

A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is a light green. They are positioned diagonally, with the blue one partially covering the green one.

Layouts in Flutter

Maximilian Klein



Me@Freiheit.com

Fullstack Engineer (32)

> 1 Year Flutter experience

Prior to Freiheit.com

Years of Web experience

Projects in VR and Physics Simulations
(most notably with Adidas and Harting)





1. Basic Layout Widgets
2. Flutter's Linear Layout Algorithm
3. Customizing Layouts



Basics – Simplified Flutter Pipeline

1. Build
2. Layout
3. Paint



Basics – Simplified Flutter Pipeline

1. Build
2. Layout
3. Paint



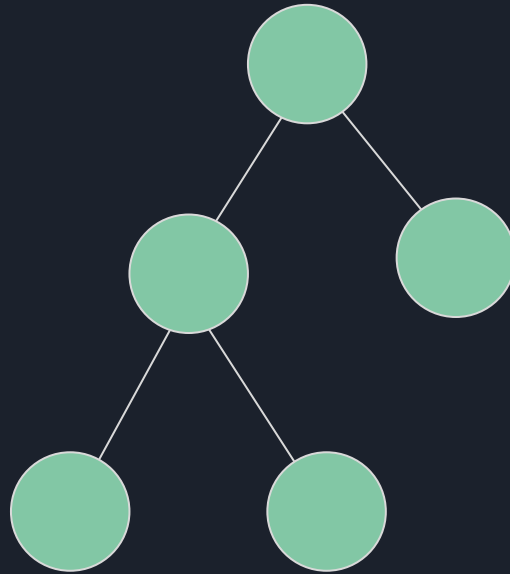
Basic Layout Widgets

Widgets

- Container (allrounder)
- Align
- Flex (Row / Column)
- Stack (overlapping layers)
- ... (all the basics and more)

Linear Layout Algorithm

Constraints

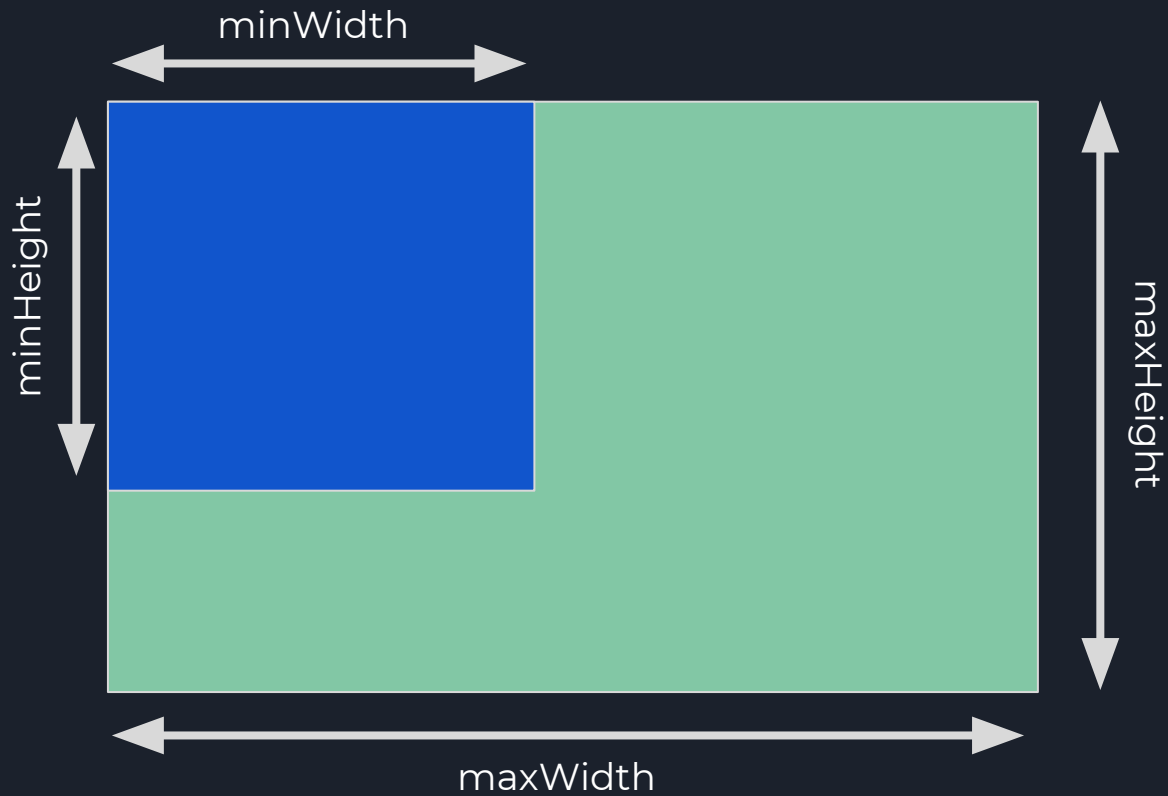


tree built from widgets



Sizes

BoxConstraints





Size: infinity

In a **constrained** layout
infinite size **fills** the **parent**
constraints

In an **unconstrained** layout
infinite size **shrinks** the **child**



Unconstrained Widgets

1. Column / Row
2. ListView (scrollable)
3. Viewports (slivers)

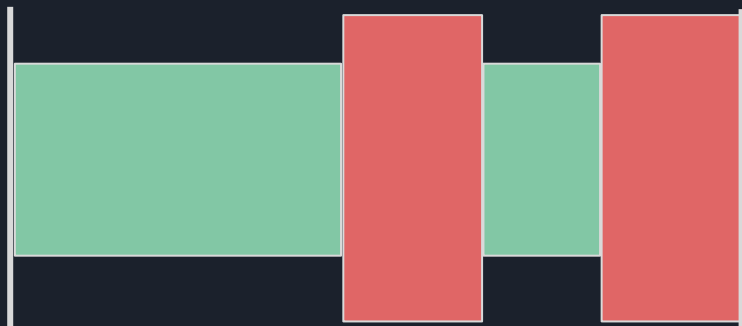


Unconstrained Widgets

1. Column / Row
2. ListView (scrollable)
3. Viewports (slivers)

Flexibles (Column / Row)

Row



 non-flex  flex

1. Calculate non-flex sizes
2. Calculate remaining size
3. Fill remaining space in equal parts with flex children



Flex *tight* vs. *loose*

Flex ***tight*** forces its child
to have a specific size

Expanded

Flex ***loose*** takes the assigned
space, but does **not force** its
child to have a size

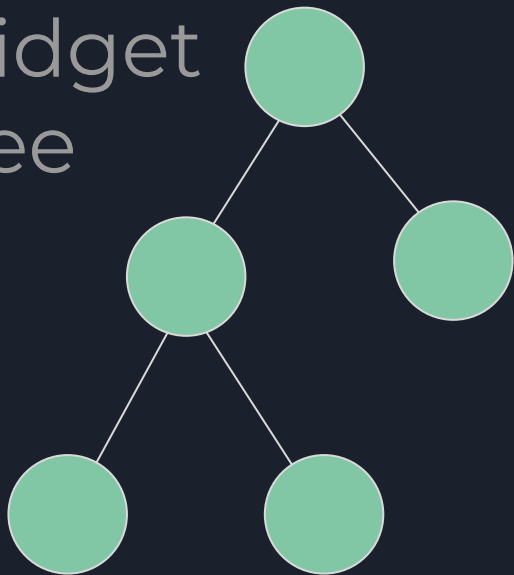


Stacks

1. Overlay one widget with another
2. Absolute positioning

Layout Internals

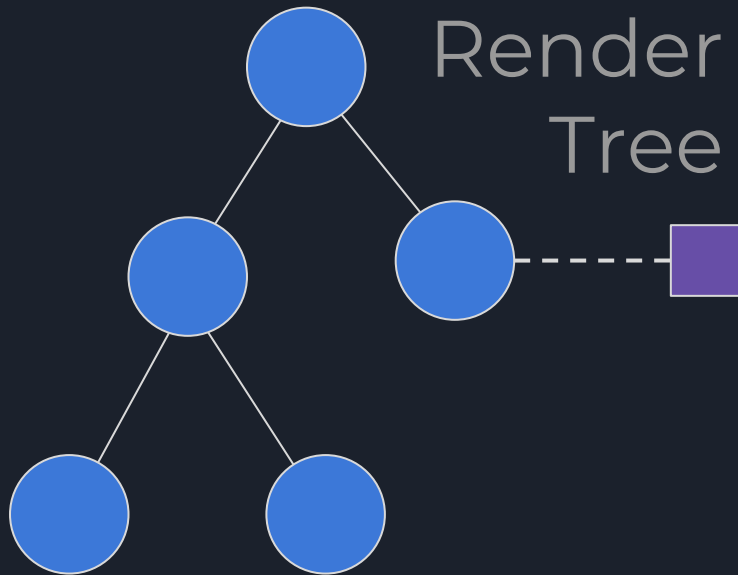
Widget
Tree



Regular rebuild



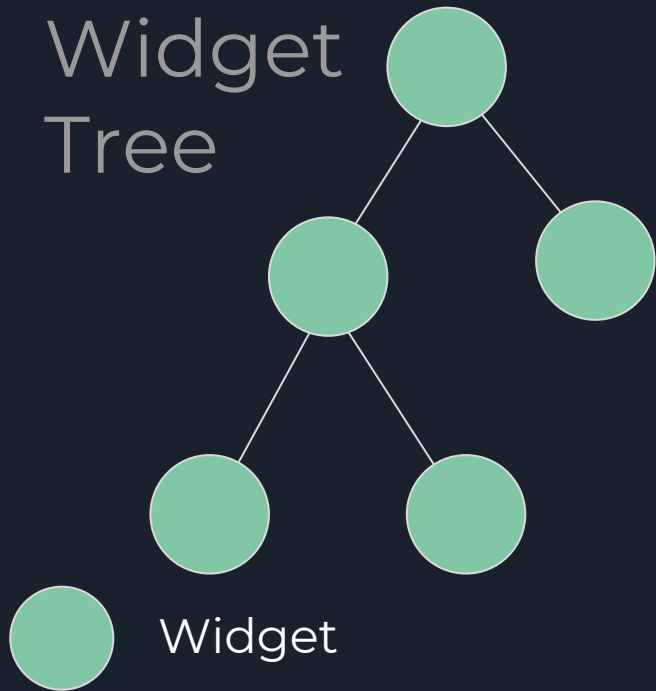
Render
Tree



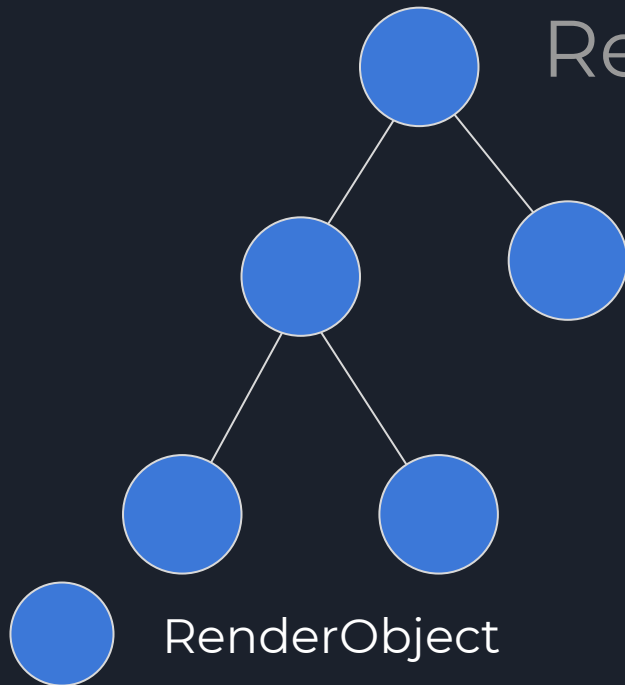
Persists state (and more)

Render Tree

Widget
Tree



Render
Tree





Render Object

- constraints
- ``parentData`` (e.g. position)
- ``performLayout``
- ``paint``
- ...

Everything is still happening in Dart



Wrap up

1. We saw some layout widgets
2. Discussed the Flutter's layout algorithm
3. Saw a custom Flutter layout