

Week 5

Positioning & Centering

Positioning

Recall the layouts we learned

Block Formatting Context Inline Formatting Context

Every child has to participate in the layout... But what if...

What if... you want something
knocked out of the **normal flow**?

Cue: Position

We use CSS positioning to make something either or not following the **normal flow**

Yet, what's a "normal flow"?

An example of normal flow

```
em {  
    background-color: lightblue;  
}
```

The name of the city, deriving from Latin forms *Venetia* and *Venetiae*, is most likely taken from "Venetia et Histria", the Roman name of *Regio X* of Roman Italy, but applied to the coastal part of the region that remained under Roman Empire outside of Gothic, Lombard, and Frankish control. The name *Venetia*, however, derives from the Roman name for the people known as the Veneti, and called by the Greeks *Enetoi*. The meaning of the word is uncertain, although there are other Indo-European tribes with similar-sounding names, such as the Celtic Veneti and the Slavic Vistula Veneti. Linguists suggest that the name is based on an Indo-European root **wen* ("love"), so that **wenetoi* would mean "beloved", "lovable", or "friendly". A connection with the Latin word *venetus*, meaning the color 'sea-blue', is also possible. Supposed connections of Venetia with the Latin verb *venire* (to come), such as Marin Sanudo's *veni etiam* ("Yet, I have come!"), the supposed cry of the first refugees to the Venetian lagoon from the mainland, or even with *venia* ("forgiveness") are fanciful.

Say **em** elements are in the normal flow, but offset

```
em {  
  background-color: lightblue;  
  /* New lines added below */  
  position: relative;  
  top: 20px;  
}
```

The name of the city, deriving from Latin forms _____ and _____, is most likely taken from "Venetia et Histria", the **F****venetia** name **venetiae** of Roman Italy, but applied to the coastal part of the region that **Regio X**ed under Roman Empire outside of Gothic, Lombard, and Frankish control. The name _____, however, derives from the Roman name for the people kno**venetia** the Veneti, and called by the Greeks _____. The meaning of the word is uncertain, although there are o**Enetoi**do-European tribes with similar-sounding names, such as the Celtic Veneti and the Slavic Vistula Veneti. Linguists suggest that the name is based on an Indo-European root _____ ("love"), so that _____ would mean "beloved", "lovable", or "fri***wen**". A connection v***wenetoi**.latin word **venetus**, meaning the color 'sea-blue', is also possible. Supposed connections of Venetia with the Latin verb _____ (to come), such as Marin Sanudo's _____ ("Yet, I have co**venire**"), the supposed cry of the first refugees to **veni etiam**an lagoon from the mainland, or even with _____ ("forgiveness") are fanciful. **venia**

A few ways you can **position** an element

static (default)

Position the element according to the normal flow.

relative

Position the element according to the normal flow, and then **offset relative to itself**.

The offset doesn't affect the surrounding elements.

absolute

The element is **pulled out of the normal flow**,
and positioned **relative to its closest positioned ancestor**.

fixed

The element is **pulled out of the normal flow**,
and positioned **relative to the window/viewport**.

A few ways you can **position** an element

static (default)

Position the element according to the normal flow.

relative

Position the element according to the normal flow, and then **offset relative to itself**.

The offset doesn't affect the surrounding elements.

absolute

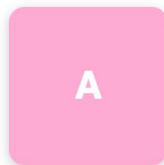
The element is **pulled out of the normal flow**,
and positioned **relative to its closest positioned ancestor**.

fixed

The element is **pulled out of the normal flow**,
and positioned **relative to the window/viewport**.

static

```
.a {  
  position: static;  
}  
.b {  
  position: static;  
}  
.c {  
  position: static;  
}
```



A few ways you can **position** an element

static (default)

Position the element according to the normal flow.

relative

Position the element according to the normal flow, and then **offset relative to itself**.

The offset doesn't affect the surrounding elements.

absolute

The element is **pulled out of the normal flow**,
and positioned **relative to its closest positioned ancestor**.

fixed

The element is **pulled out of the normal flow**,
and positioned **relative to the window/viewport**.

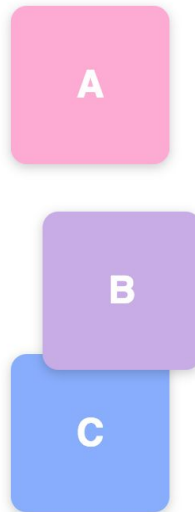
static

```
.a {  
  position: static;  
}  
.b {  
  position: static;  
}  
.c {  
  position: static;  
}
```



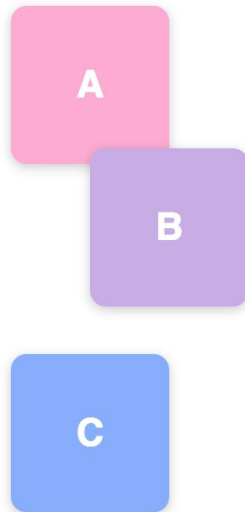
relative

```
.a {  
  position: static;  
}  
.b {  
  position: relative;  
  top: 20px;  
  left: 20px;  
}  
.c {  
  position: static;  
}
```



relative

```
.a {  
  position: static;  
}  
.b {  
  position: relative;  
  left: 50px;  
  bottom: 20px;  
}  
.c {  
  position: static;  
}
```



A few ways you can **position** an element

static (default)

Position the element according to the normal flow.

relative

Position the element according to the normal flow, and then **offset relative to itself**.

The offset doesn't affect the surrounding elements.

absolute

The element is **pulled out of the normal flow**,
and positioned **relative to its closest positioned ancestor**.

fixed

The element is **pulled out of the normal flow**,
and positioned **relative to the window/viewport**.

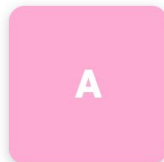
static

```
.a {  
  position: static;  
}  
.b {  
  position: static;  
}  
.c {  
  position: static;  
}
```



absolute

```
.a {  
  position: static;  
}  
.b {  
  position: absolute;  
  top: 0; /* Top margin edge */  
  right: 0; /* Right margin edge */  
}  
.c {  
  position: static;  
}
```



A few ways you can **position** an element

static (default)

Position the element according to the normal flow.

relative

Position the element according to the normal flow, and then **offset relative to itself**.

The offset doesn't affect the surrounding elements.

absolute

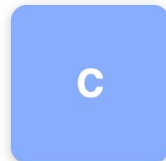
The element is **pulled out of the normal flow**,
and positioned **relative to its closest positioned ancestor**.

fixed

The element is **pulled out of the normal flow**,
and positioned **relative to the window/viewport**.

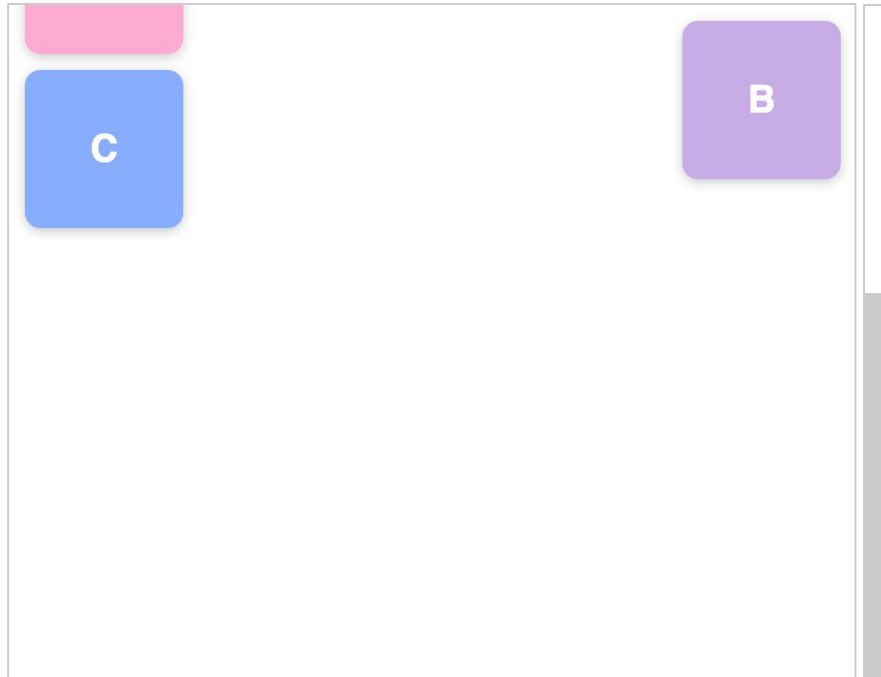
fixed

```
.a {  
  position: static;  
}  
.b {  
  position: fixed;  
  top: 0;  
  right: 0;  
}  
.c {  
  position: static;  
}
```



fixed

```
.a {  
  position: static;  
}  
.b {  
  position: fixed;  
  top: 0;  
  right: 0;  
}  
.c {  
  position: static;  
}
```



A few ways you can **position** an element

static (default)

Position the element according to the normal flow.

relative

Position the element according to the normal flow, and then **offset relative to itself**.

The offset doesn't affect the surrounding elements.

absolute

The element is **pulled out of the normal flow**,

and positioned **relative to its closest positioned ancestor** (padding-box).

fixed

The element is **pulled out of the normal flow**,

and positioned **relative to the window/viewport**.

Centering

To center things in a **block/inline** layout...

To center things in a **block/inline** layout...

Horizontally center **block** elements

Horizontally center **inline** elements

Vertically center **block** elements

Vertically center **inline** elements

To center things in a **block/inline** layout...

Horizontally center **block** elements

Horizontally center inline elements

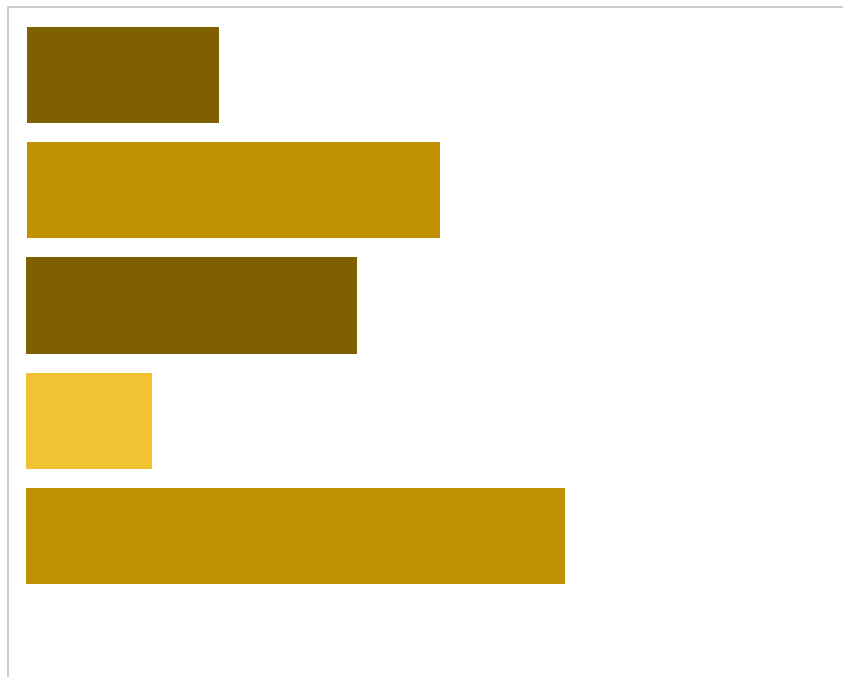
Vertically center block elements

Vertically center inline elements

Horizontally center **block** elements

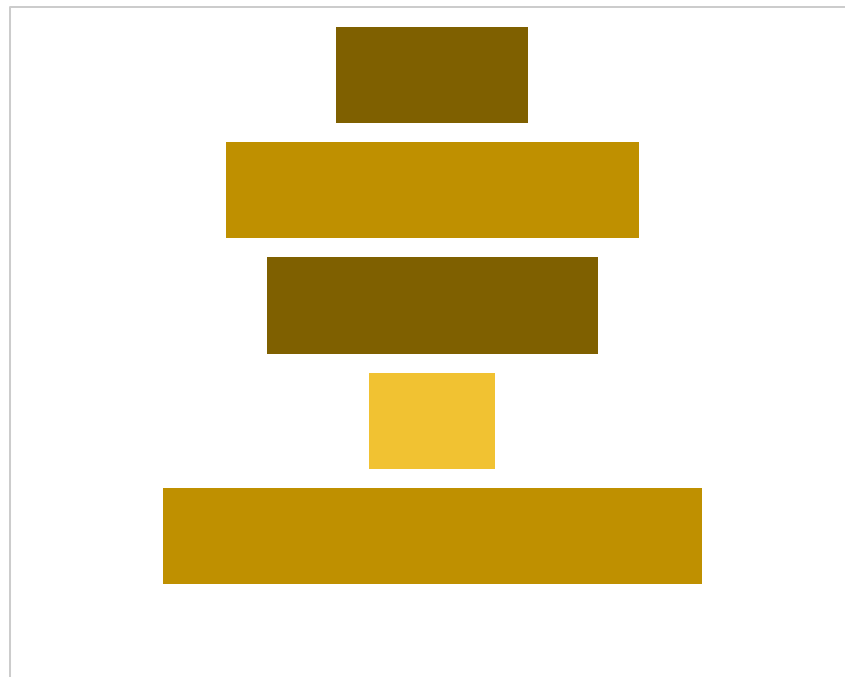
```
.potato {
```

```
}
```



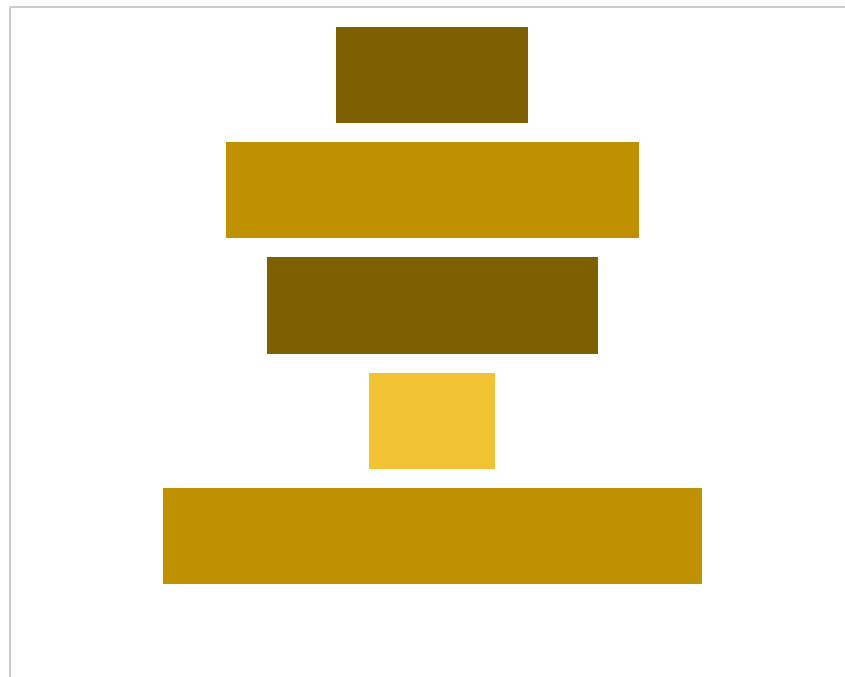
Horizontally center **block** elements

```
.potato {  
  margin-right: auto;  
  margin-left: auto;  
}
```



Horizontally center **block** elements

```
.potato {  
  margin: 0 auto;  
}
```



To center things in a **block/inline** layout...

Horizontally center block elements

Horizontally center **inline** elements

Vertically center block elements

Vertically center inline elements

Horizontally center inline elements

```
.small-potato {
```

```
}
```



Horizontally center inline elements

```
.farm {  
    text-align: center;  
}  
  
.small-potato {  
  
}
```





Questions?

To center things in a **block/inline** layout...

Horizontally center block elements

Horizontally center inline elements

Vertically center **block** elements

Vertically center inline elements

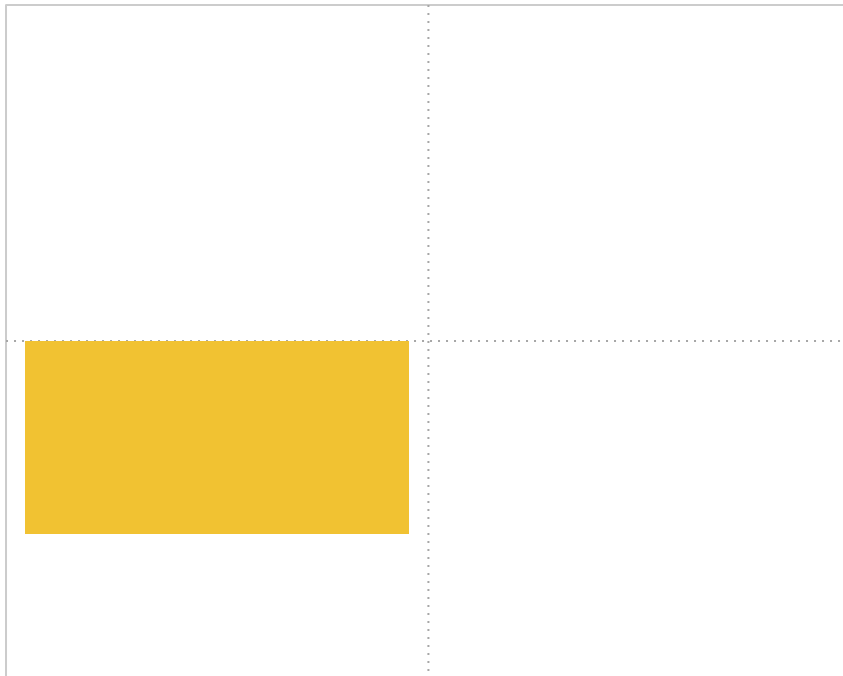
Vertically center **block** elements w/ **absolute** position

```
#big-potato {  
  width: 100px;  
  height: 50px;  
}
```



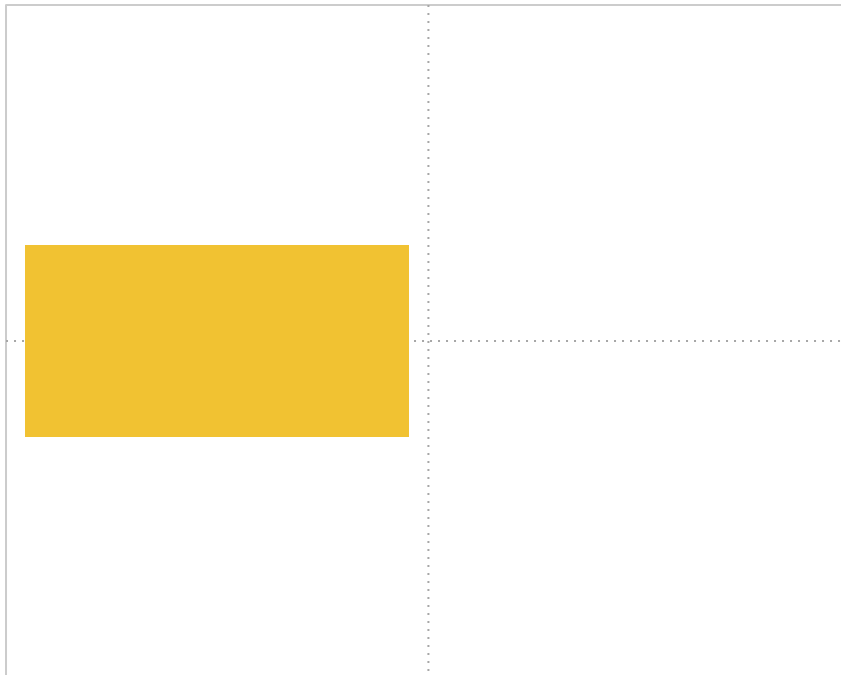
Vertically center **block** elements w/ **absolute** position

```
#big-potato {  
  width: 100px;  
  height: 50px;  
  position: absolute;  
  top: 50%;  
}
```



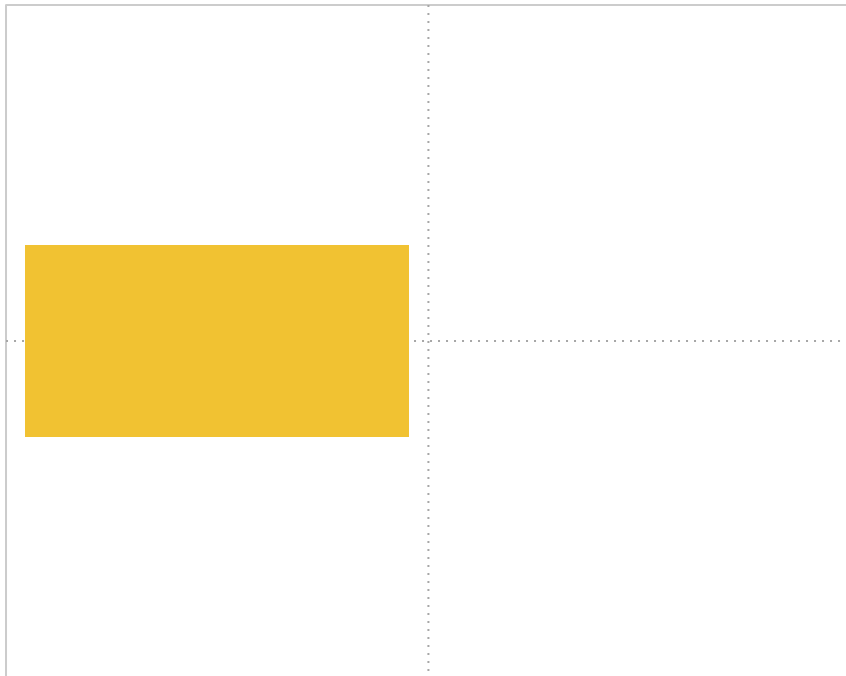
Vertically center **block** elements w/ **absolute** position

```
#big-potato {  
  width: 100px;  
  height: 50px;  
  position: absolute;  
  top: 50%;  
  margin-top: -25px;  
}
```



Vertically center **block** elements w/ **absolute** position

```
#big-potato {  
  width: 100px;  
  height: 50px;  
  position: absolute;  
  top: 50%;  
  transform: translateY(-50%);  
}
```

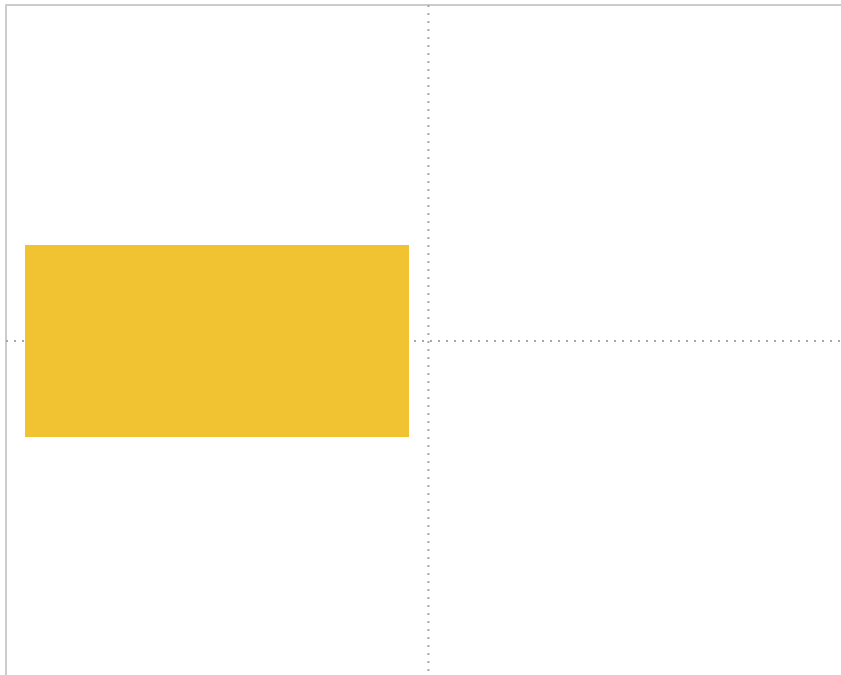


What's the 50% doing in the transform?

We can center-center things too

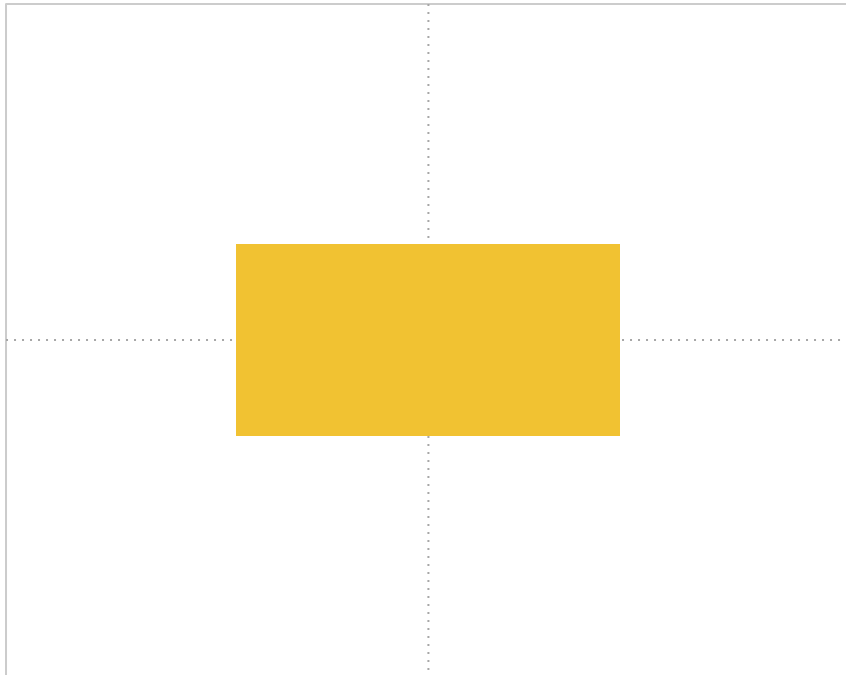
Vertically center **block** elements w/ **absolute** position

```
#big-potato {  
  width: 100px;  
  height: 50px;  
  position: absolute;  
  top: 50%;  
  margin-top: -25px;  
}
```



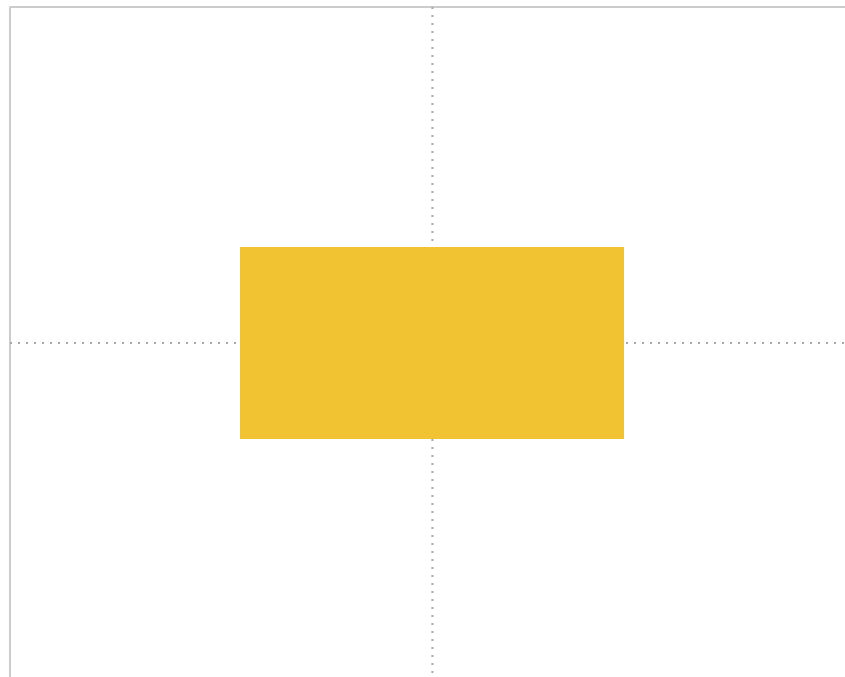
Center center **block** elements

```
#big-potato {  
  width: 100px;  
  height: 50px;  
  position: absolute;  
  top: 50%;  
  left: 50%;  
  margin-top: -25px;  
  margin-left: -50px;  
}
```



Center center **block** elements

```
#big-potato {  
  width: 100px;  
  height: 50px;  
  position: absolute;  
  top: 50%;  
  left: 50%;  
  transform: translate(-50%, -50%);  
}
```



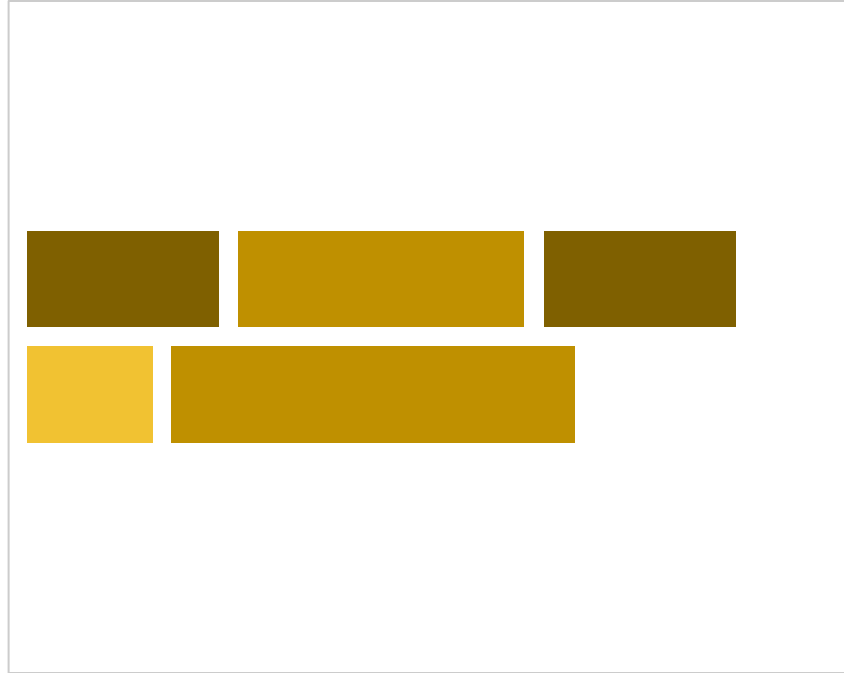
To center things in a **block/inline** layout...

Horizontally center block elements

Horizontally center inline elements

Vertically center block elements

Vertically center **inline** elements



Like this... **Well it's not that easy to do**

Web Design DeCal Spring 2019

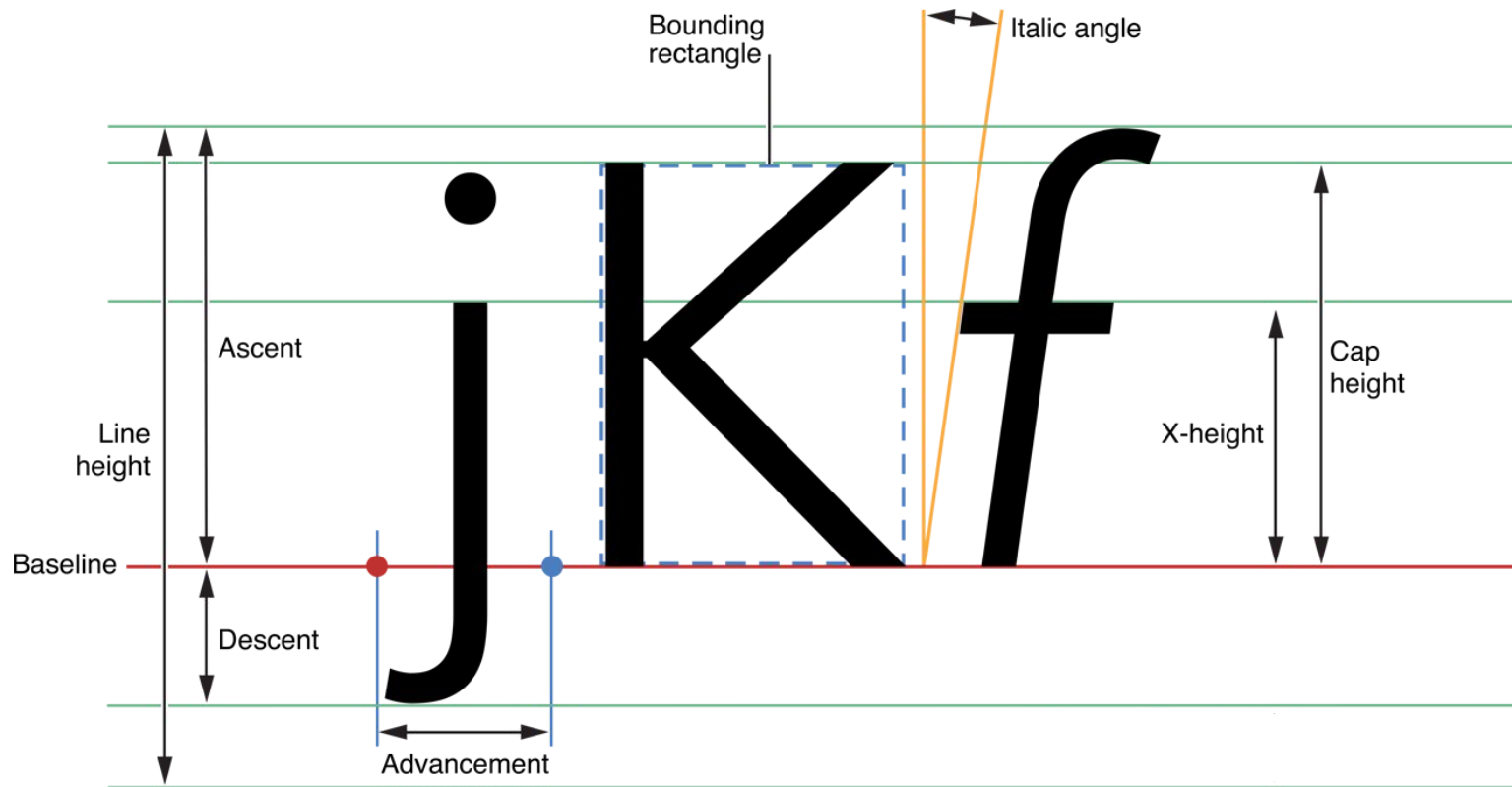
However, you can position inline elements within their line box!
Let's start with a questionably motivational question

**"Sometimes I find some
blank space below my image?!"**



Example: <https://codepen.io/sethlu/pen/pYErMe>

Web Design DeCal Spring 2019



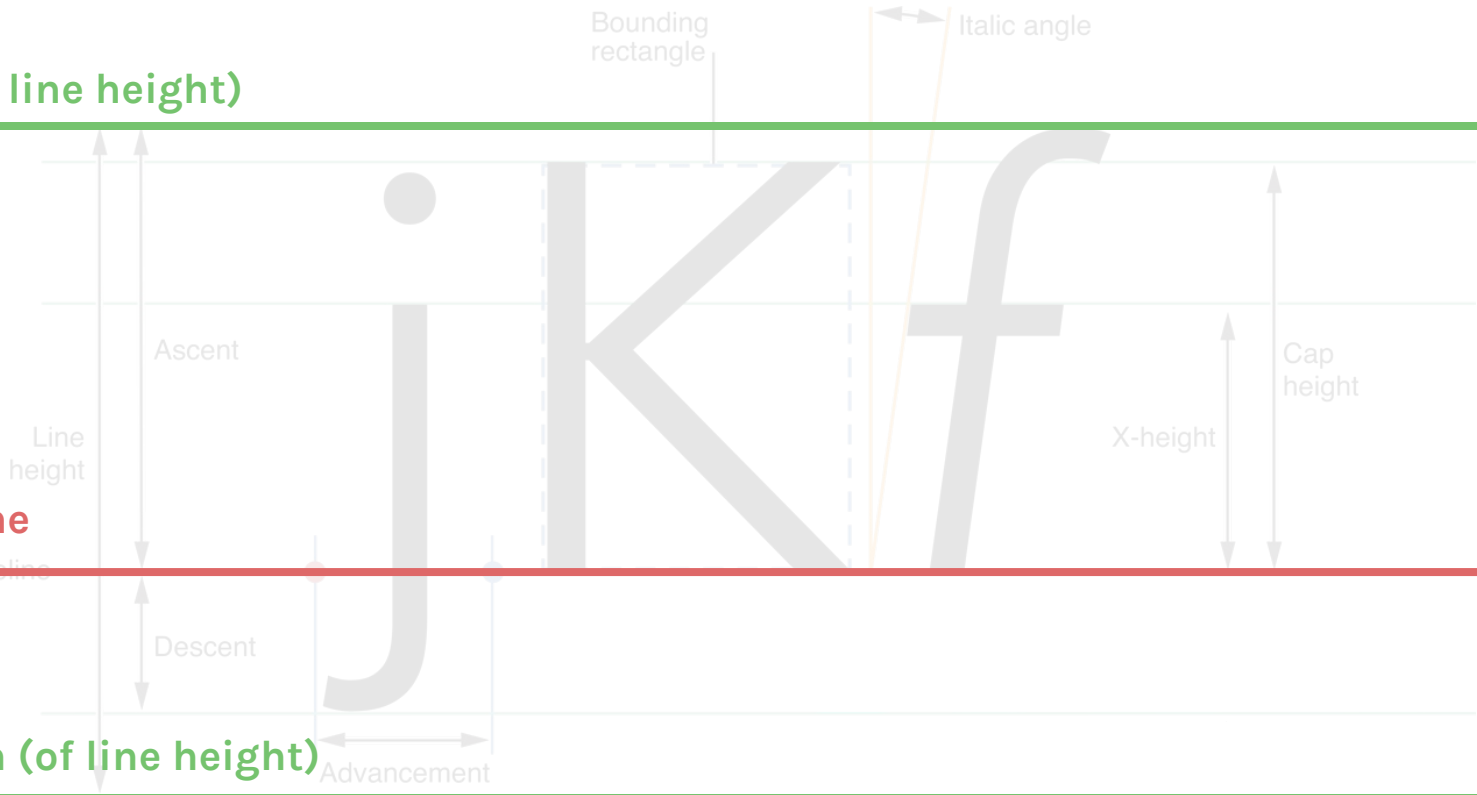
Some font metrics (this is a lot... we don't expect you to know all of it)

Web Design DeCal Spring 2019

Top (of line height)

Baseline

Bottom (of line height)



By default, inline elements are placed on the baseline

Web Design DeCal Spring 2019

Bottom (of line height)

If there's space between the baseline and the bottom of the line height, then there won't be blank space

If we don't fix the blank space, it will grow with font size or with line height

Baseline

Bottom (of line height)



And now we can work around the issue!

Either place the image at the bottom...
Or display the image as a block element
(then no need to care about the vertical aligning)



Questions?