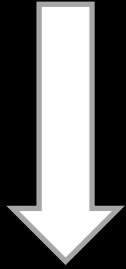


flex-wrap



```
.container{  
  flex-wrap: nowrap | wrap | wrap-reverse;  
}
```

CONTAINER



flex-direction
flex-wrap
justify-content
align-content
align-items
flex-flow

ITEM



flex
flex-grow
flex-shrink
flex-basis
align-self
order

Inbox

Contacts

Account

Change theme

Legal



Science potion

Costs \$5



The calendar feature makes scheduling a snap.

Our dashboard gives you a bird's-eye-view-at-a-glance-on-the-go.



You can get notified by email or SMS.

- the end -