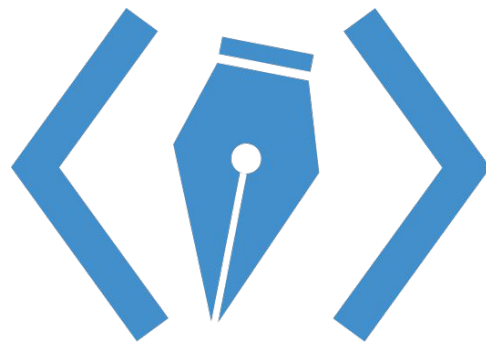


Week 05

Flexbox and Layouts



Announcements

Submission Errors

Make sure your lab/hw zip files are less than 20MB or else they won't submit!

Midterm Project

The project check in (4 points) deadline is this **Thursday, Oct 8th!**

The final project is due on **Tues, Oct 20th.**

Submission (due 10/20) : wdd.io/go/project-submit

Come to office hours via Piazza or wdd.io/go/OH

Give us anonymous feedback at wdd.io/go/feedback

Box model

content

The picture itself

padding

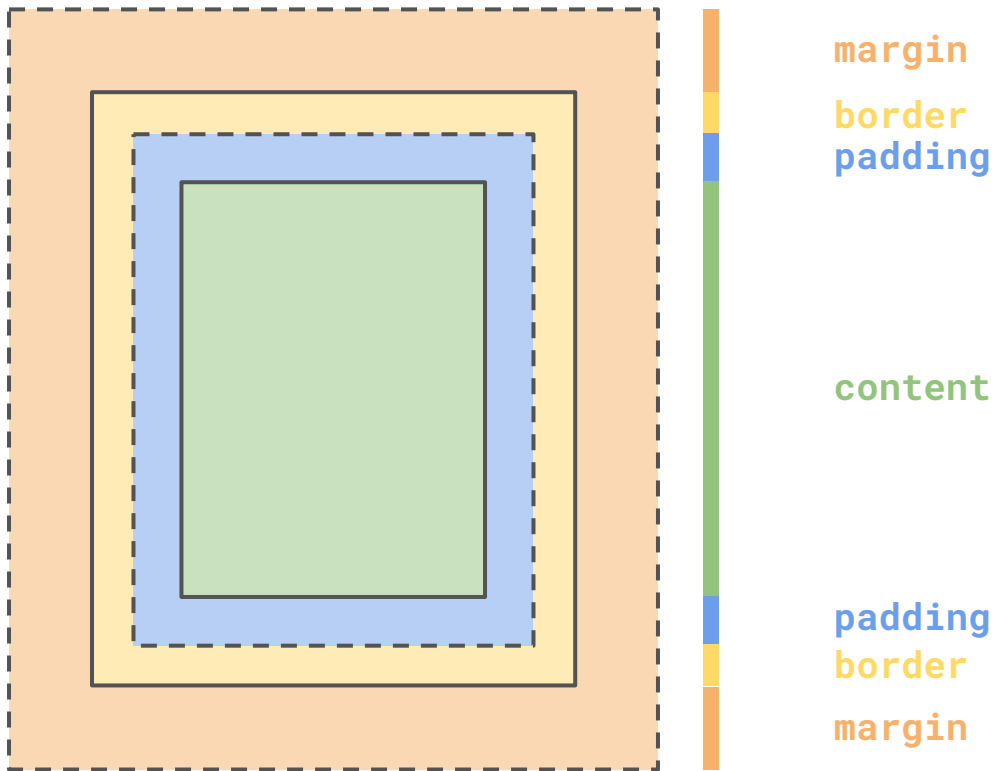
Space between the picture and the picture frame

border

The picture frame

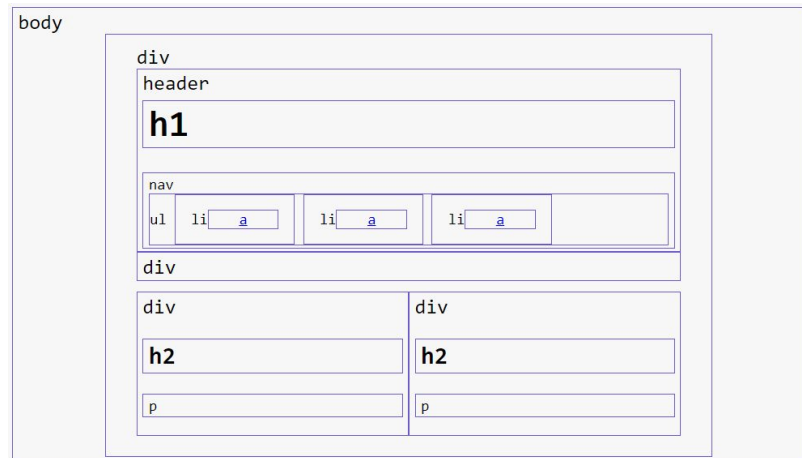
margin

Spacing in between pictures

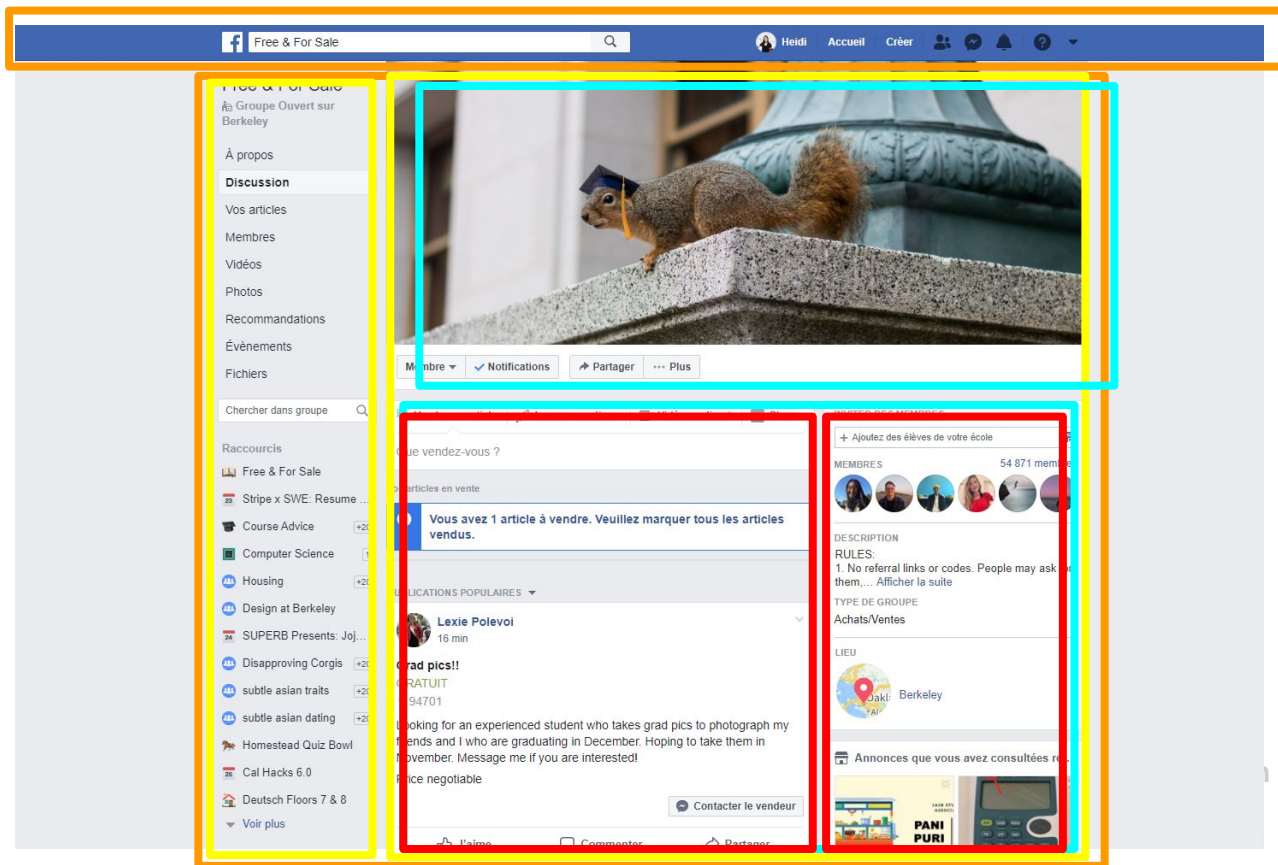


Positioning things
manually is a **pain.**

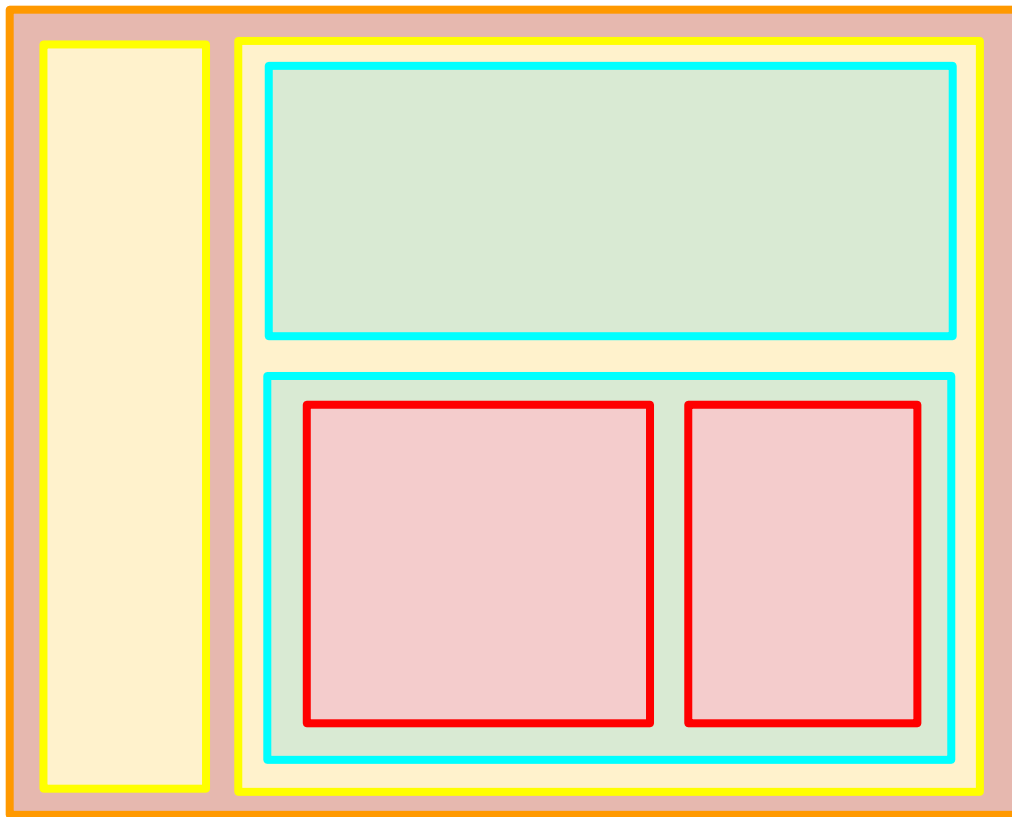
It's important to think of these divs as **containers** in parent-child hierarchies.



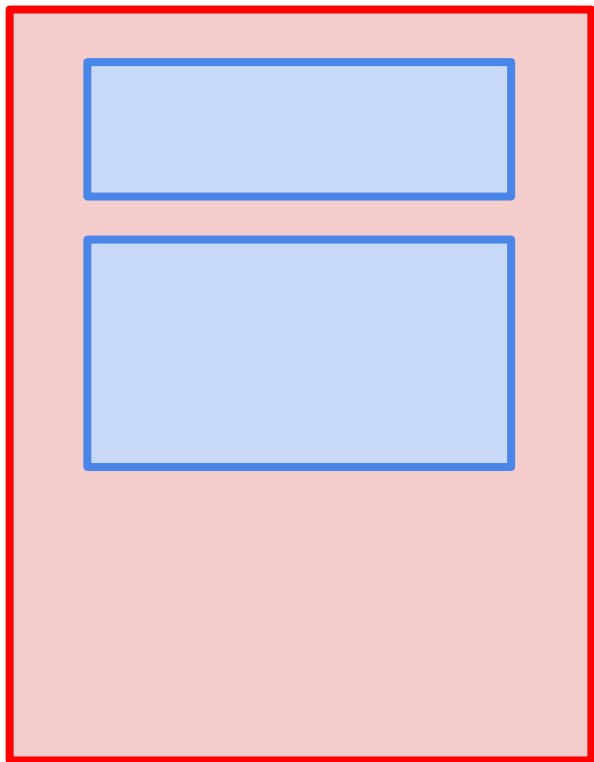
Thinking of a page as building blocks put together



Thinking of a page as building blocks put together



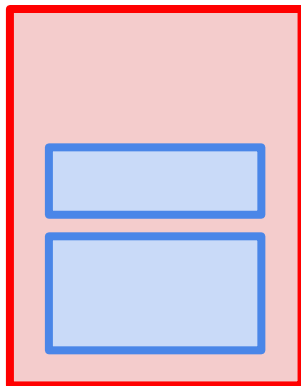
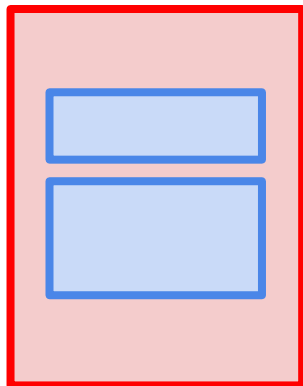
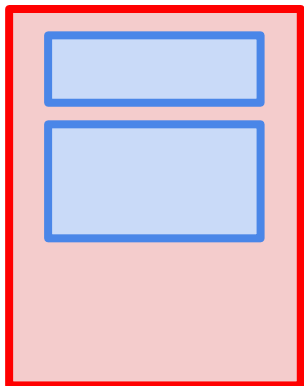
Flexbox



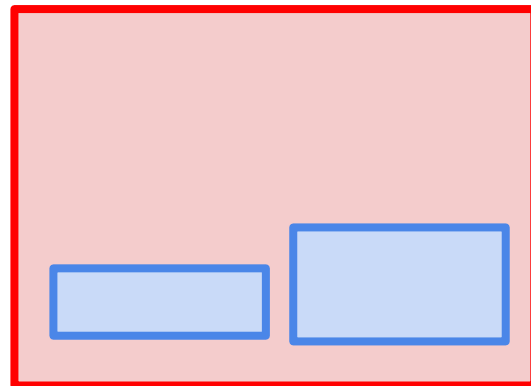
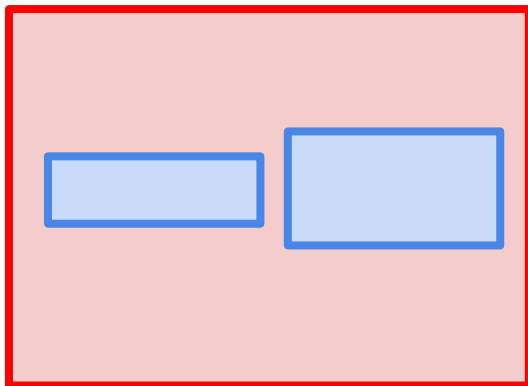
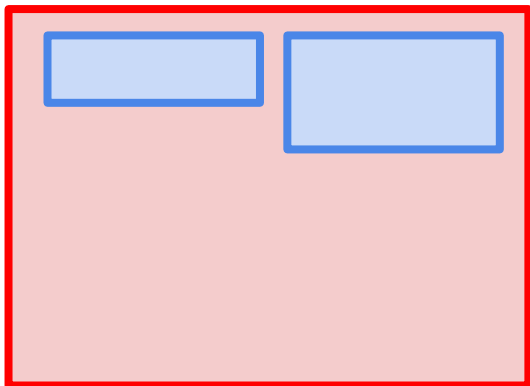
So let's say my HTML is like this..

We need to be able to tell CSS:

- Should the blue boxes be arranged horizontally or vertically?
- How much spacing should be in-between the boxes?
- Should the blue boxes be at the top or bottom of the red box?



How do we make
these different
layouts??



Fleshing out by flexing

```
<div id="best-box">  
  <div class="child">  
    child  
  </div>  
  <div class="child">  
    child  
  </div>  
</div>
```

```
#best-box {  
  display: flex;  
}
```

Specify **display: flex** on the **parent** container.

Allows us to use flexbox css properties to define how the **children** will be aligned and spaced out.

Flex direction: row

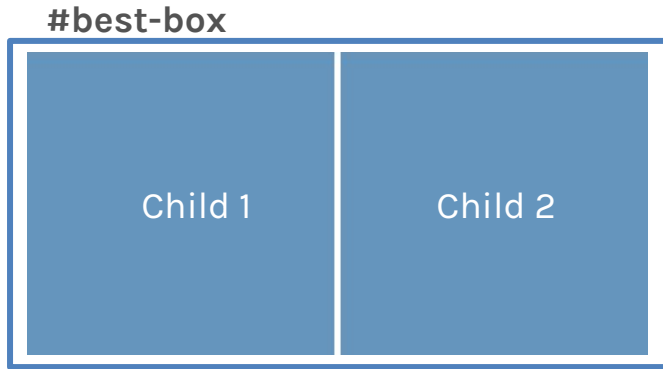
```
<div id="best-box">
  <div class="child">
    Child 1
  </div>
  <div class="child">
    Child 2
  </div>
</div>
```

```
#best-box {
  display: flex;
  flex-direction: row;
}
```

row is the default direction.

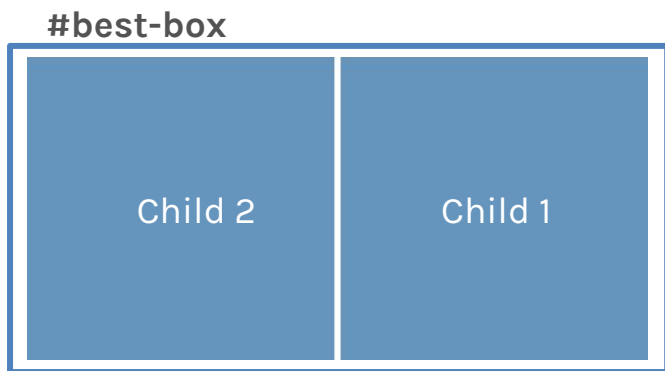
This says, everything inside of this div will be in a row.

Flex direction: row



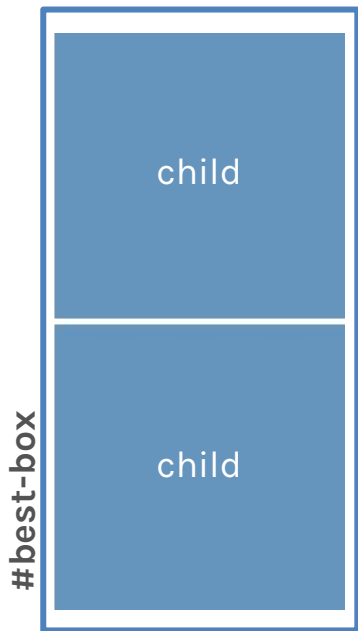
```
#best-box {  
  display: flex;  
  flex-direction: row;  
}
```

Flex direction: row



```
#best-box {  
  display: flex;  
  flex-direction: row-reverse;  
}
```

Flex direction: column



```
#best-box {  
  display: flex;  
  flex-direction: column;  
}
```

There is also
column-reverse

Nested

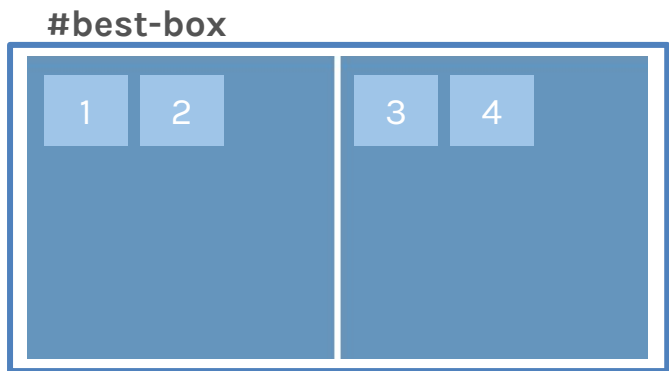
```
<div id="best-box">
  <div class="child">
    <p>1</p>
    <p>2</p>
  </div>
  <div class="child">
    <p>3</p>
    <p>4</p>
  </div>
</div>
```

```
#best-box {
  display: flex;
  flex-direction: row;
}

.child {
  display: flex;
  flex-direction: row;
}
```

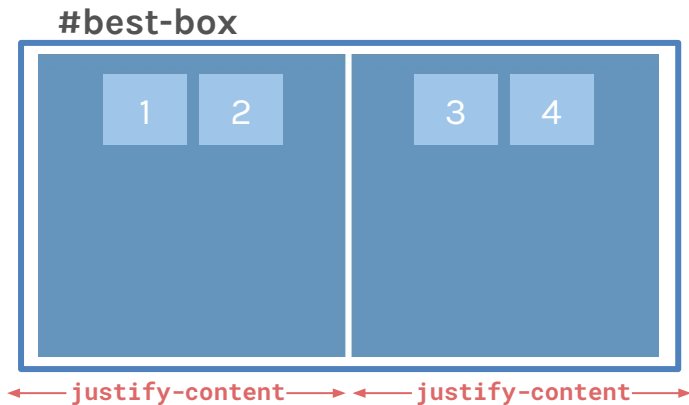
Remember how we saw that layouts can be made up of rows of columns of rows? This is happening!

Nested



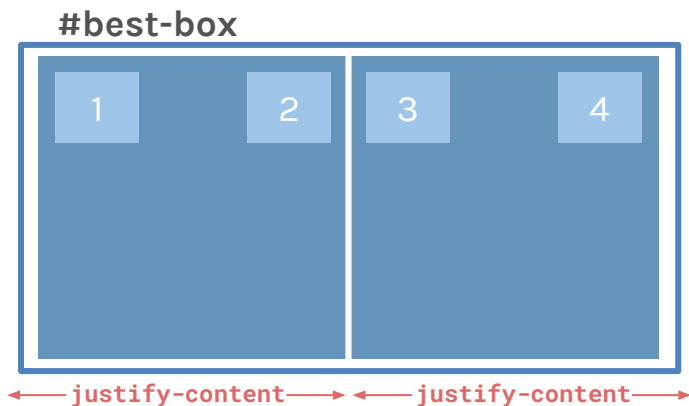
```
#best-box {  
  display: flex;  
  flex-direction: row;  
}  
  
.child {  
  display: flex;  
  flex-direction: row;  
}
```

justify-content possibilities: centering



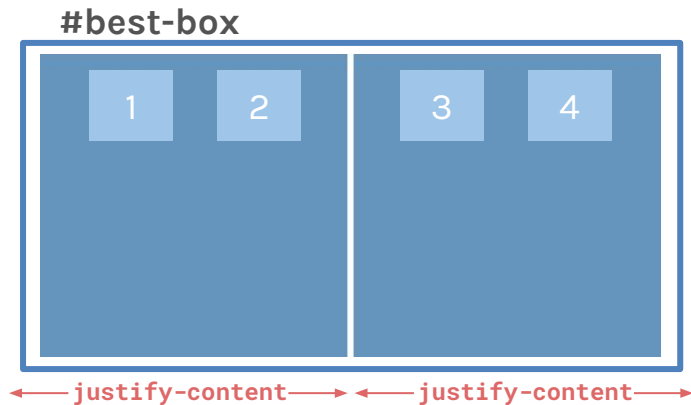
```
#best-box {  
  display: flex;  
  flex-direction: row;  
}  
  
.child {  
  display: flex;  
  flex-direction: row;  
  justify-content: center;  
}
```

justify-content possibilities: space-between



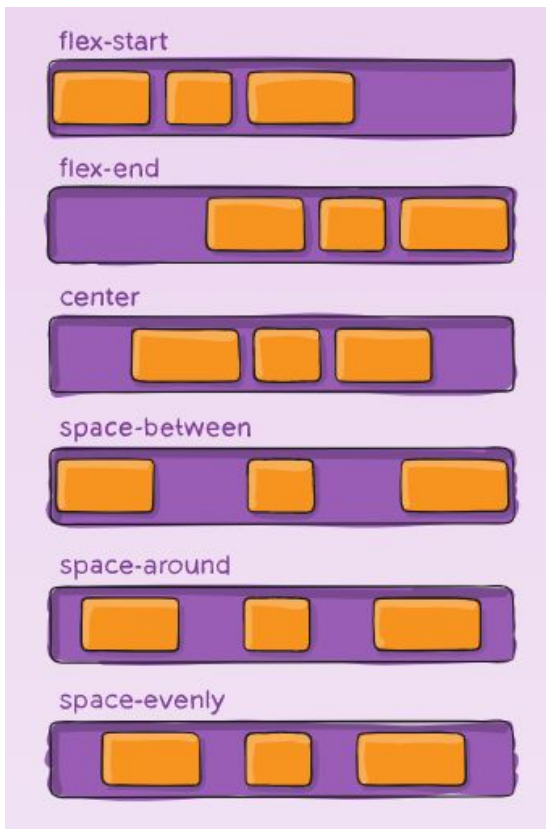
```
#best-box {  
  display: flex;  
  flex-direction: row;  
}  
  
.child {  
  display: flex;  
  flex-direction: row;  
  justify-content: space-between;  
}
```

justify-content possibilities: space-around



```
#best-box {  
  display: flex;  
  flex-direction: row;  
}  
  
.child {  
  display: flex;  
  flex-direction: row;  
  justify-content: space-around;  
}
```

justify-content summary



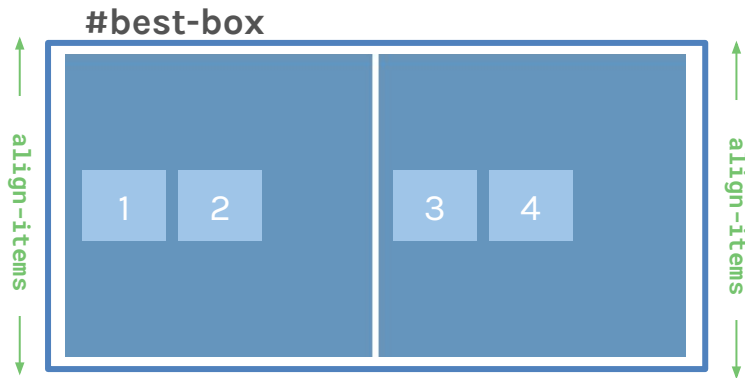
Defines how extra space is distributed along the **main** flex-direction axis!

Image source: CSS Tricks

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

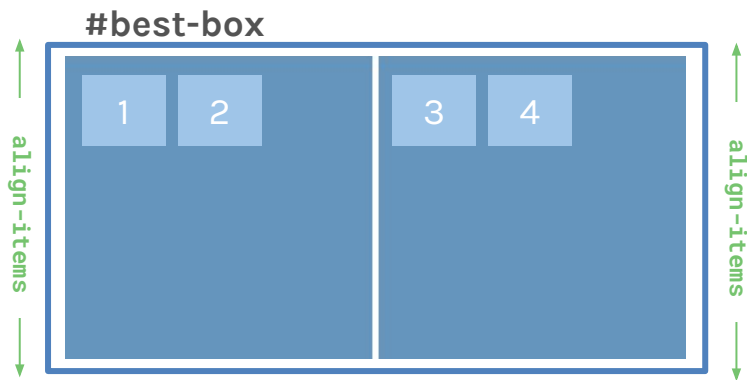
(a great guide to flexbox!!)

align-items possibilities: centering



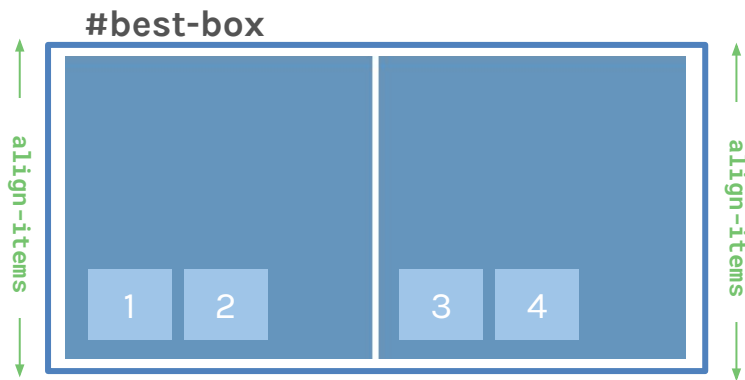
```
#best-box {  
  display: flex;  
  flex-direction: row;  
}  
  
.child {  
  display: flex;  
  flex-direction: row;  
  align-items: center;  
}
```

align-items possibilities: flex-start



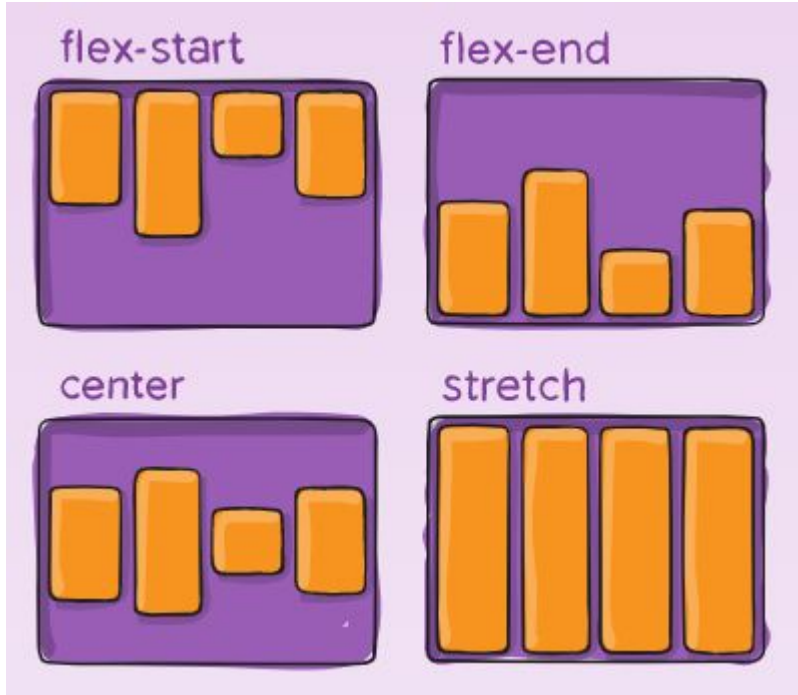
```
#best-box {  
  display: flex;  
  flex-direction: row;  
}  
  
.child {  
  display: flex;  
  flex-direction: row;  
  align-items: flex-start;  
}
```


align-items possibilities: flex-end



```
#best-box {  
  display: flex;  
  flex-direction: row;  
}  
  
.child {  
  display: flex;  
  flex-direction: row;  
  align-items: flex-end;  
}
```

align-items summary

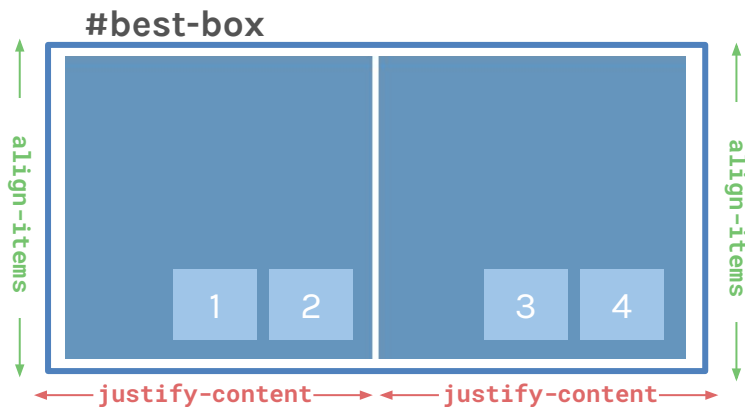


Defines how the children will be aligned along the **cross** axis!

Image source: CSS Tricks

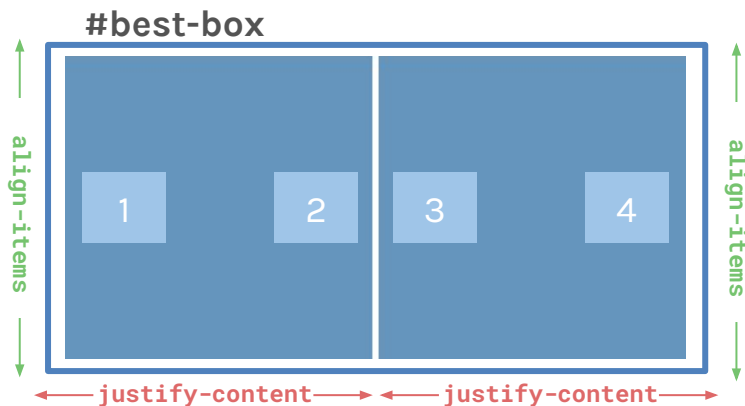
<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
(a great guide to flexbox!!)

justify-content and align-items combined



```
#best-box {  
  display: flex;  
  flex-direction: row;  
}  
  
.child {  
  display: flex;  
  flex-direction: row;  
  justify-content: flex-end;  
  align-items: flex-end;  
}
```

justify-content and align-items combined



```
#best-box {  
  display: flex;  
  flex-direction: row;  
}  
  
.child {  
  display: flex;  
  flex-direction: row;  
  justify-content: space-between;  
  align-items: center;  
}
```

Flexbox axes with `flex-direction: row`



Image modified from

https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout/Basic_Concepts_of_Flexbox

Axes are FLIPPED with **flex-direction: column**

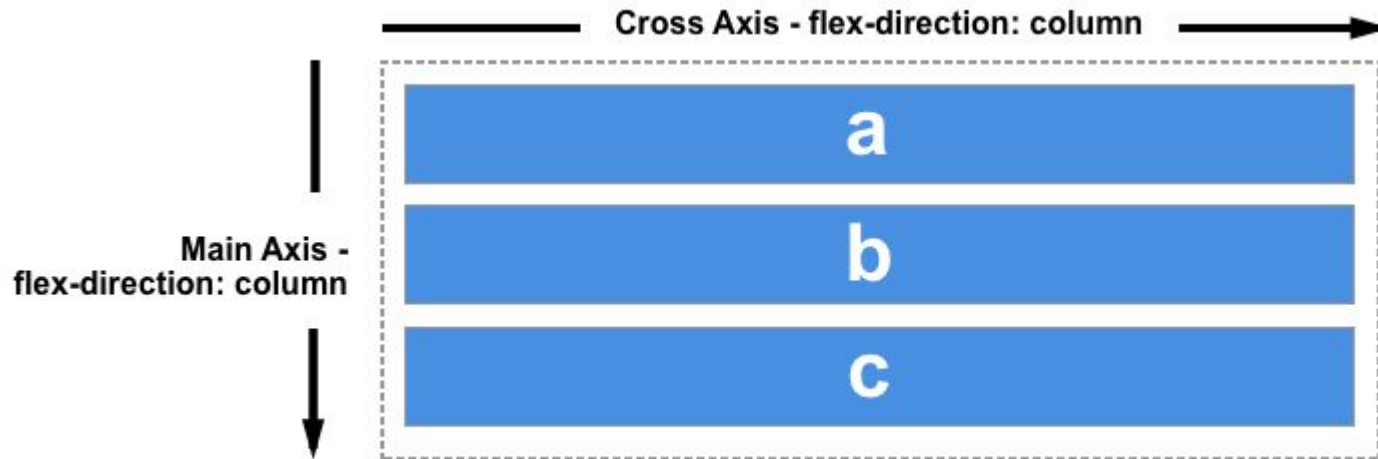


Image modified from

https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout/Basic_Concepts_of_Flexbox

Flexbox Practice

<https://tinyurl.com/wddflexy>



The Lone Warrior

Has invited you to play **Untitled Goose Game**

Accept

Decline



The Lone Warrior

Has invited you to play **Untitled
Goose Game**

Accept

Decline



The Lone Warrior

Has invited you to play **Untitled
Goose Game**

Accept

Decline

Font

16px Times



The Lone Warrior

Has invited you to play **Untitled
Goose Game**

Accept

Decline

Font 16px Times



The Lone Warrior

Has invited you to play **Untitled
Goose Game**

Accept

Decline



The Lone Warrior

Has invited you to play **Untitled**
ame

div.buttons 360 × 60

Accept

Decline



wdd.io/go/flex-demo

wdd.io/go/flex-demo-sol

Flexbox Froggy

<http://flexboxfroggy.com/>



FLEXBOX FROGGY ◀ Level 5 of 24 ▶

Now use **align-items** to help the frogs get to the bottom of the pond. This CSS property aligns items vertically and accepts the following values:

- **flex-start**: Items align to the top of the container.
- **flex-end**: Items align to the bottom of the container.
- **center**: Items align at the vertical center of the container.
- **baseline**: Items display at the baseline of the container.
- **stretch**: Items are stretched to fit the container.

```
1 #pond {
2   display: flex;
3   align-items: flex-end
4 }
5
6
7
8
9
10
```

Submit



Additional links & guides

- <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
- <https://medium.freecodecamp.org/an-animated-guide-to-flexbox-d280cf6afc35>
- <https://giphy.com/gifs/how-flexbox-zfv3Kog26SL04>
- <https://scotch.io/tutorials/a-visual-guide-to-css3-flexbox-properties>

real world applications :ooo



Bianca Lee

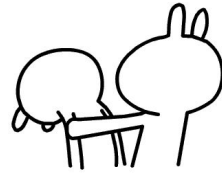
Haotian Ye display: flex; flex-direction: on-friends;

2h

Haha

Reply





Questions?