

# AutoLayout

Mastering positioning and sizing of **UIViews** using  
AutoLayout constraints

# AutoLayout

don't set **frame** directly

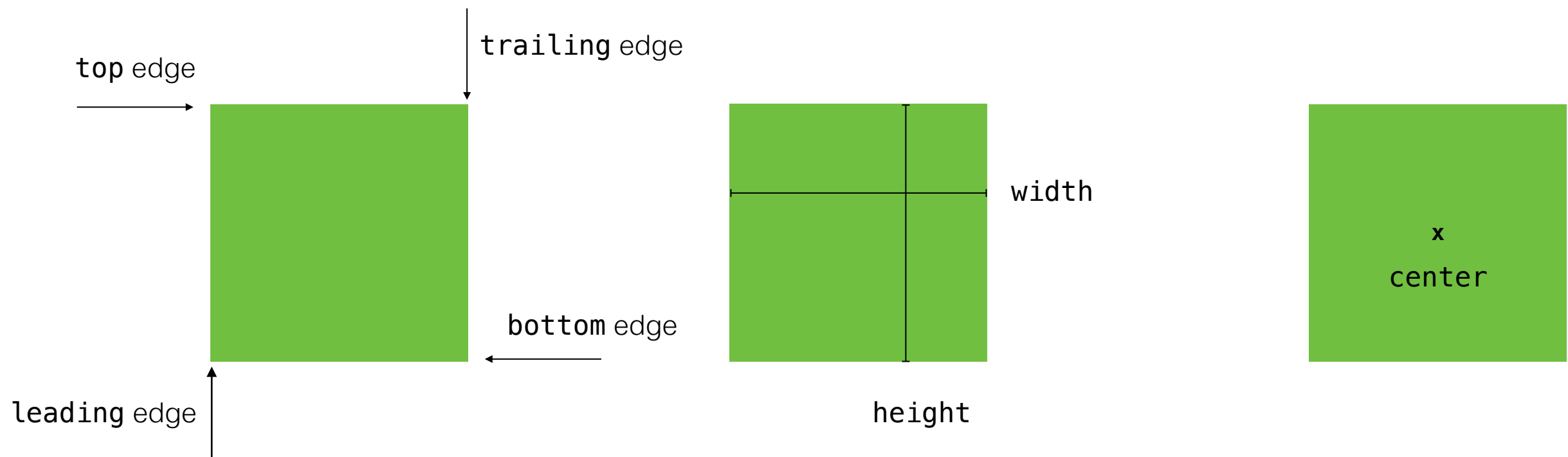
specify a set of rules (*constraints*) and let AutoLayout calculate the actual **frame** values

gives us a simple way to specify size and positioning of **UIViews** 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**)

# How to define constraints?

## AutoLayout *Attributes*



# Setting AutoLayout Constraints

## Intuition:

The `leading` edge of the `redView` is has a 50 point distance to the `leading` edge of the `greenView`.

## Pseudocode:

```
redView.leadingEdge = greenView.leadingEdge + 50
```

## Actual code 🤖:

```
NSLayoutConstraint(  
  item: redView, attribute: .leading,  
  relatedBy: .equal,  
  toItem: greenView, attribute: .leading,  
  multiplier: 1, constant: +50)
```

## Intuition:

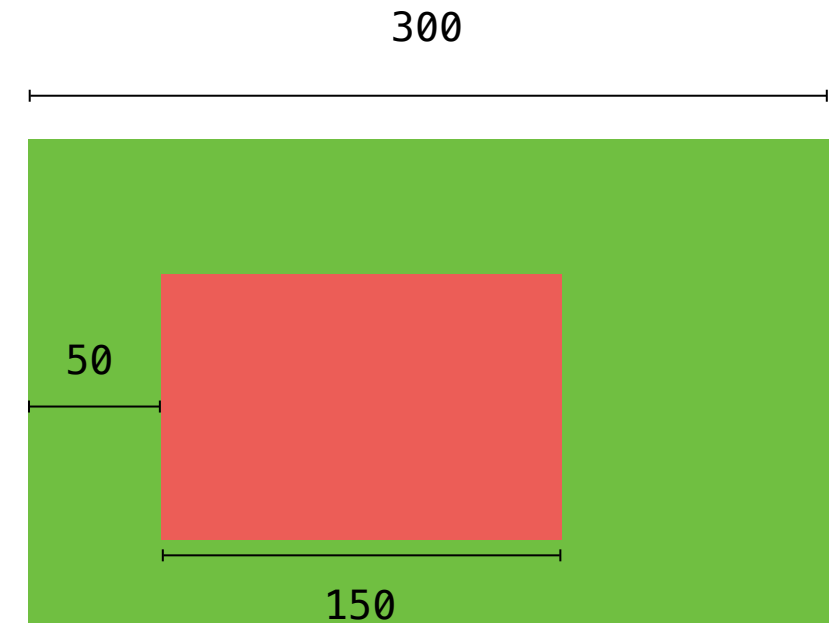
The width of the `redView` is half as long as the width of the `greenView`.

## Pseudocode:

```
redView.width = greenView.width * 0.5
```

## Actual code 🤖:

```
NSLayoutConstraint(  
  item: redView, attribute: .width,  
  relatedBy: .equal,  
  toItem: greenView, attribute: .width,  
  multiplier: 0.5, constant: 0)
```



Note that this example is **incomplete**: With the current two constraints, AutoLayout would complain because it can not compute the full **frame** (i.e. **x**, **y**, **width** and **height**) for the `redView`. We only told it how to compute **x** and **width**.