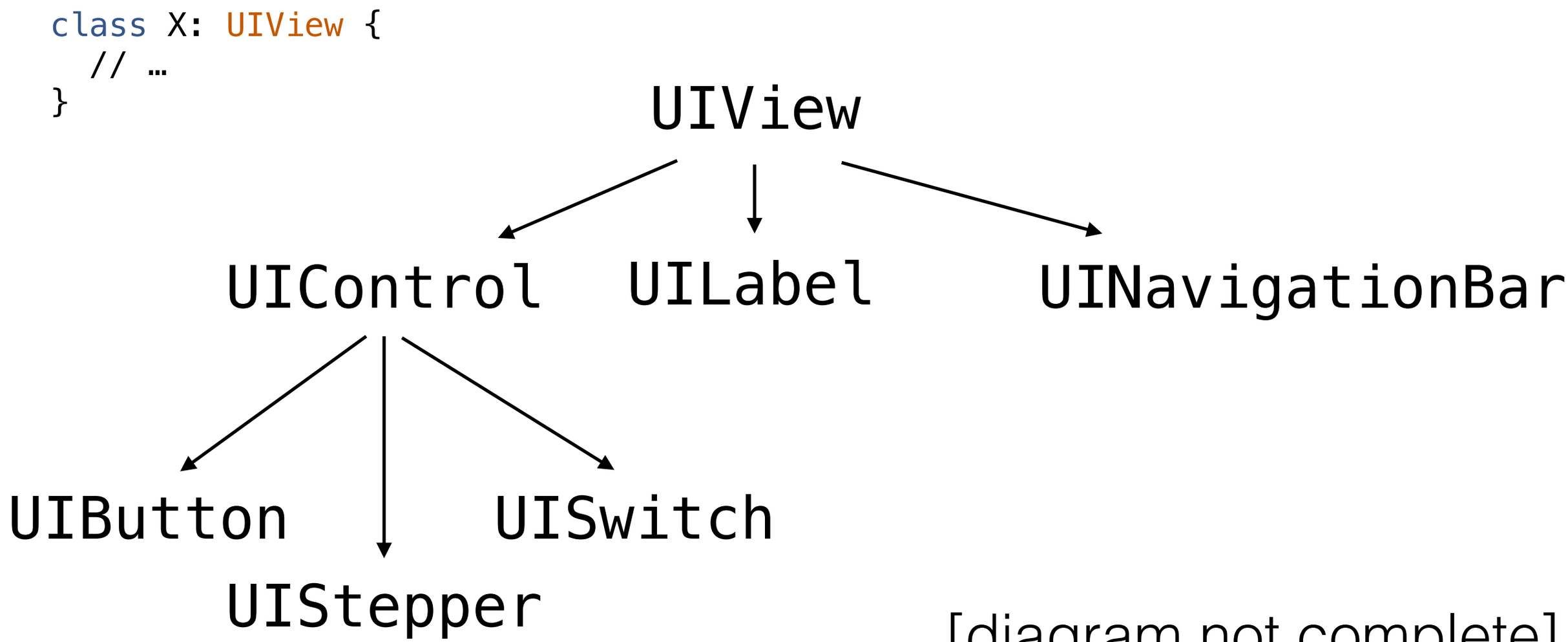
Basic `UIKit`

Mastering positioning and sizing of `UIViews`

UIView

`UIView` is the most fundamental base class in **`UIKit`**, all other UI components *inherit* from it -
i.e. everything you *see* on the screen ultimately *is* a **`UIView`**

```
class X: UIView {
  // …
}
```



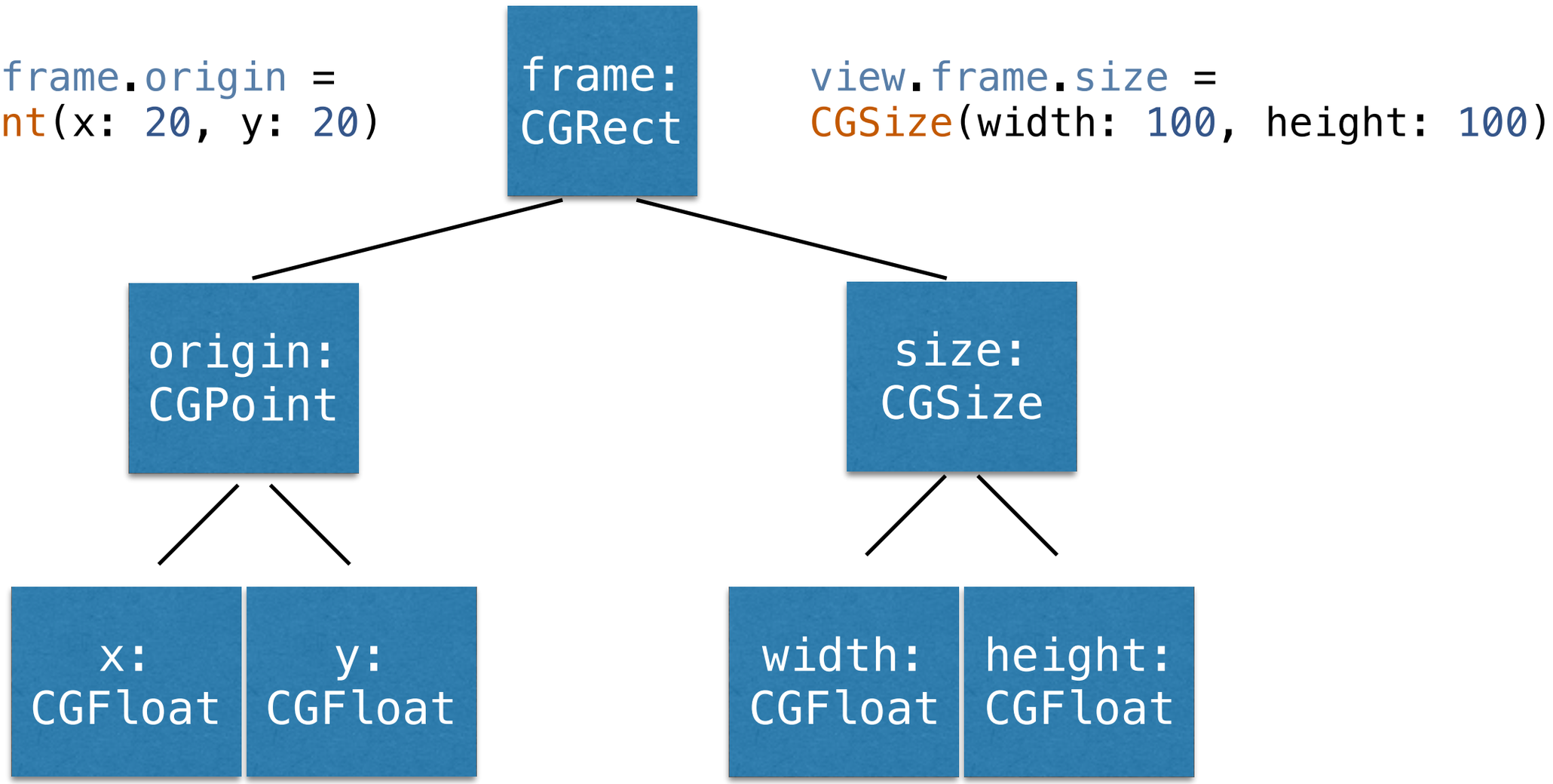
[diagram not complete]

Anatomy of a `UIView`

```
view.frame = CGRect(x: 20, y: 20, width: 20, height: 20)
```

```
view.frame.origin =
CGPoint(x: 20, y: 20)
```

```
view.frame.size =
CGSize(width: 100, height: 100)
```




```
view.frame.origin.x = 20
view.frame.origin.y = 20
```

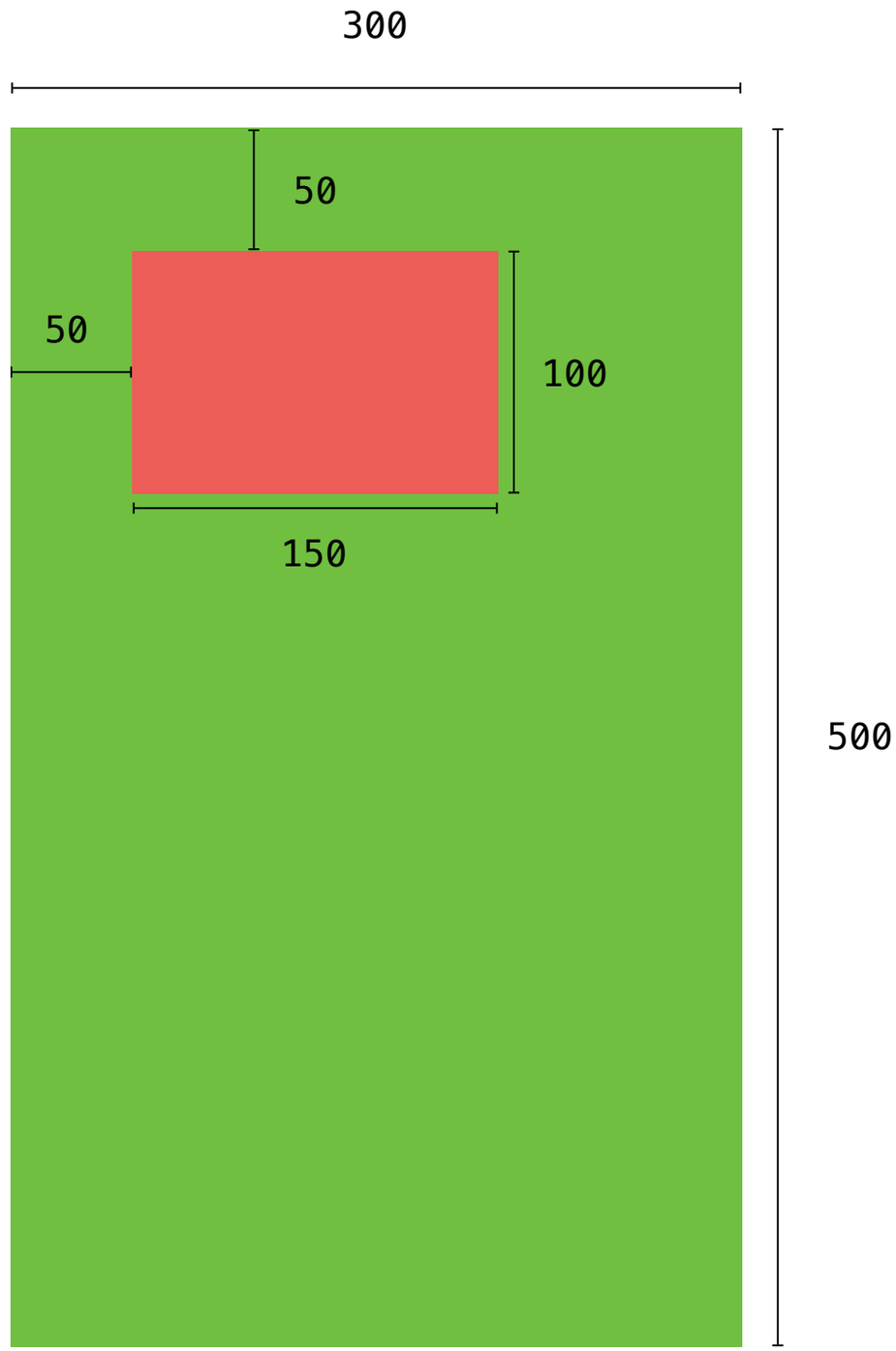
```
view.frame.size.width = 100
view.frame.size.height = 100
```

Positioning

the positioning of a **`UIView`** is determined by its **`origin`** which is *relative* to the superview

```
redView.frame.origin = CGPoint(x: 50, y: 50)
redView.frame.size   = CGSize(width: 150, height: 100)


greenView.frame.origin = CGPoint(x: 0, y: 0)
greenView.frame.size   = CGSize(width: 300, height: 500)
```

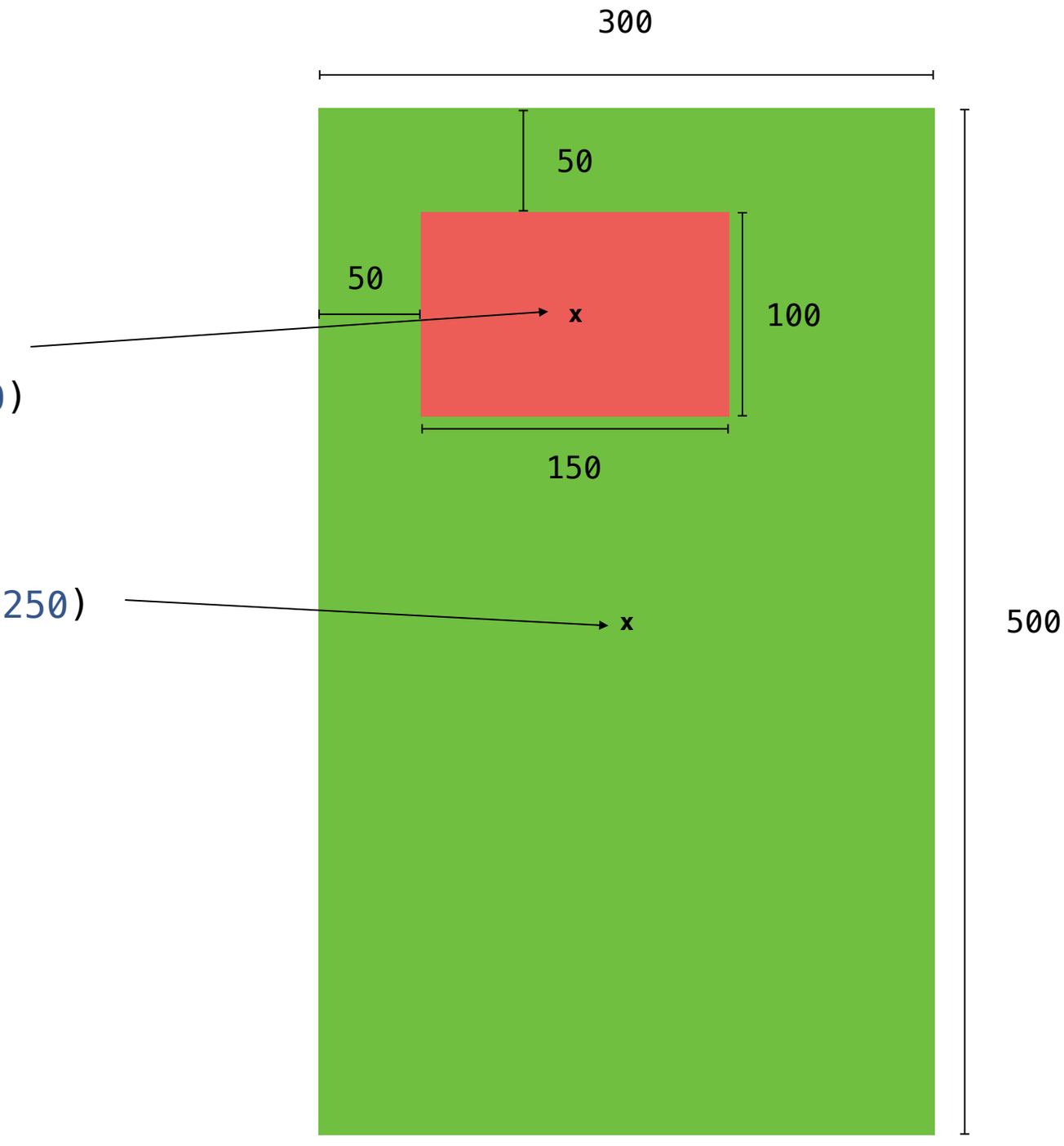


center

a `UIView` also has a property called `center`, which is *relative* to its superview as well

```
redView.center = CGPoint(x: 75, y: 50)
redView.center = CGPoint(x: 125, y: 100)
```

```
greenView.center = CGPoint(x: 150, y: 250)
```



note that the `center` can always be derived from the `frame`, but not the other way around!

AutoLayout

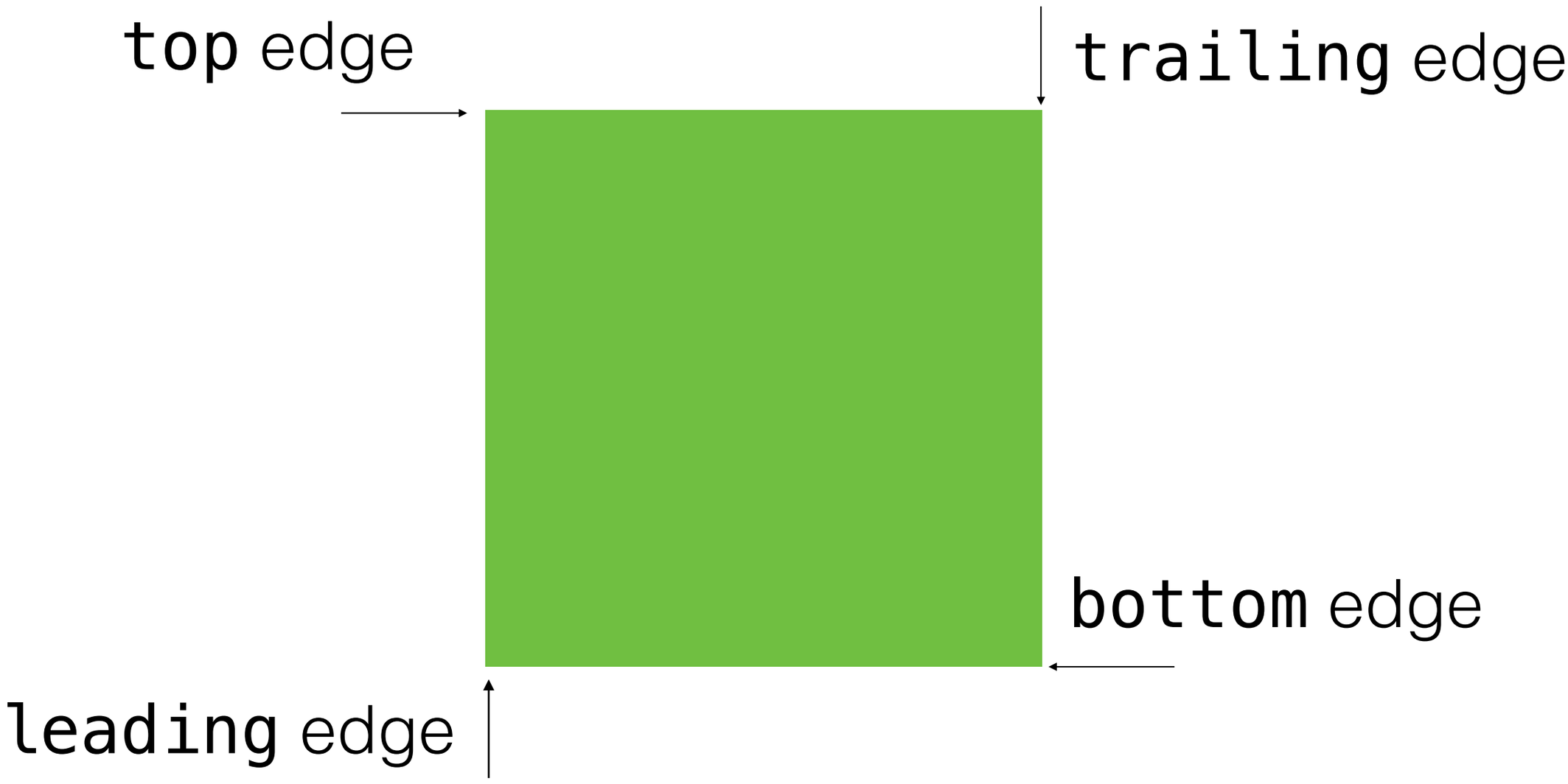
don't set `frame` directly

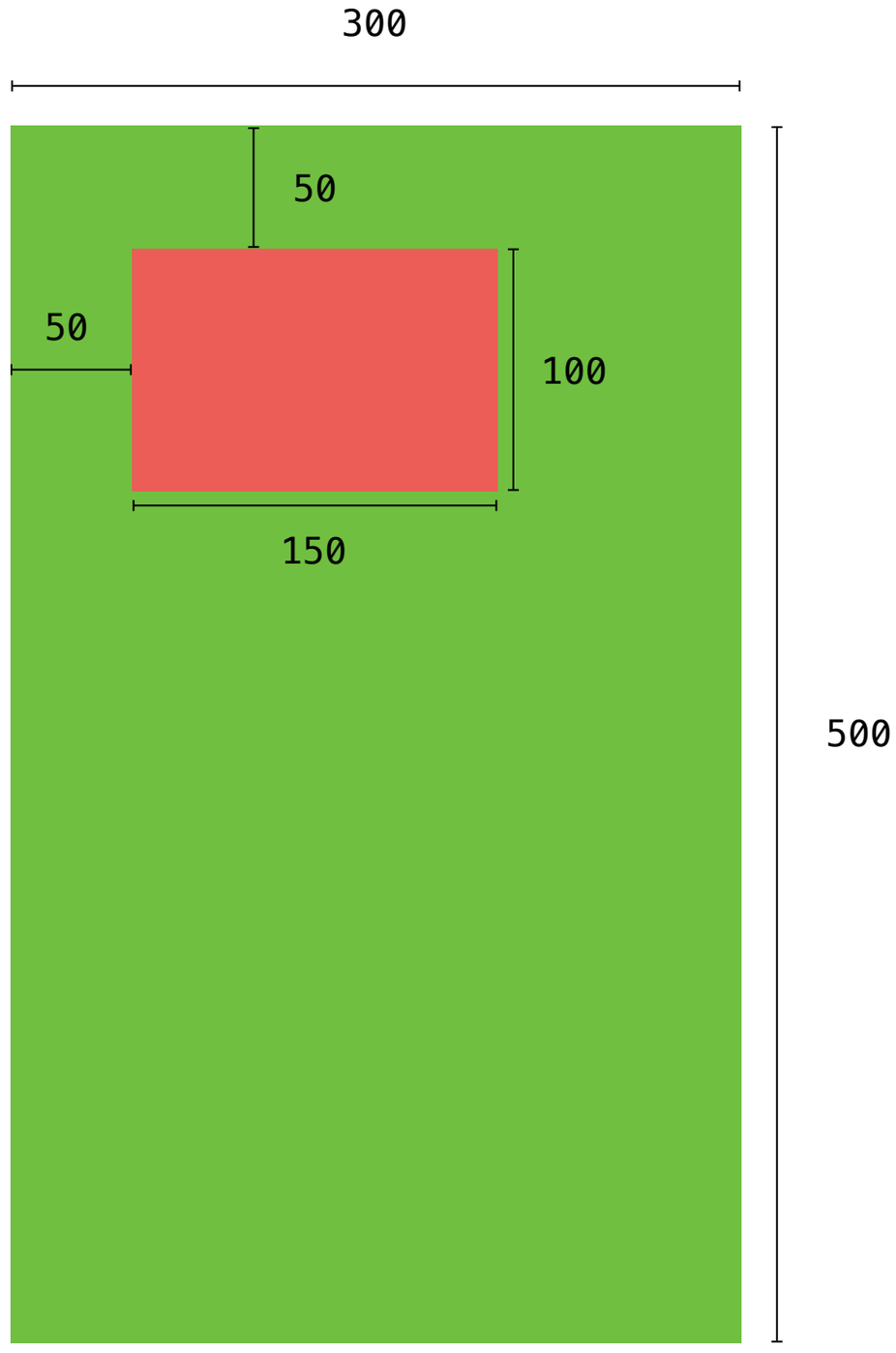
specify a set of rules (*constraints*) and let AutoLayout calculate the actual `frame` values

gives us a simple way to specify size and positioning of `UIView`s relative to each other

AutoLayout requires us to specify enough constraints so it can calculate the `frame` for each view (i.e. `x`, `y`, `width` and `height`)

Layout Guides




300
50
50
100
150
500