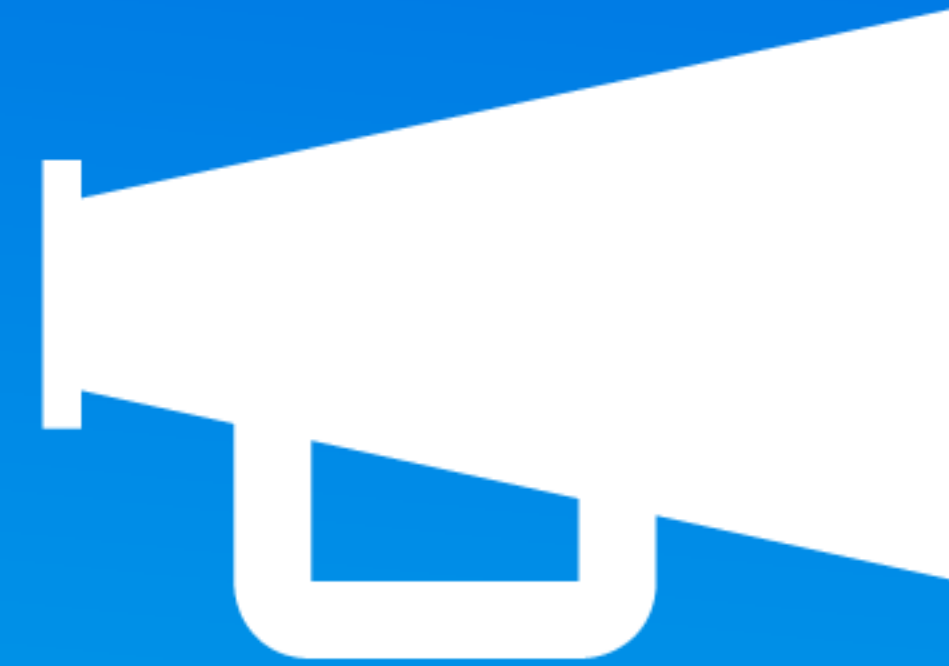
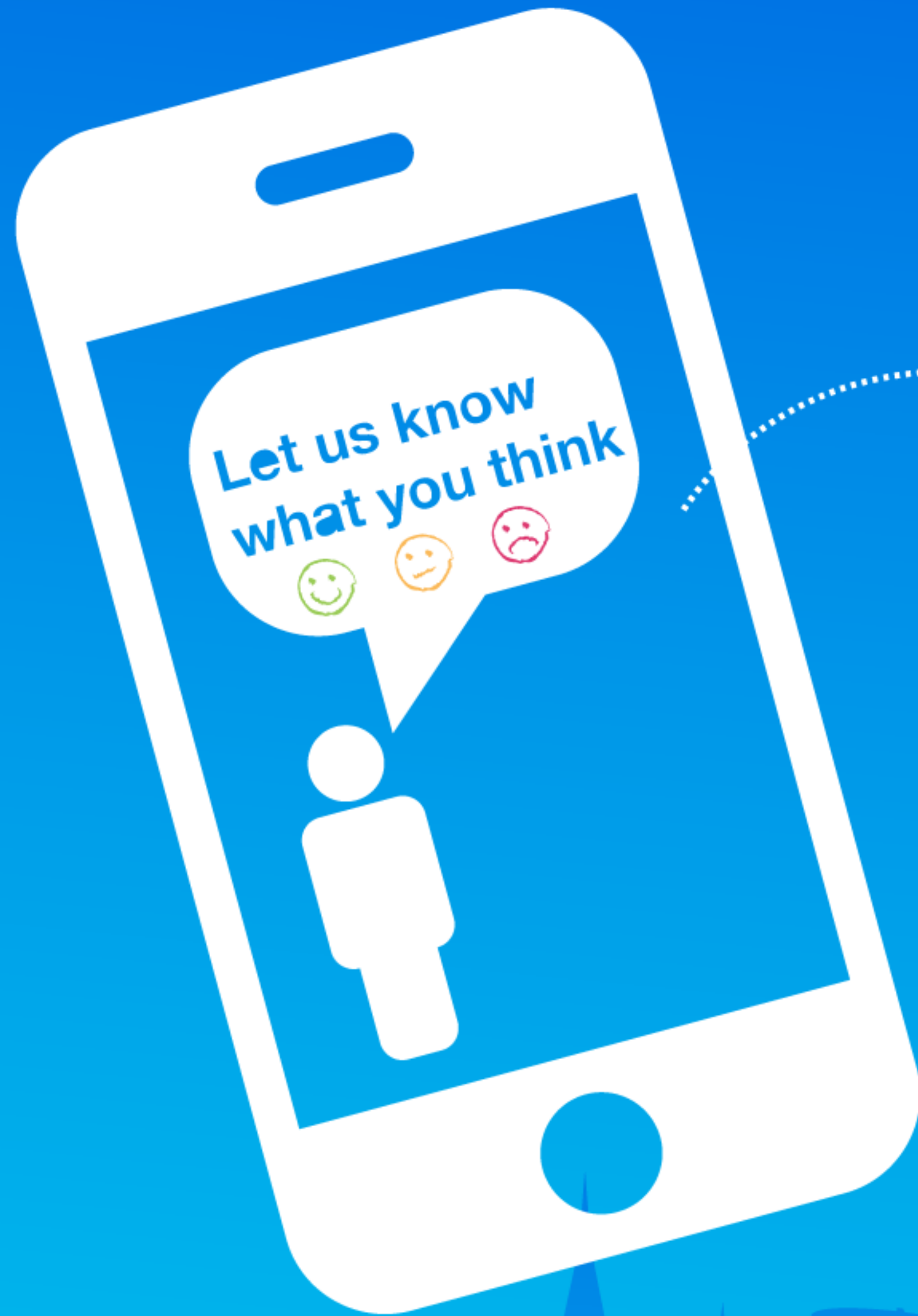


Adaptive UI

Sam Davies

@iwantmyrealname





Click 'engage'
to rate sessions
and ask questions



A close-up photograph of a man with short brown hair, wearing red-rimmed glasses. He has a wide-eyed, excited expression with his mouth open in a smile, showing his teeth. The background is a blurred indoor setting with warm lighting. The text 'i'm sam' is overlaid in the top left corner in a white, lowercase, serif font.

i'm sam

@iwantmyrealname

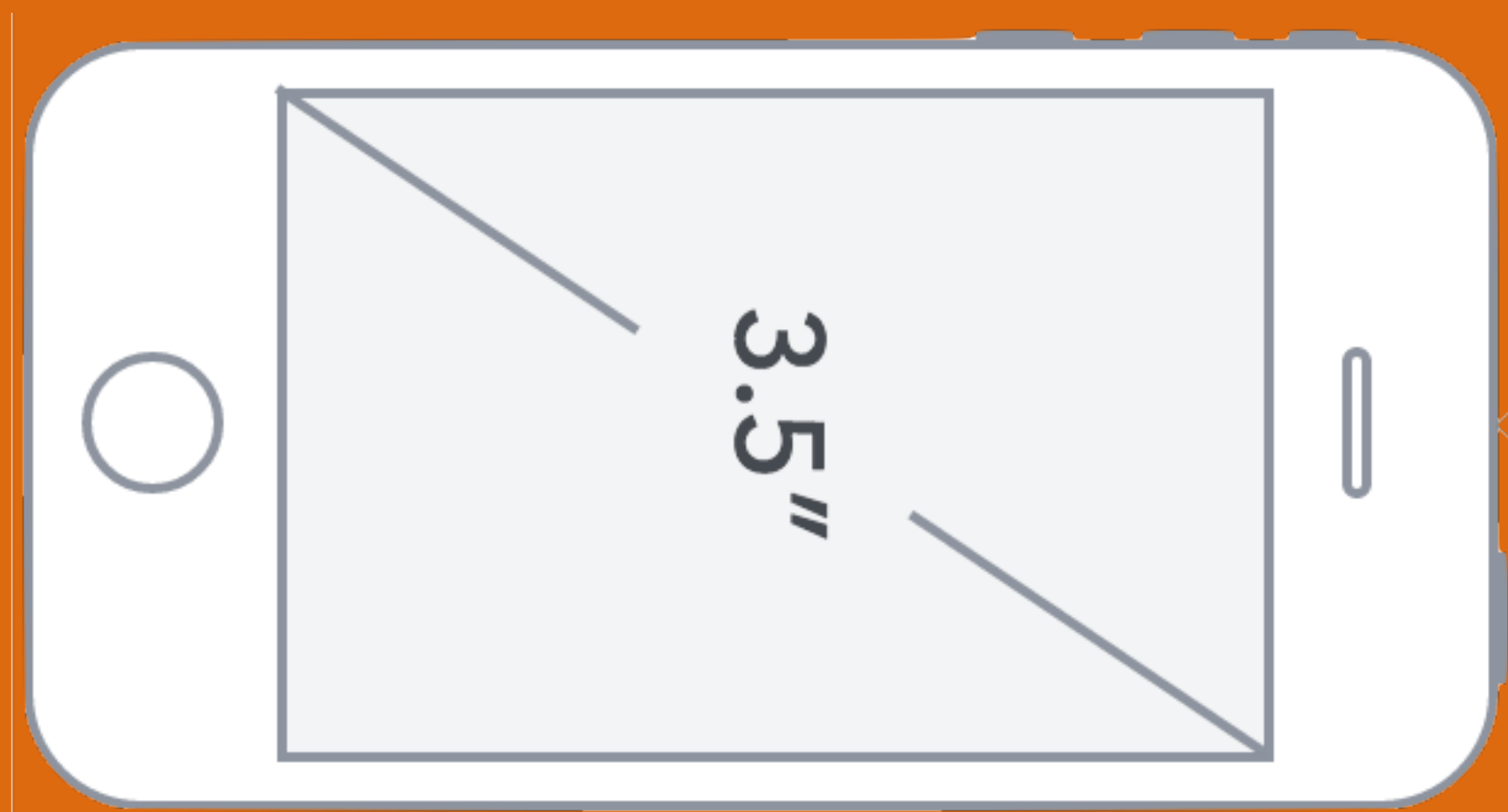
A large, stylized green letter 'R' is positioned on the left side of the image. The background is a solid orange color. The 'R' is composed of a dark green shape with a white, jagged, mountain-like cutout in its center. The cutout has three peaks of varying heights. The overall style is minimalist and modern.

raywenderlich.com

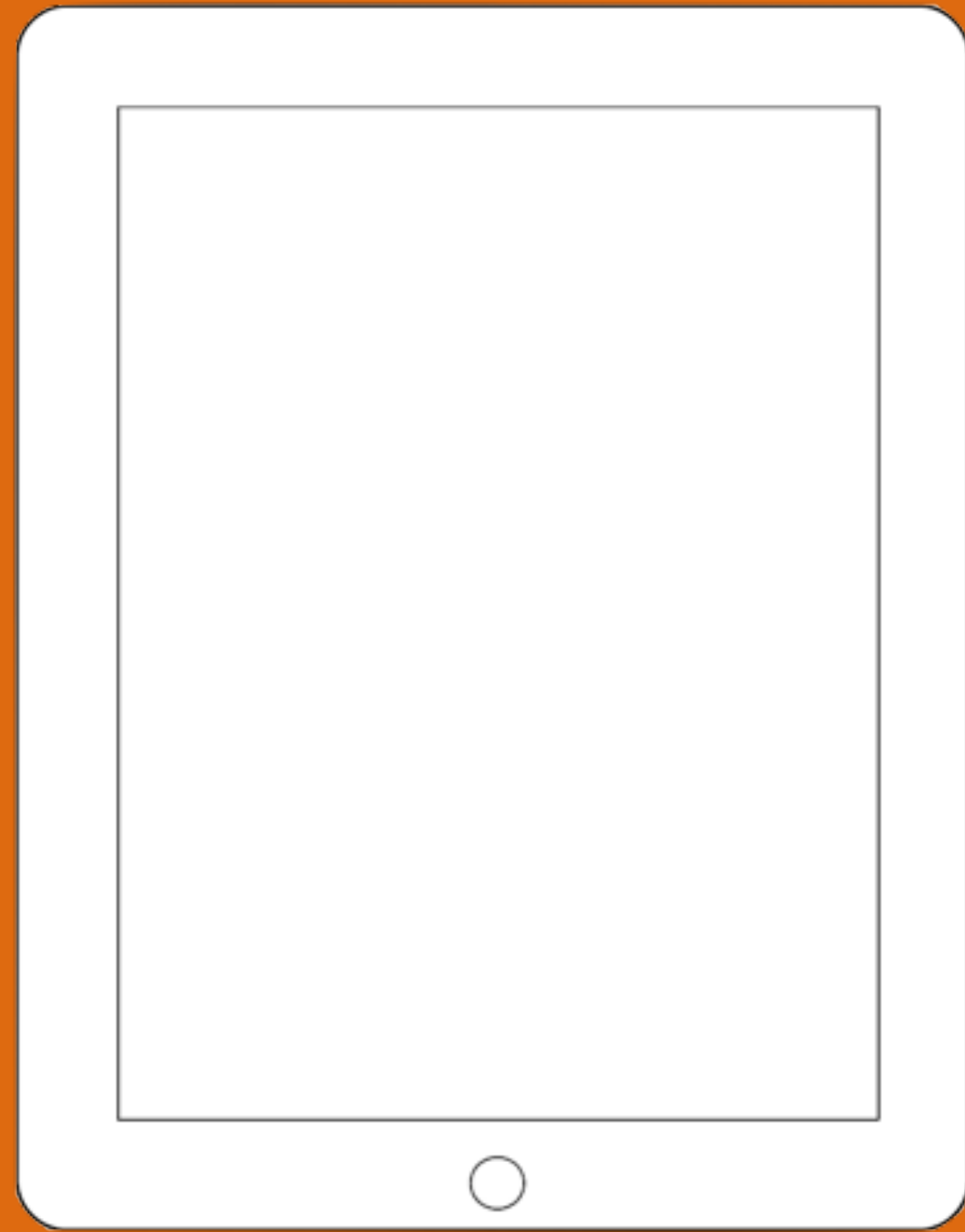
In the beginning there was...



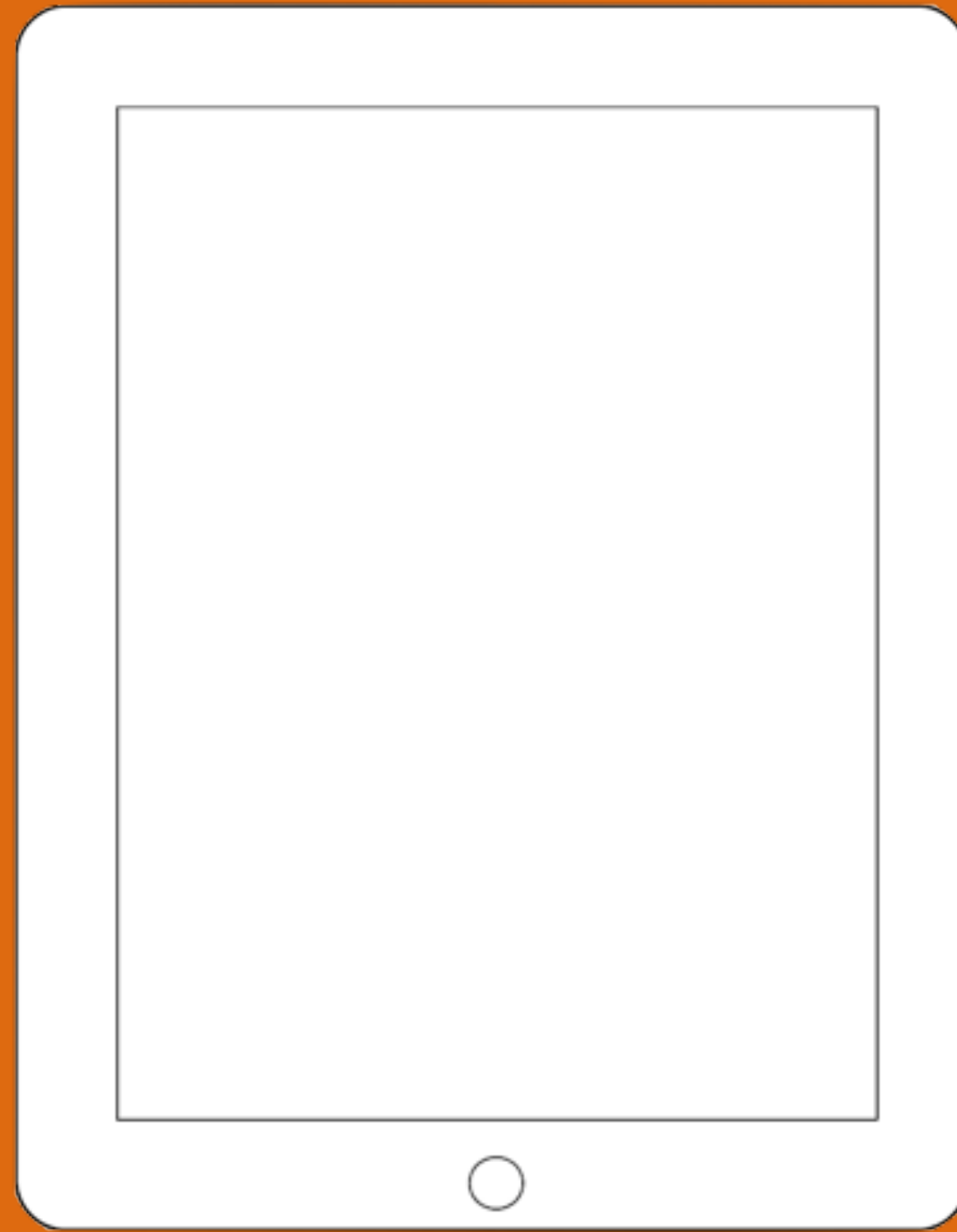
In the beginning there was...



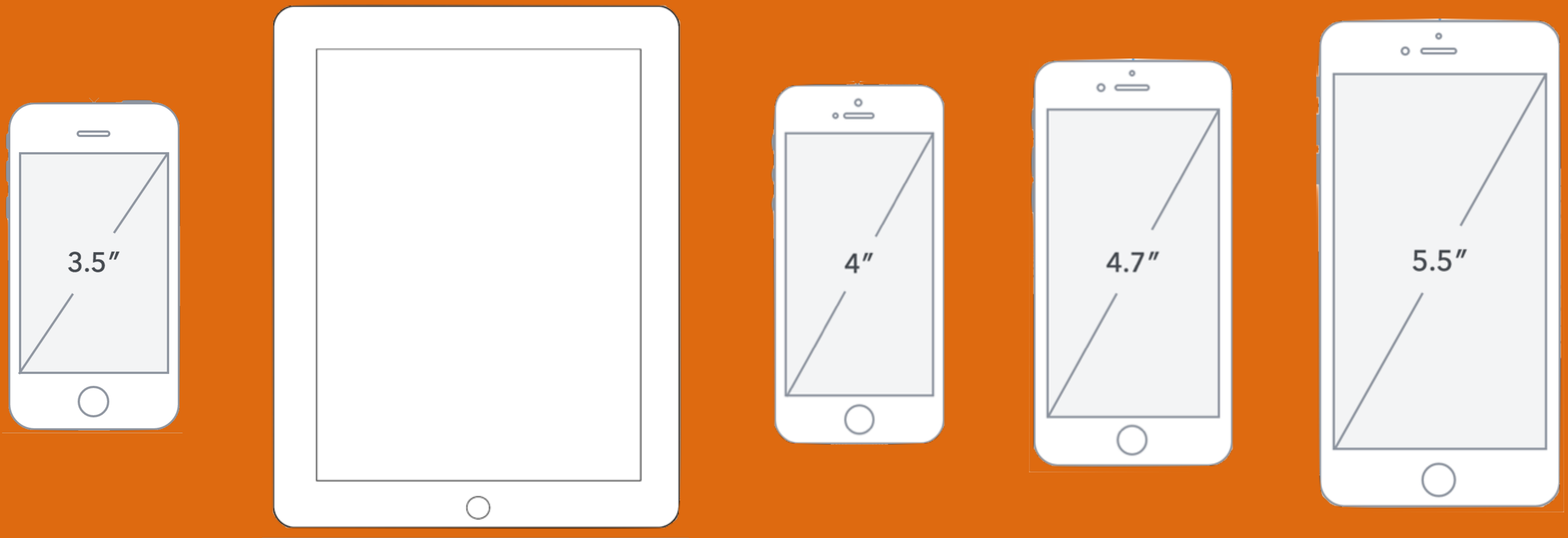
In the beginning there was...



In the beginning there was...

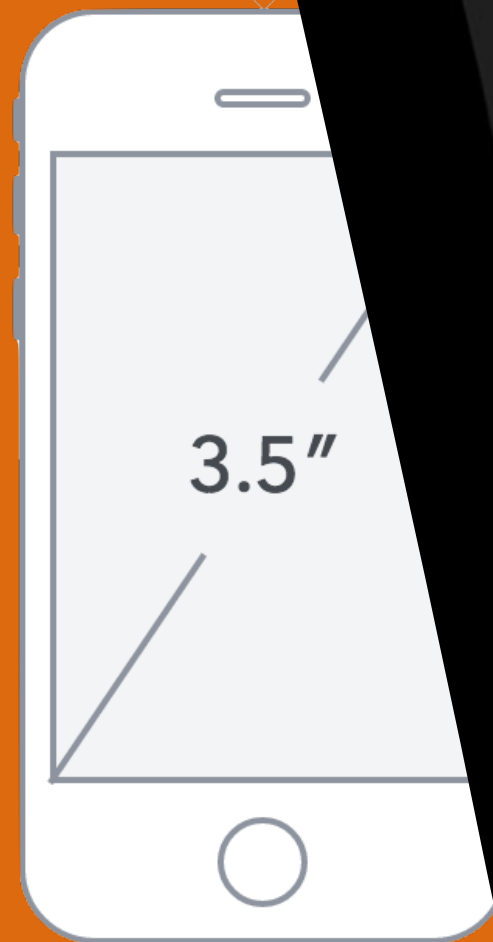


In the beginning there was...

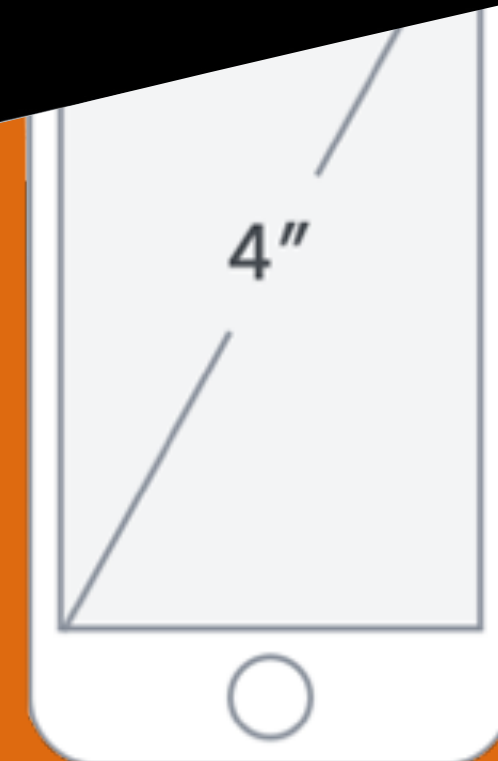
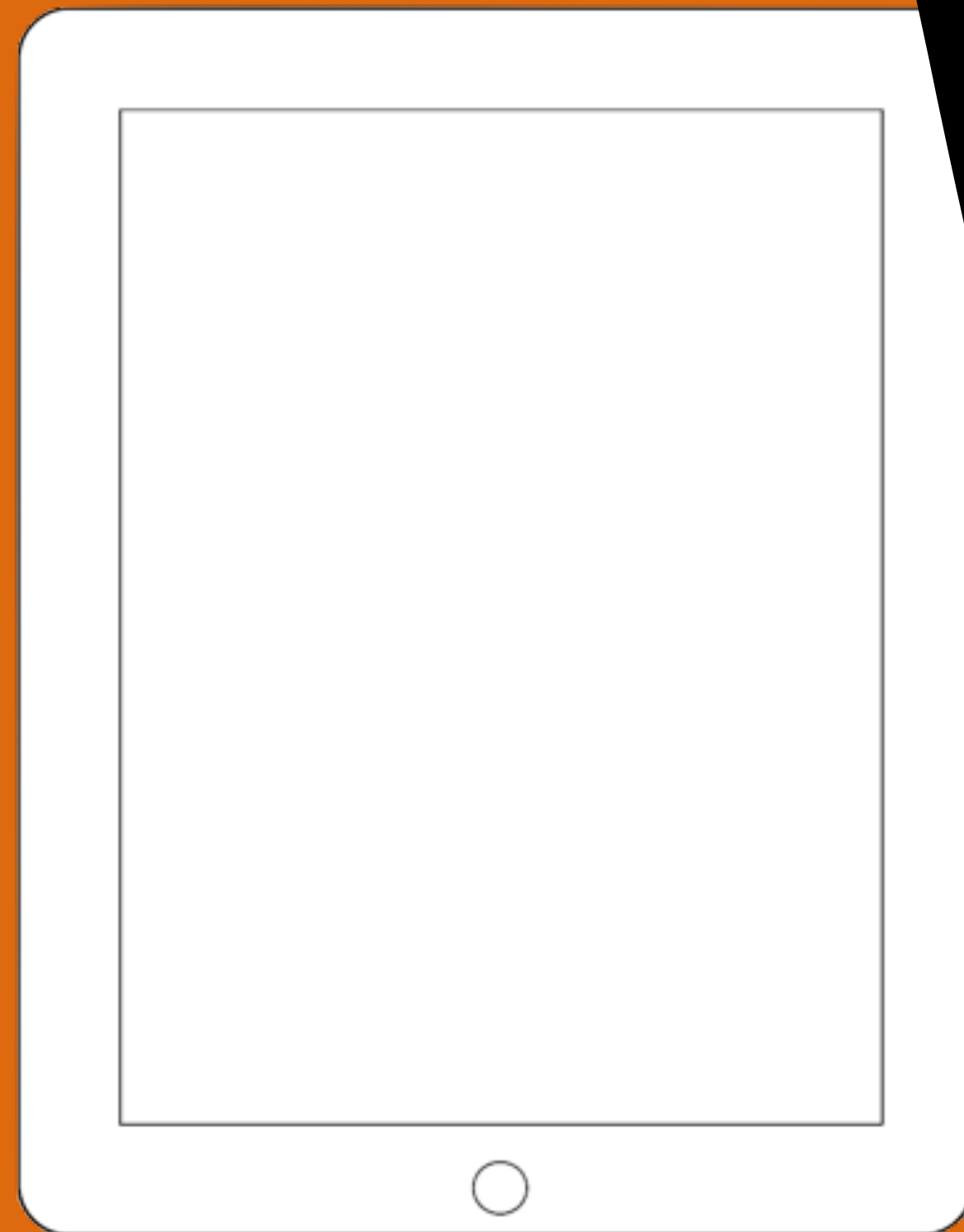


In

S...



In the beginning was...



**12 different
form factors**

in days gone by...

```
46     if UIDevice.currentDevice().userInterfaceIdiom == .Pad {
47         if UIDevice.currentDevice().orientation == .LandscapeLeft ||
48            UIDevice.currentDevice().orientation == .LandscapeRight {
49             doSomething()
50         } else {
51             doSomethingElse()
52         }
53     } else {
54         if UIDevice.currentDevice().orientation == .LandscapeLeft ||
55            UIDevice.currentDevice().orientation == .LandscapeRight {
56             yetAnotherAlternative()
57         } else {
58             theFourthWay()
59         }
60     }
```

this doesn't
scale

Introducing...

Adaptive Layout

Adaptive Layout

- Abstracts layout away from device specifics
- Introducing concept of size classes
- Available in iOS 8
- Fully supported in Xcode and IB

**"amount of
space available
for content"**

Size Classes



compact

A diagram showing the word "compact" in a dark grey font. It is flanked by two vertical yellow bars. A horizontal yellow double-headed arrow spans the width between the bars, passing behind the text.



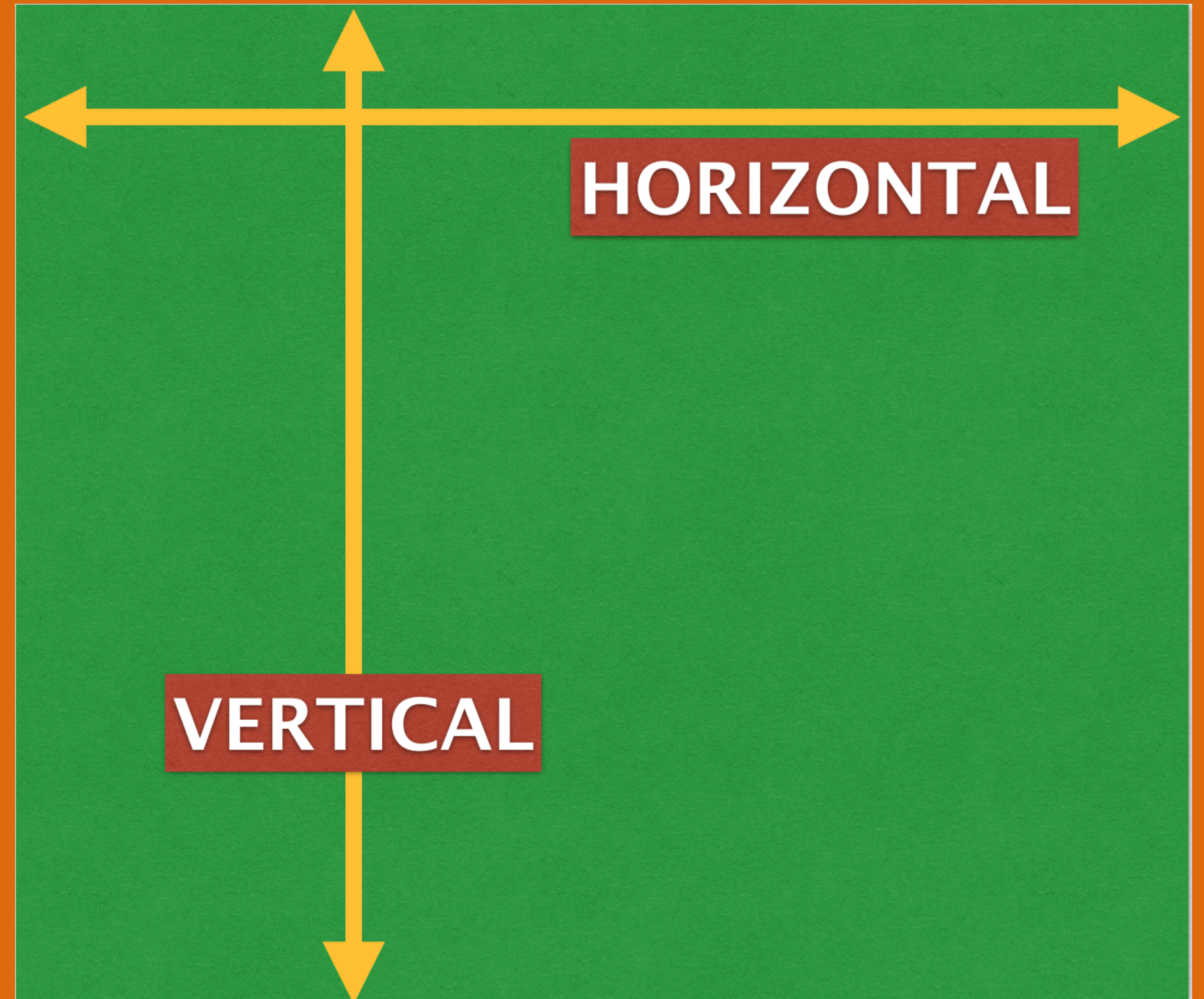
regular

A diagram showing the word "regular" in a dark grey font. It is flanked by two vertical yellow bars. A horizontal yellow double-headed arrow spans the width between the bars, passing behind the text.

Size Classes

compact

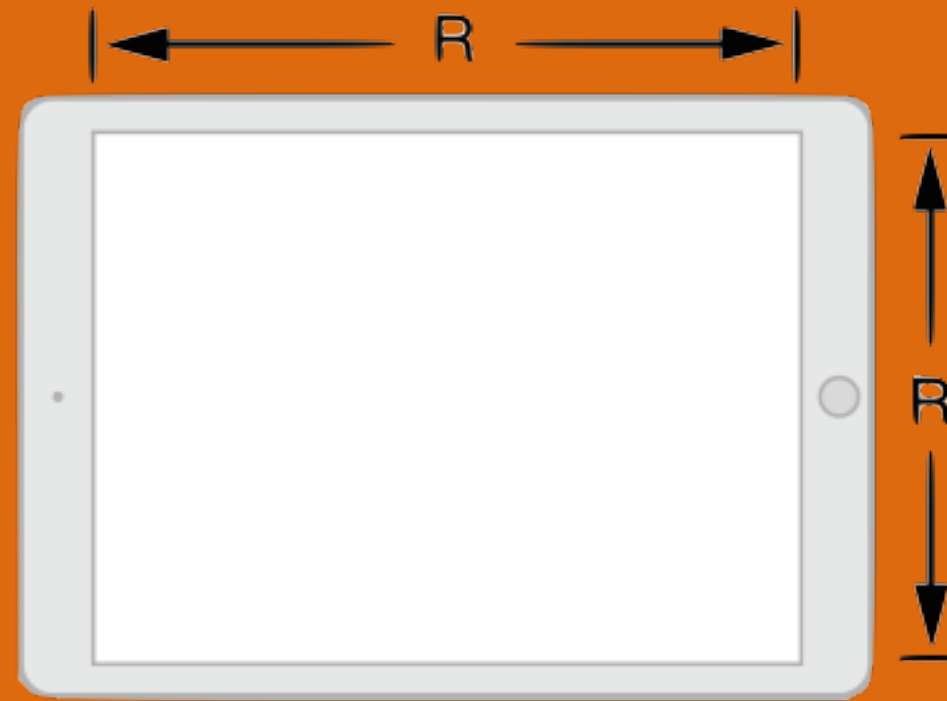
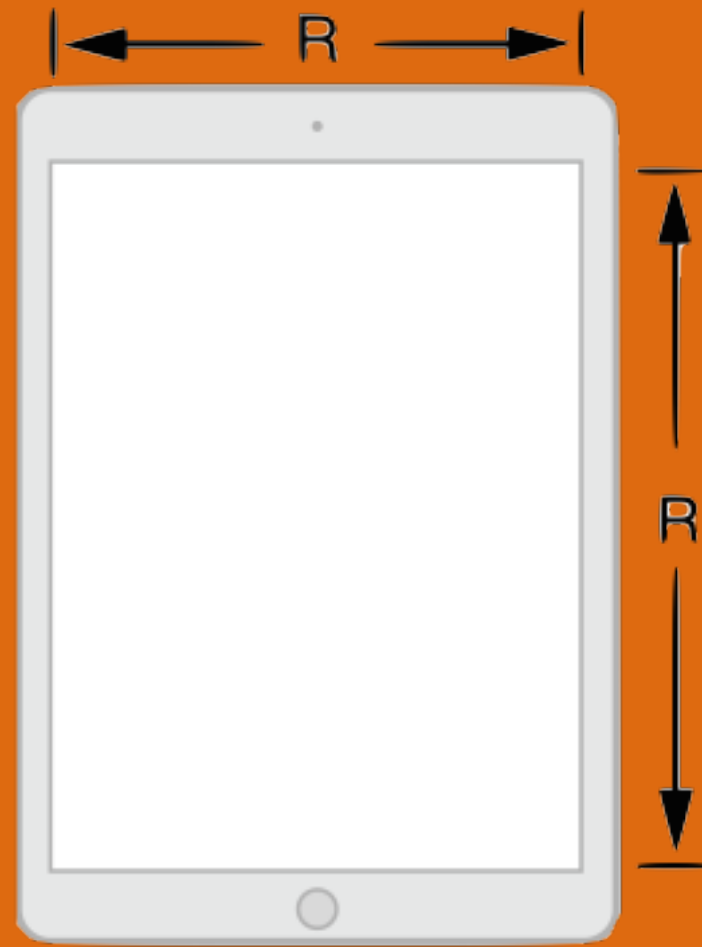
regular



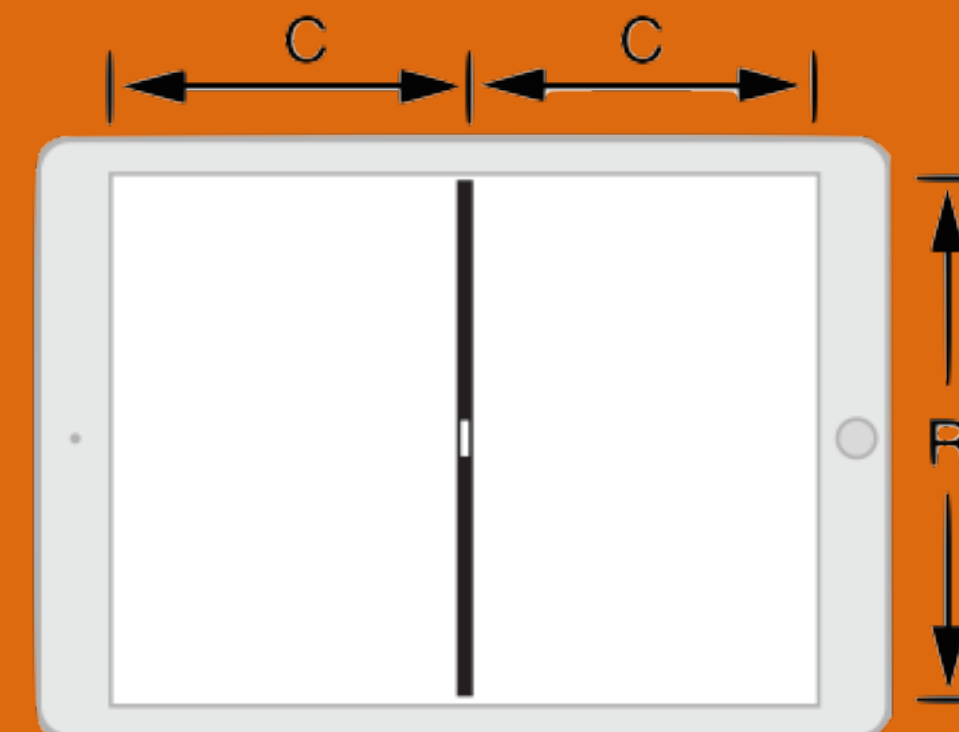
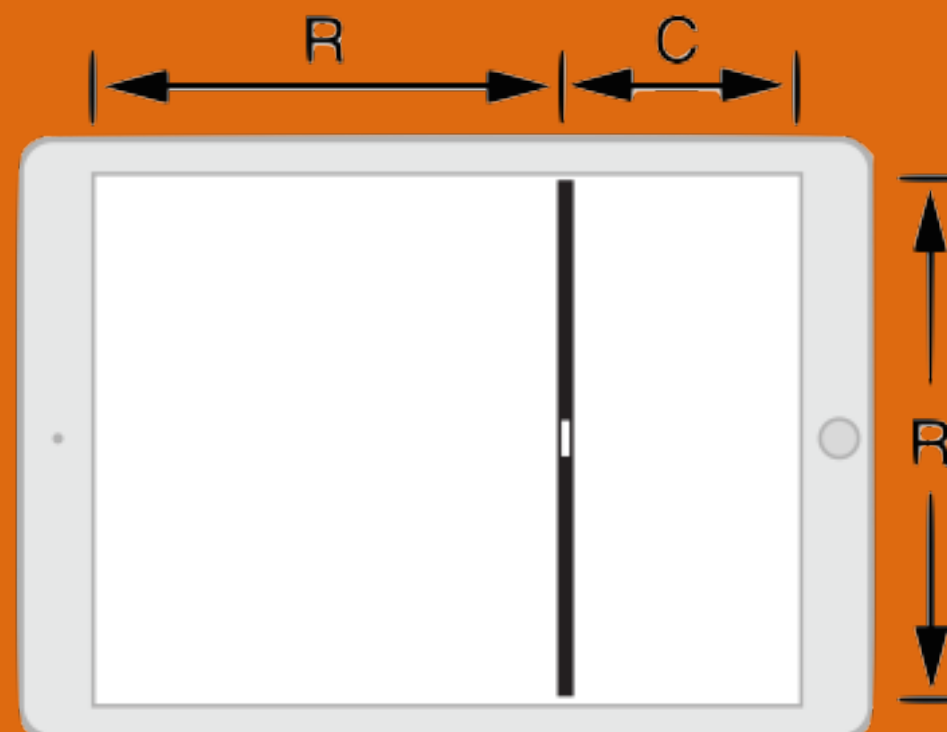
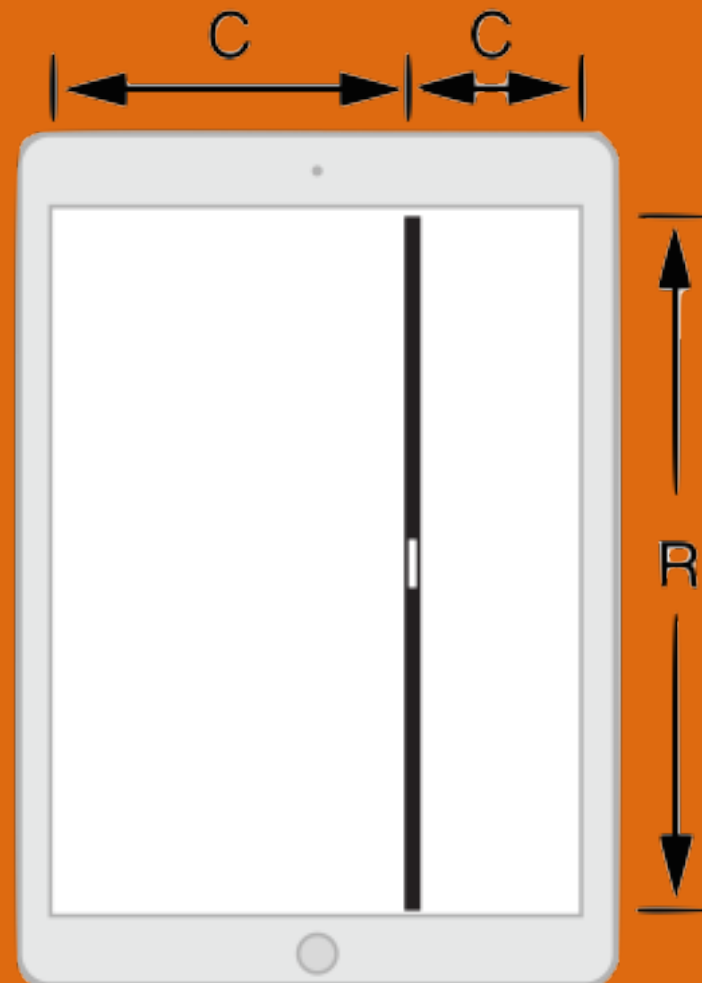
Size Classes

	Horizontal	Vertical
iPad Portrait	Regular	Regular
iPad Landscape	Regular	Regular
iPhone Portrait	Compact	Regular
iPhone Landscape	Compact	Compact
iPhone 6[s] Plus Landscape	Regular	Compact

Multitasking



R = Regular
C = Compact



Adopting Adaptive Layout

1. Build base layout
2. Choose size class override
3. Uninstall irrelevant constraints
4. Add new constraints specific to size class
5. Rinse and repeat

Demo!

What's Adaptive?

- Constraint installation
- Constraint constant
- Font
- View installation

Trait Collections

```
public class UITraitCollection : NSObject, NSCopying, NSSecureCoding, NSCoding {  
    ...  
  
    public var userInterfaceIdiom: UIUserInterfaceIdiom { get }  
    public var displayScale: CGFloat { get }  
    public var horizontalSizeClass: UIUserInterfaceSizeClass { get }  
    public var verticalSizeClass: UIUserInterfaceSizeClass { get }  
  
    @available(iOS 9.0, *)  
    public var forceTouchCapability: UIForceTouchCapability { get }  
}
```

Trait Environments

```
public protocol UITraitEnvironment : NSObjectProtocol {  
    public var traitCollection: UITraitCollection { get }  
  
    public func traitCollectionDidChange(previousTraitCollection: UITraitCollection?)  
}
```

- UIScreen
- UIWindow
- UIPresentationController
- UIViewController
- UIView

Overriding Size Classes

```
extension UIViewController {  
    public func setOverrideTraitCollection(collection: UITraitCollection?,  
        forChildViewController childViewController: UIViewController)  
  
    public func overrideTraitCollectionForChildViewController(  
        childViewController: UIViewController) -> UITraitCollection?  
}
```

UIContentContainer

```
public protocol UIContentContainer : NSObjectProtocol {  
    ...  
  
    public func viewWillTransitionToSize(size: CGSize,  
        withTransitionCoordinator coordinator:  
        UIViewControllerTransitionCoordinator)  
  
    public func willTransitionToTraitCollection(  
        newCollection: UITraitCollection,  
        withTransitionCoordinator coordinator:  
        UIViewControllerTransitionCoordinator)  
}
```

Rotation Deprecation

```
extension UIViewController {
    @available(iOS, introduced=2.0, deprecated=8.0)
    public var interfaceOrientation: UIInterfaceOrientation { get }

    @available(iOS, introduced=2.0, deprecated=8.0,
        message="Implement viewWillTransitionToSize:withTransitionCoordinator: instead")
    public func willRotateToInterfaceOrientation(
        toInterfaceOrientation: UIInterfaceOrientation, duration: NSTimeInterval)
    @available(iOS, introduced=2.0, deprecated=8.0)
    public func didRotateFromInterfaceOrientation(
        fromInterfaceOrientation: UIInterfaceOrientation)

    @available(iOS, introduced=3.0, deprecated=8.0,
        message="Implement viewWillTransitionToSize:withTransitionCoordinator: instead")
    public func willAnimateRotationToInterfaceOrientation(
        toInterfaceOrientation: UIInterfaceOrientation, duration: NSTimeInterval)
}
```

Rotation

```
override func viewWillTransitionToSize(size: CGSize,
withTransitionCoordinator coordinator: UIViewControllerTransitionCoordinator) {

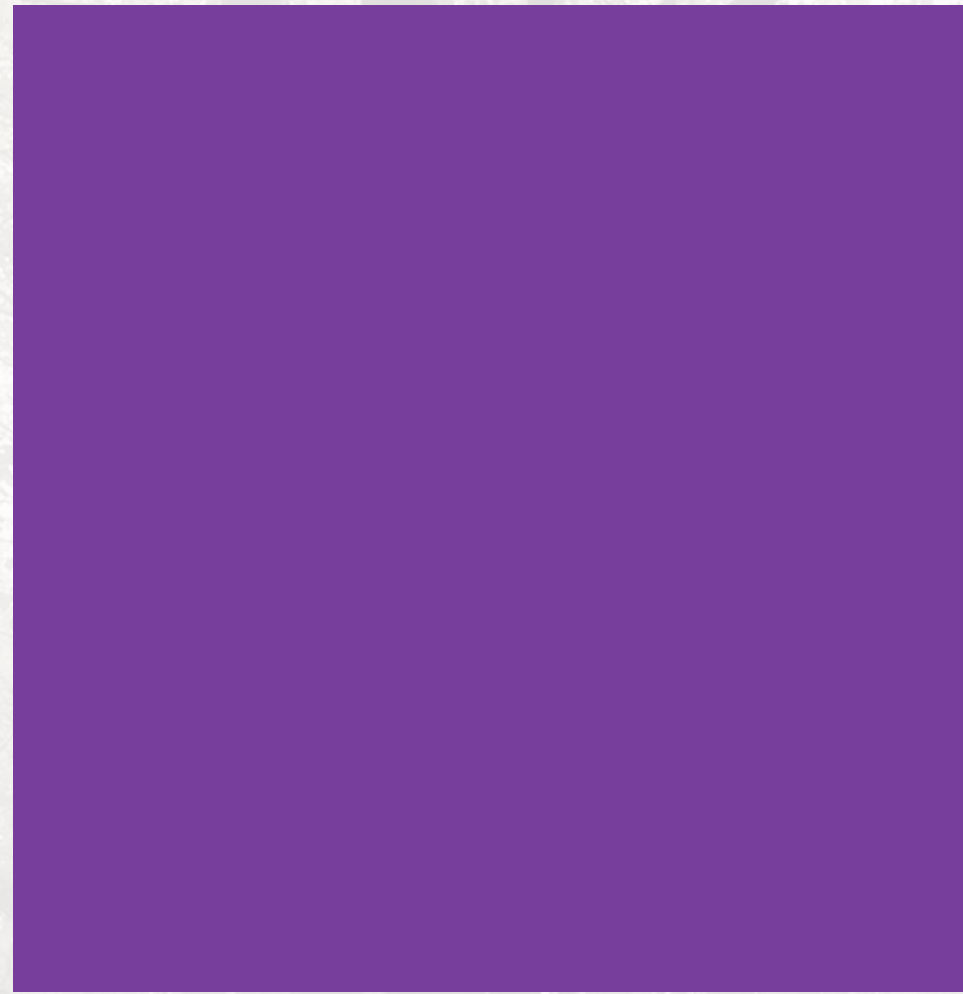
    super.viewWillTransitionToSize(size, withTransitionCoordinator: coordinator)
    let image = imageForAspectRatio(size.width / size.height)

    coordinator.animateAlongsideTransition({
        context in
        // Create a transition and match the context's duration
        let transition = CATransition()
        transition.duration = context.transitionDuration()

        // Make it fade
        transition.timingFunction = CAMediaTimingFunction(name: kCAMediaTimingFunctionEaseInEaseOut)
        transition.type = kCATransitionFade
        self.backgroundImageView.layer.addAnimation(transition, forKey: "Fade")

        // Set the new image
        self.backgroundImageView.image = image
    }, completion: nil)
}
```

Stack Views



Pin to Top

Pin to Leading

Pin to Trailing

Pin to Bottom



Pin to Top

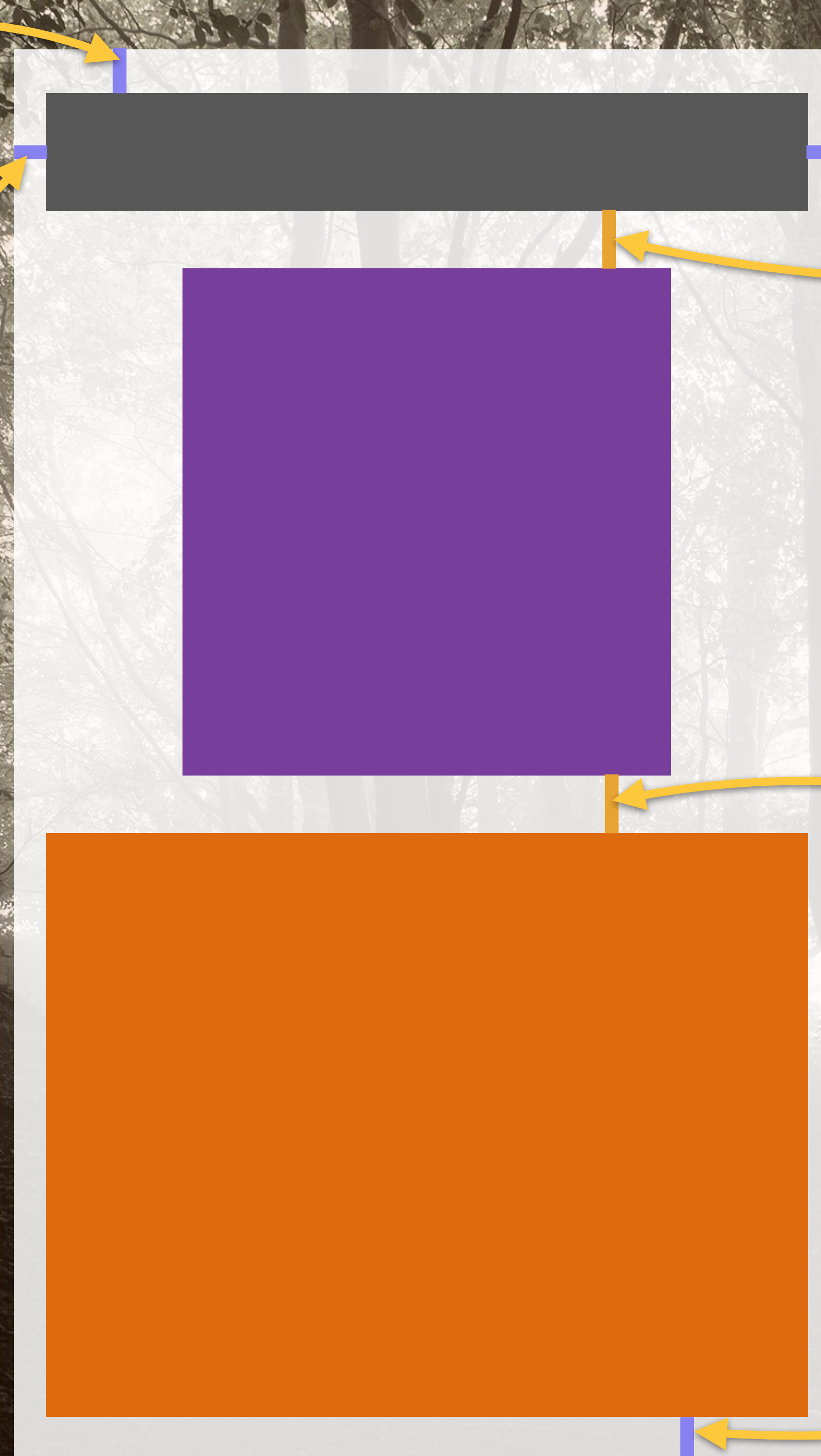
Pin to Leading

Pin to Trailing

Vertical Spacing

Vertical Spacing

Pin to Bottom



Pin to Top

Pin to Leading

Centre Align

Centre Align

Pin to Trailing

Vertical Spacing

Vertical Spacing

Pin to Bottom



Pin to Top

Pin to Leading

Centre Align

Centre Align

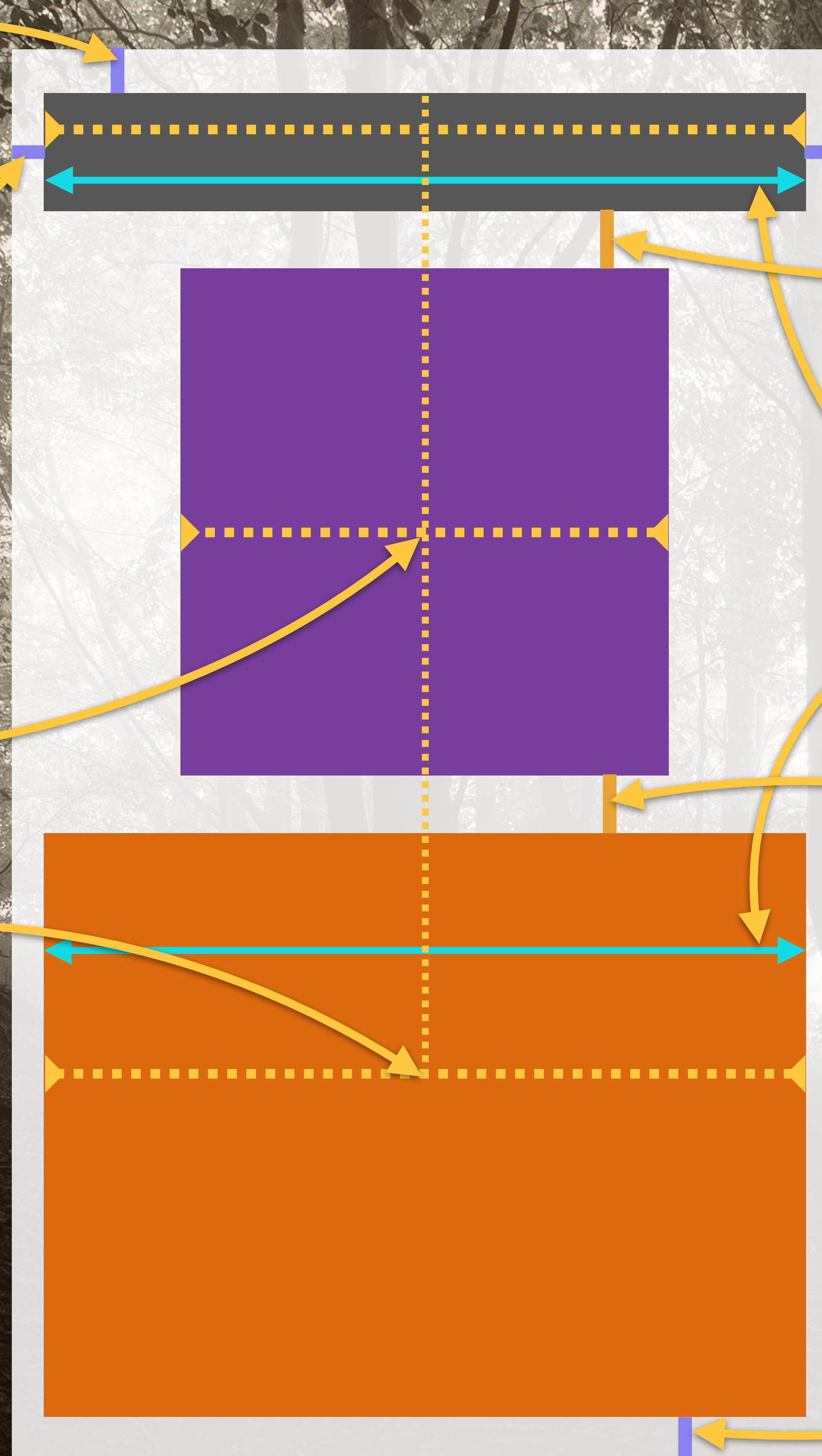
Pin to Trailing

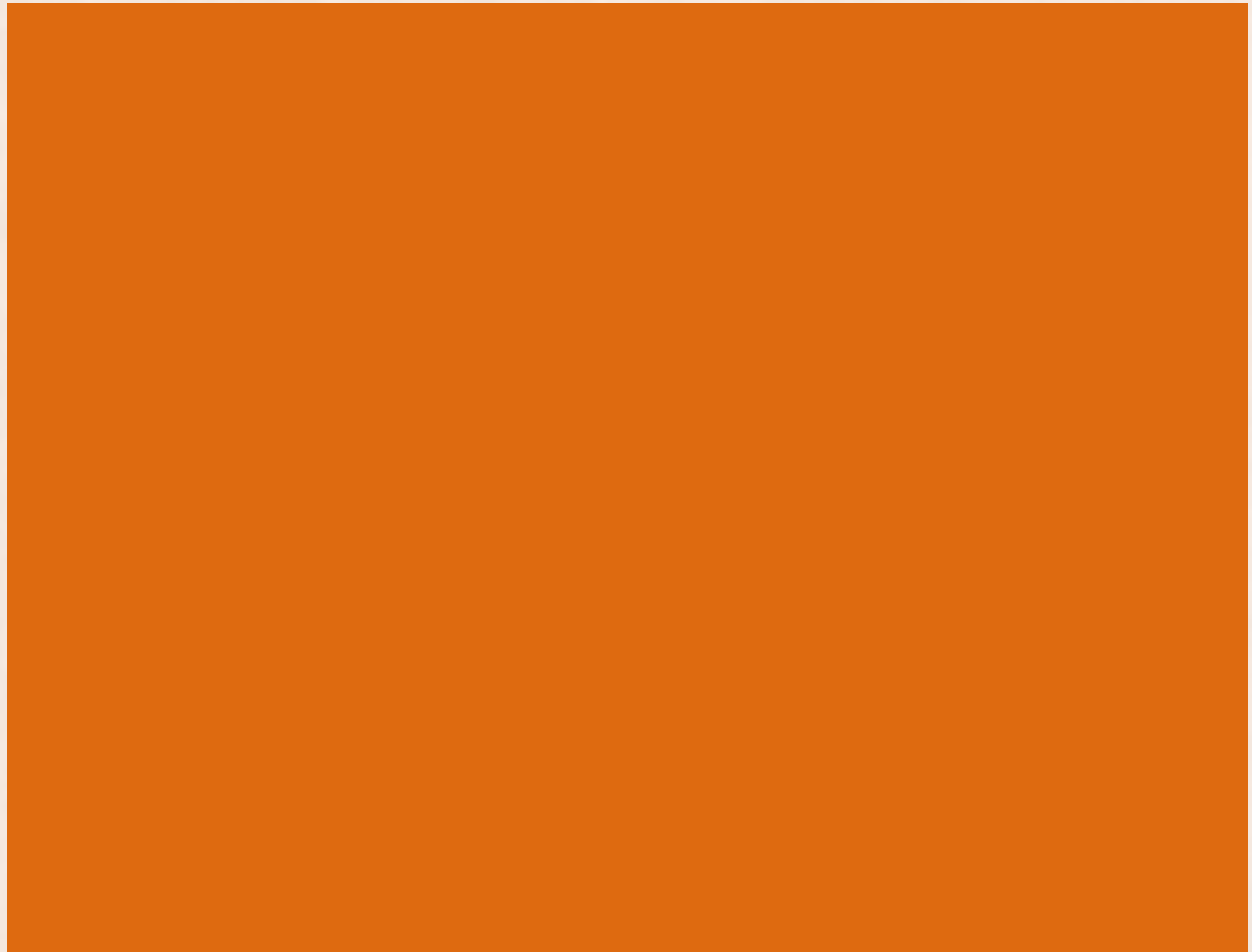
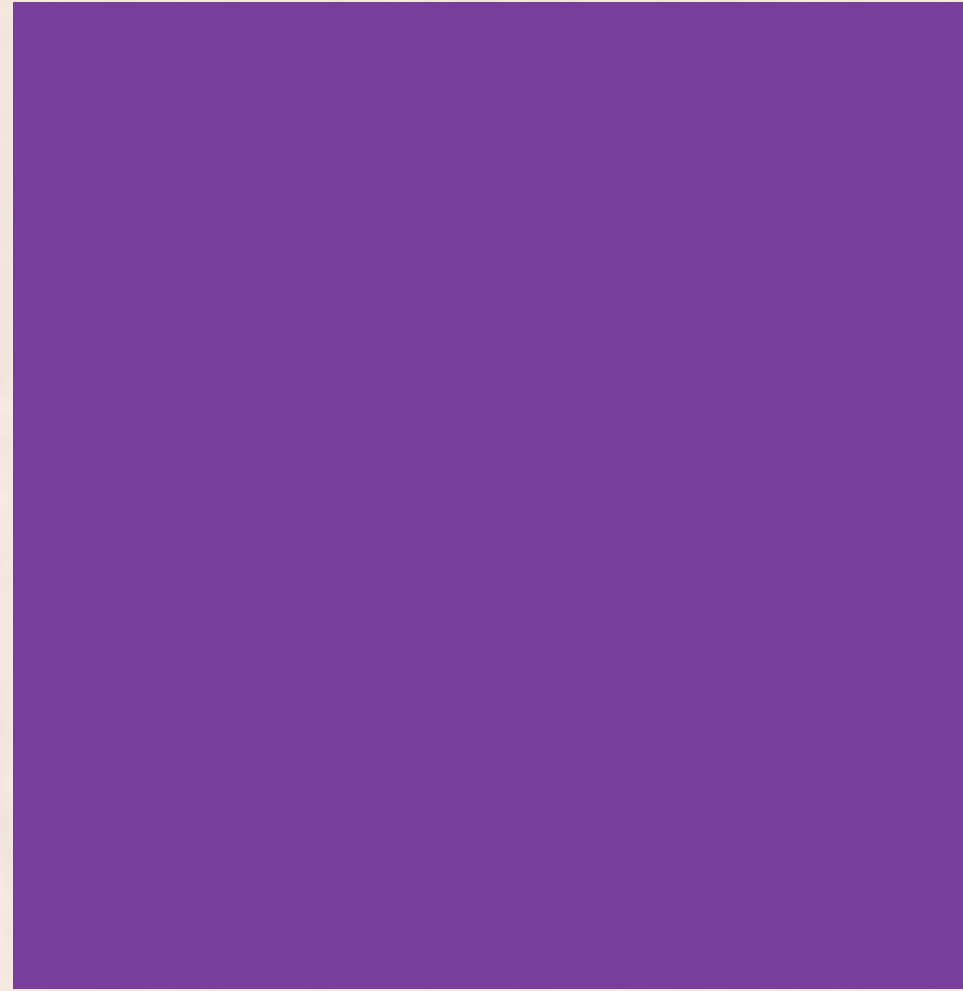
Vertical Spacing

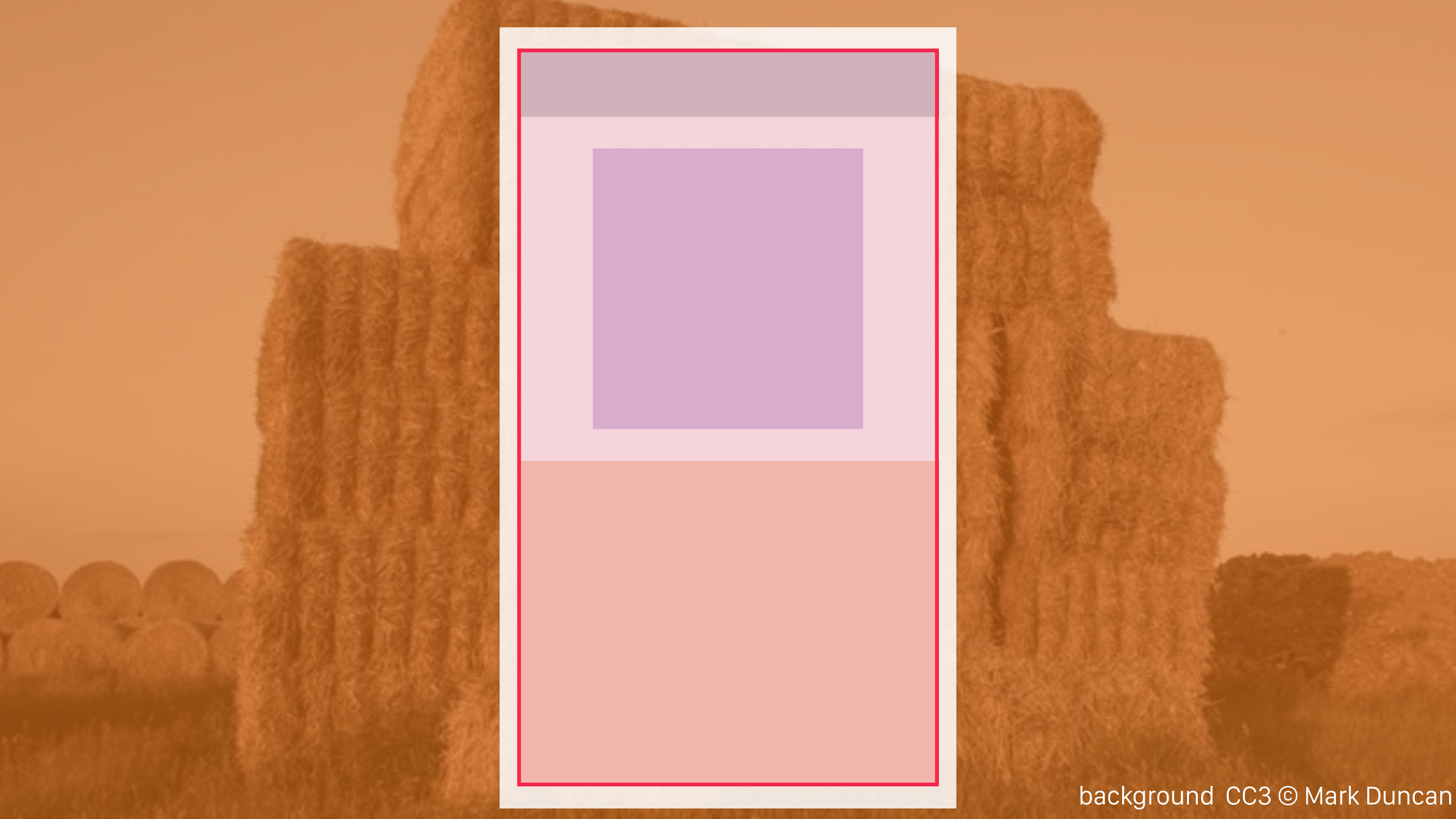
Equal Width

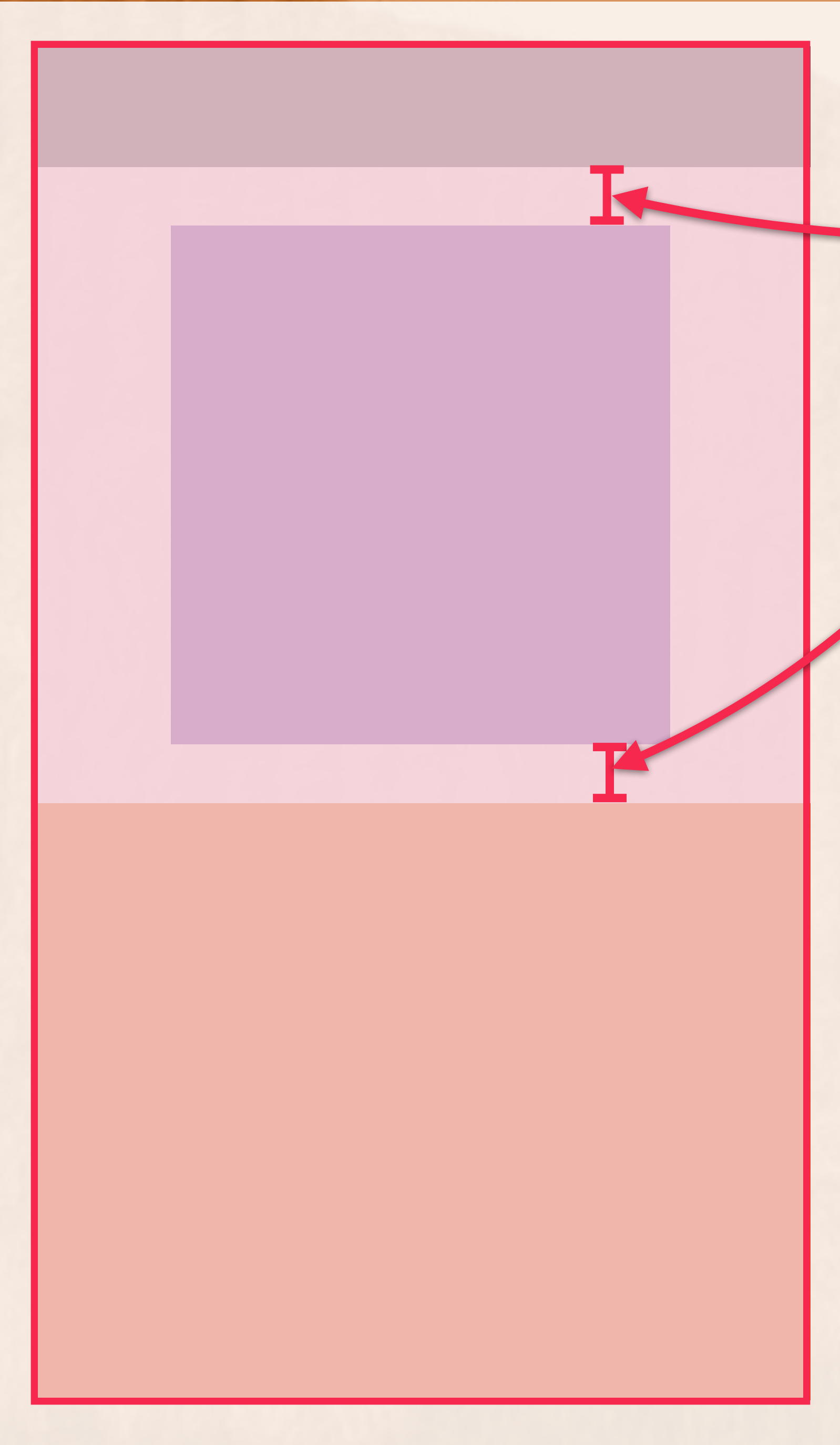
Vertical Spacing

Pin to Bottom

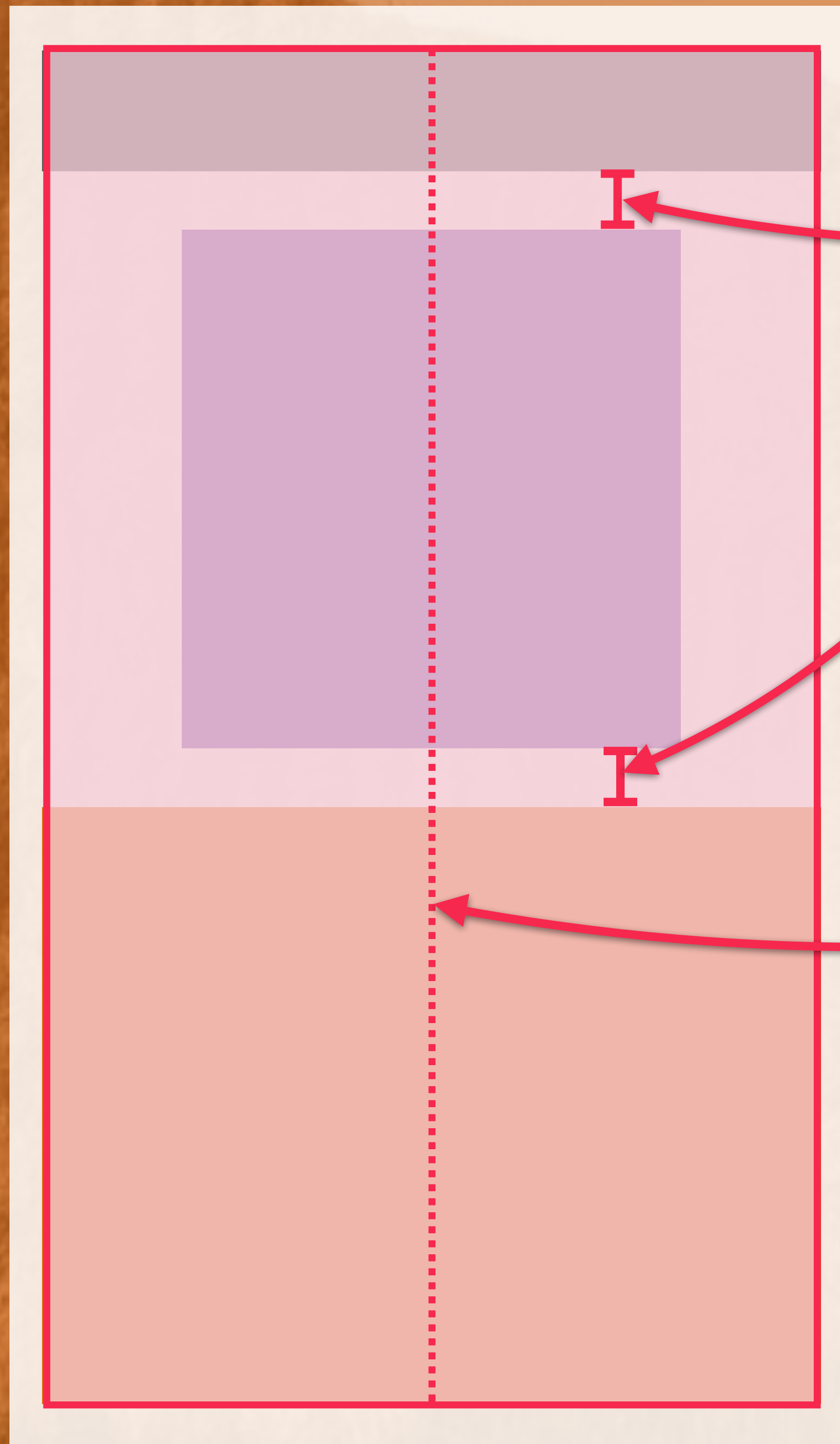








Spacing



Spacing

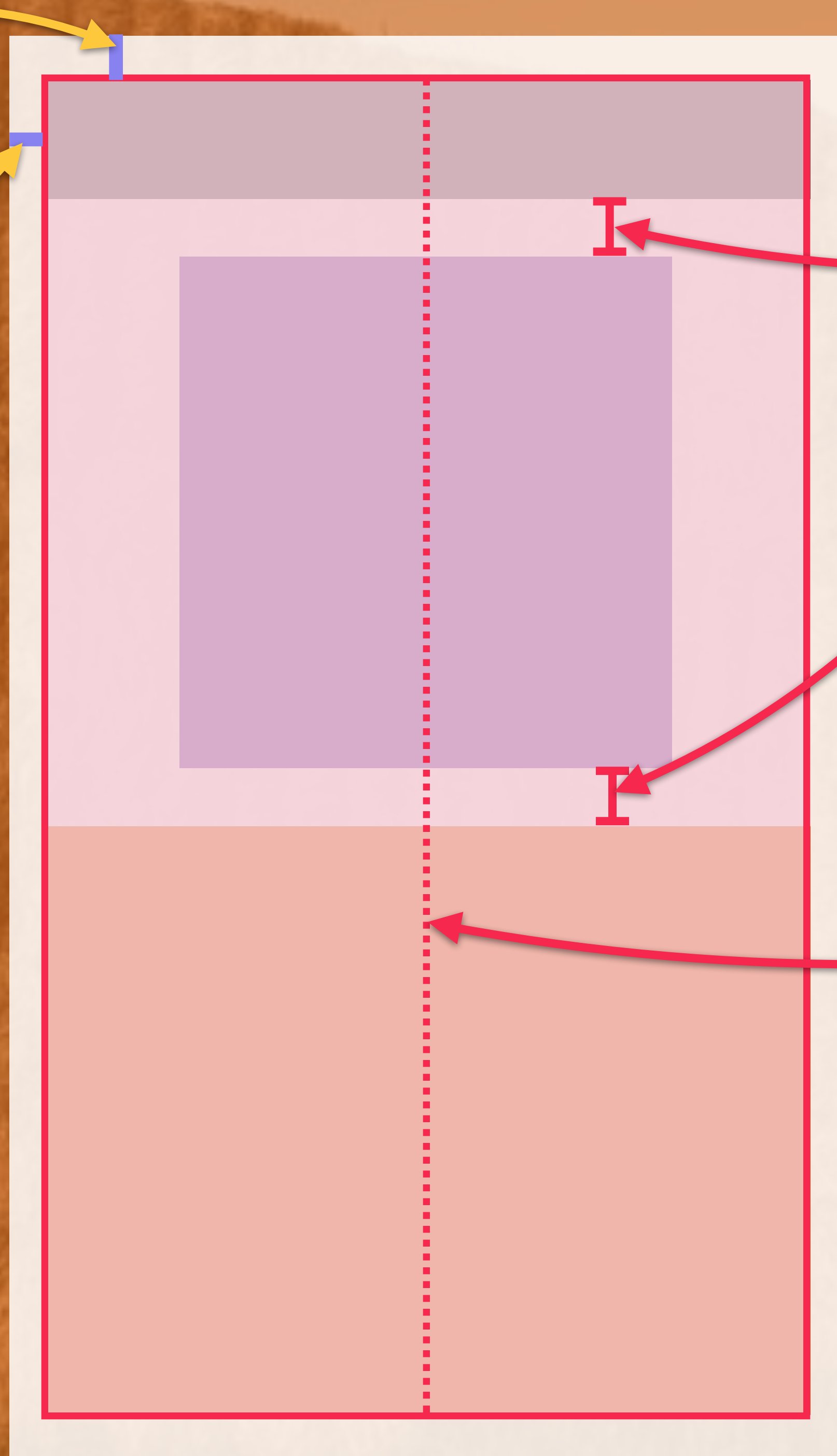
Centre Aligned

Pin to Top

Pin to Leading

Spacing

Centre Aligned



Pin to Top

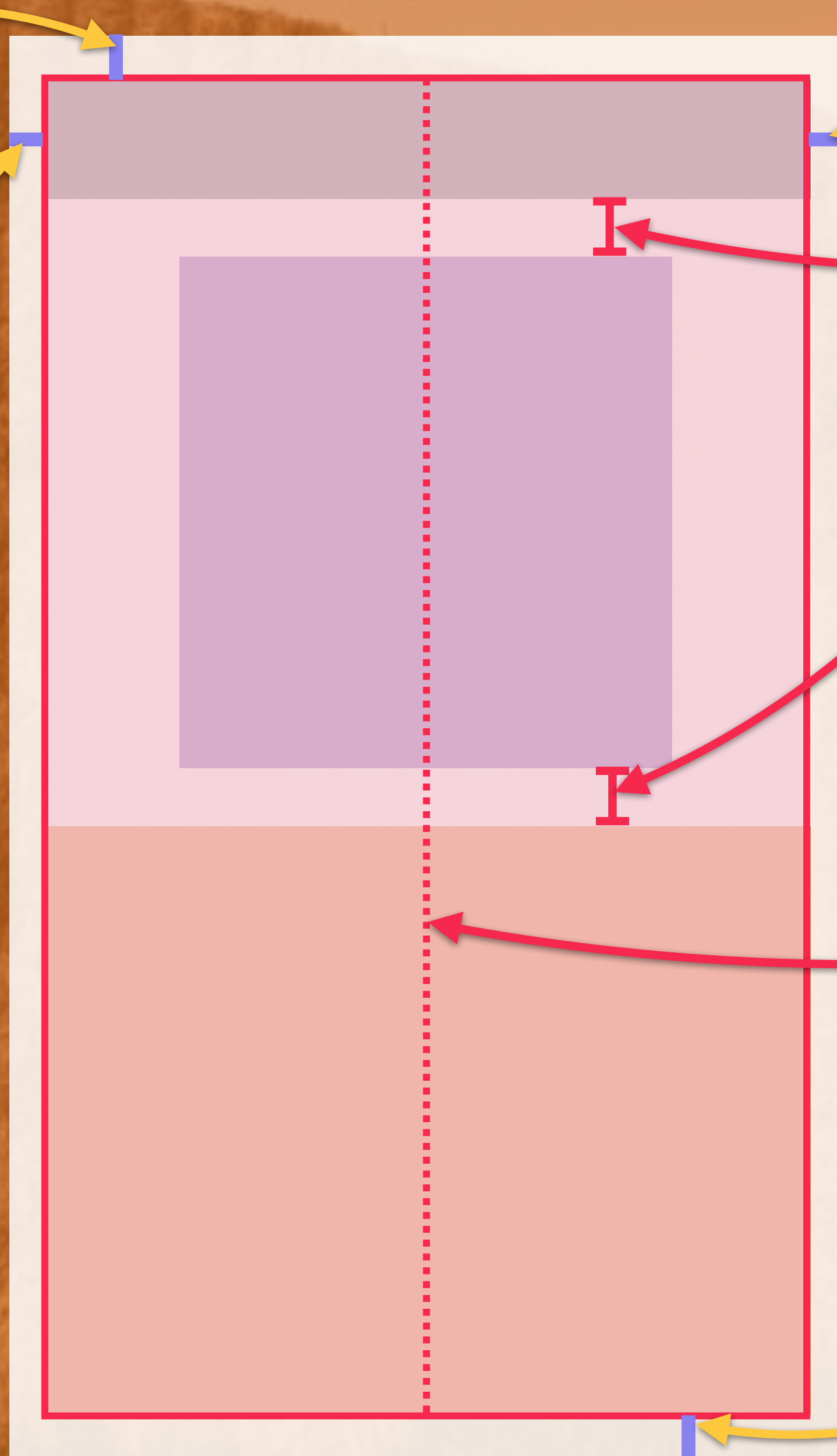
Pin to Leading

Pin to Trailing

Spacing

Centre Aligned

Pin to Bottom



Demo!

**Some tips for
adaptivity**

1

get to know

Auto Layout

2

*use adaptive
layout for*
Broad Strokes

3

start with

Base Layout

and then

Override

4

life is easier with

**Stack
Views**

The background of the slide features a warm, orange-toned image. It shows a wooden ruler with black markings and numbers, placed diagonally across the frame. In the background, a smartphone is visible, with the Apple logo clearly shown. The text is overlaid on this background.

Now's the time to get adaptive

@iwantmyrealname

github.com/sammyd/AdaptiveUI-GOTOTConf



Please

**Remember to
rate this session**

Thank you!

