VentureDive

Flutter Training

Custom widgets and responsive screen design

By Vinod Kumar

Recap

- Explore different types of widgets
- Explore widget catalog
- Week #2 Recorded Session:

  https://drive.google.com/open?id=1rUJ3vrpwZrmV-KBHxbR07-4guiGTUwwk
- Week #2 Presentation (Flutter Basics):

  https://docs.google.com/presentation/d/1bL1pE-VvCrVGSTxlWPIunH7x0_154tI7-TZiXhBZRow/edit?usp=sharing

Agenda

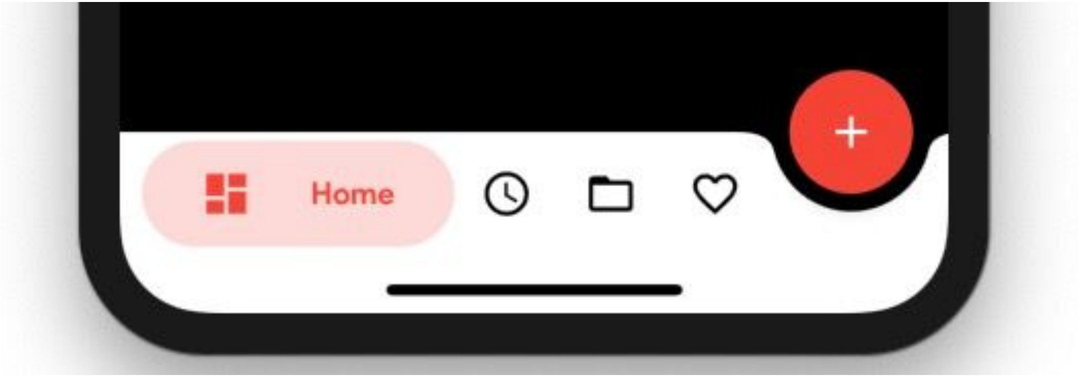
- How to build reusable components
- How to make your UI responsive
- Explore responsive widgets

Constructor parameters

- Required parameters
  - Must be provided and in order.
  - Must be listed before optional parameters
  - Cannot have default value
- Optional parameters
  - Positional parameters []
  - Named Parameters {}
  - Can have default value

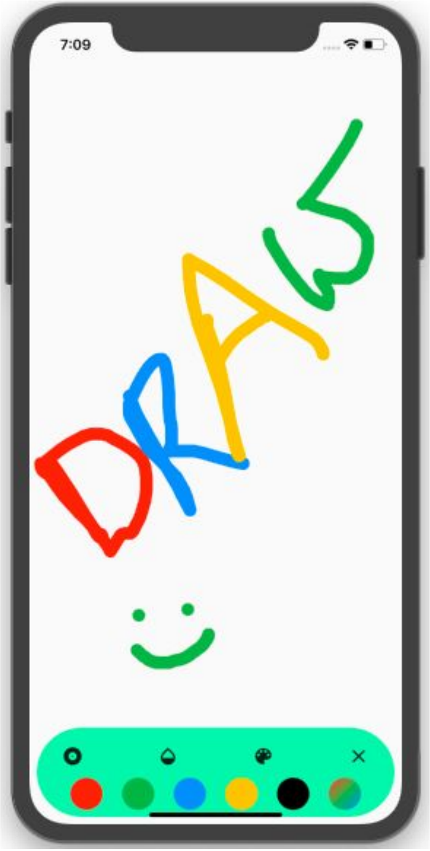
Reusable Components

- Custom Widget
  - Composition over Inheritance



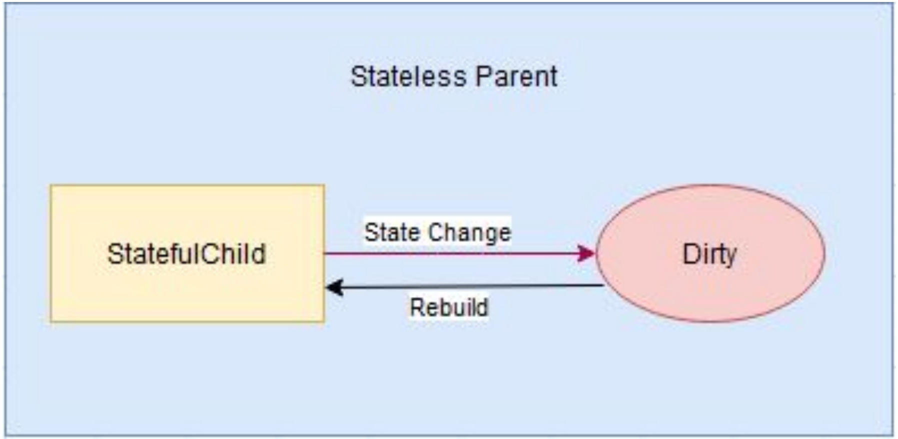
Reusable Components

- Custom Painting
  - CustomPaint - A Flutter Widget
  - CustomPainter - uses Canvas to draw 2d shapes



Stateful child of stateless parent

- Parent can be stateless while child can be stateful
- Allows to rebuild only stateful child on state change
- Use stateful custom widget to avoid making root widget stateful



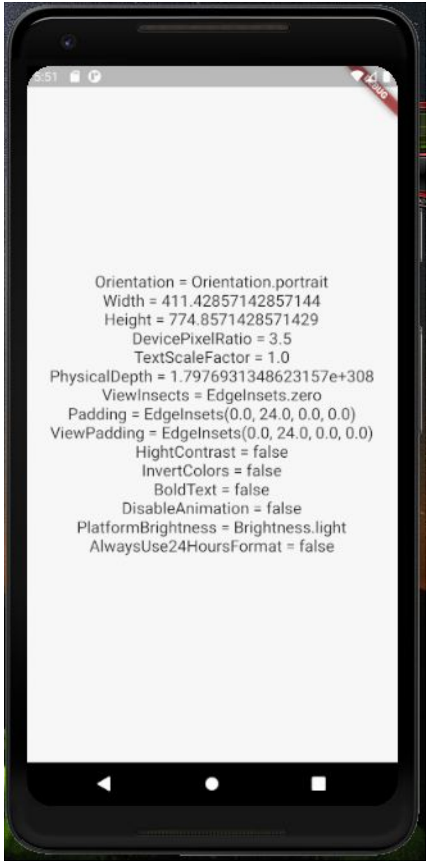
Responsive User Interface

Flutter support in making Responsive UI

- MediaQuery
- LayoutBuilder
- OrientationBuilder
- Other responsive widgets

MediaQuery

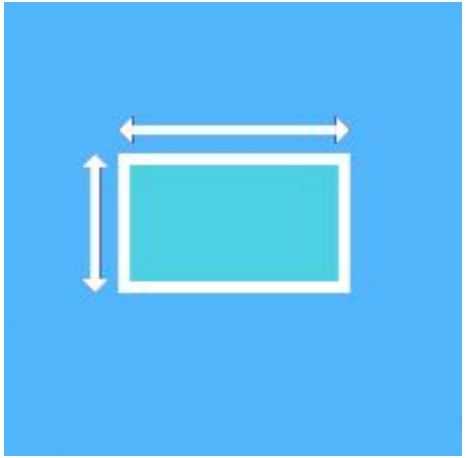
- Provides device data using **MediaQuery.of(context)** to allow make decisions on layout
    - Orientation
    - Size (in logical pixels)
    - devicePixelRatio
    - viewInsets
- Rebuilds widget on data change.



```
Orientation = Orientation.portrait
Width = 411.42857142857144
Height = 774.8571428571429
DevicePixelRatio = 3.5
TextScaleFactor = 1.0
PhysicalDepth = 1.7976931348623157e+308
ViewInsects = EdgeInsets.zero
Padding = EdgeInsets(0.0, 24.0, 0.0, 0.0)
ViewPadding = EdgeInsets(0.0, 24.0, 0.0, 0.0)
HightContrast = false
InvertColors = false
BoldText = false
DisableAnimation = false
PlatformBrightness = Brightness.light
AlwaysUse24HoursFormat = false
```

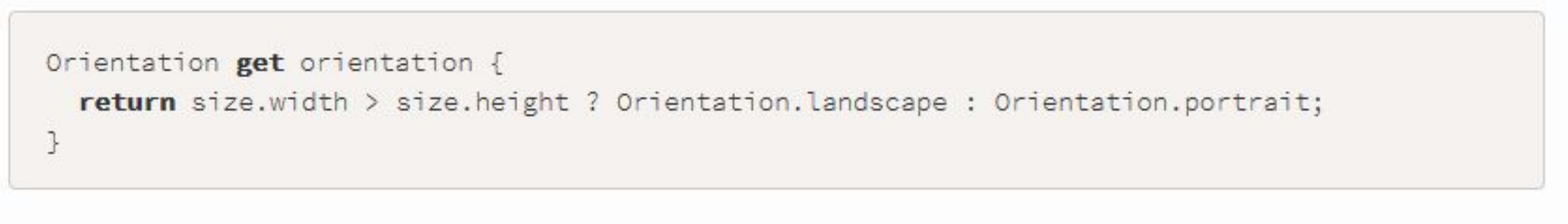
LayoutBuilder

- Used to define child widget size based on parent widget’s constraints
- Provides parent’s widget constraints as **BoxConstraints**
- BoxConstraints contains
  - minWidth
  - maxWidth
  - minHeight
  - maxHeight



OrientationBuilder

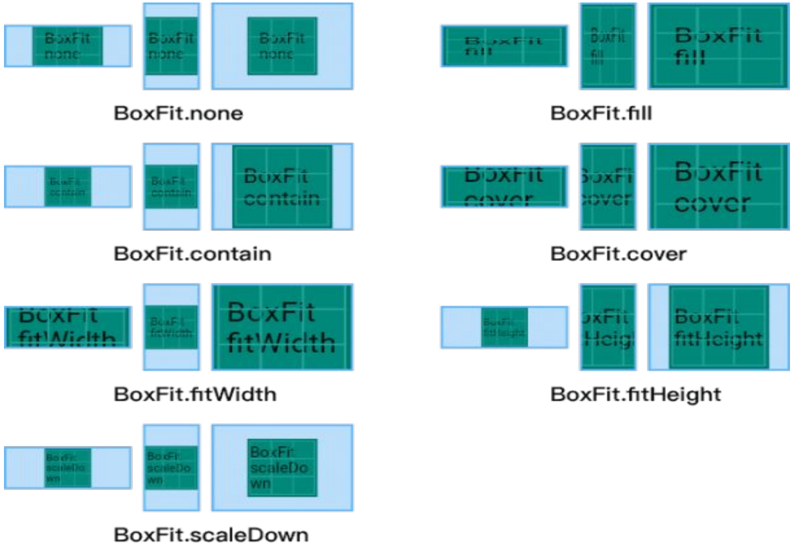
- Builds a widget tree that can depend on the parent widget's orientation (distinct from the device orientation).
- Orientation
  - Landscape: wider than tall
  - Portrait: taller than wide



```
Orientation get orientation {
  return size.width > size.height ? Orientation.landscape : Orientation.portrait;
}
```

Other responsive widgets

- FittedBox
  - Scales and positions its child within itself according to **fit**



BoxFit.none

BoxFit.fill

BoxFit.contain

BoxFit.cover

BoxFit.fitWidth

BoxFit.fitHeight

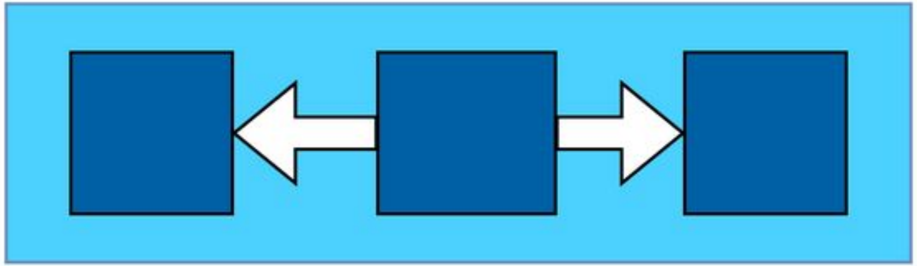
BoxFit.scaleDown

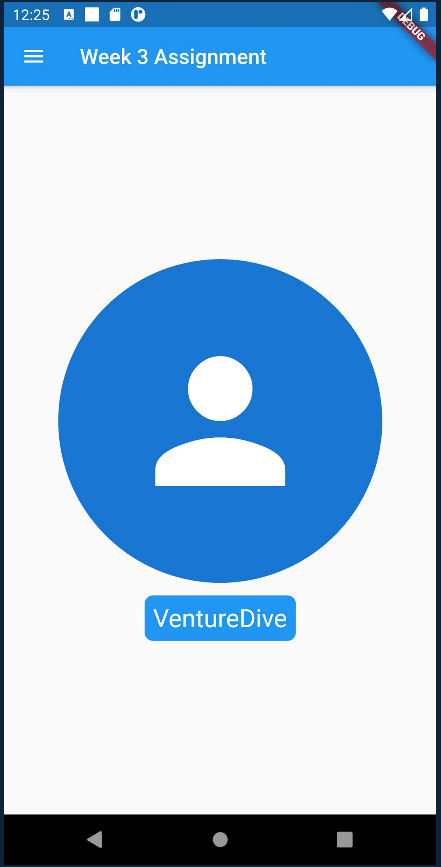
Other responsive widgets

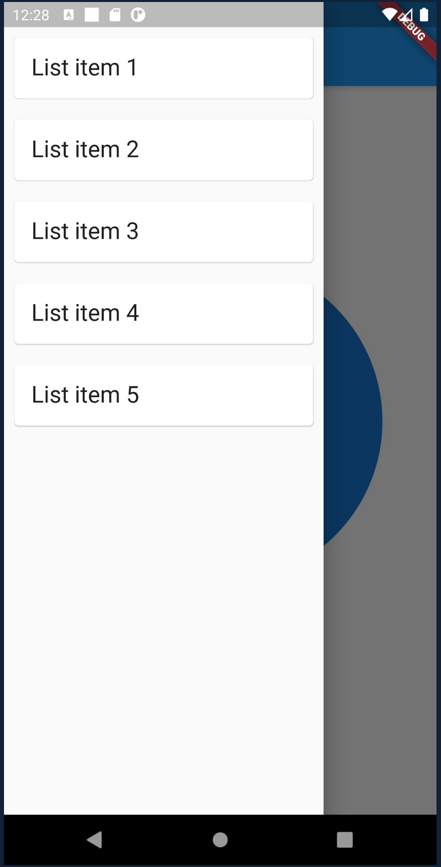
- Aspect Ratio
  - A widget that attempts to size the child to a specific aspect ratio.
  - Used when width and height should be proportional instead of actual values.

Other responsive widgets

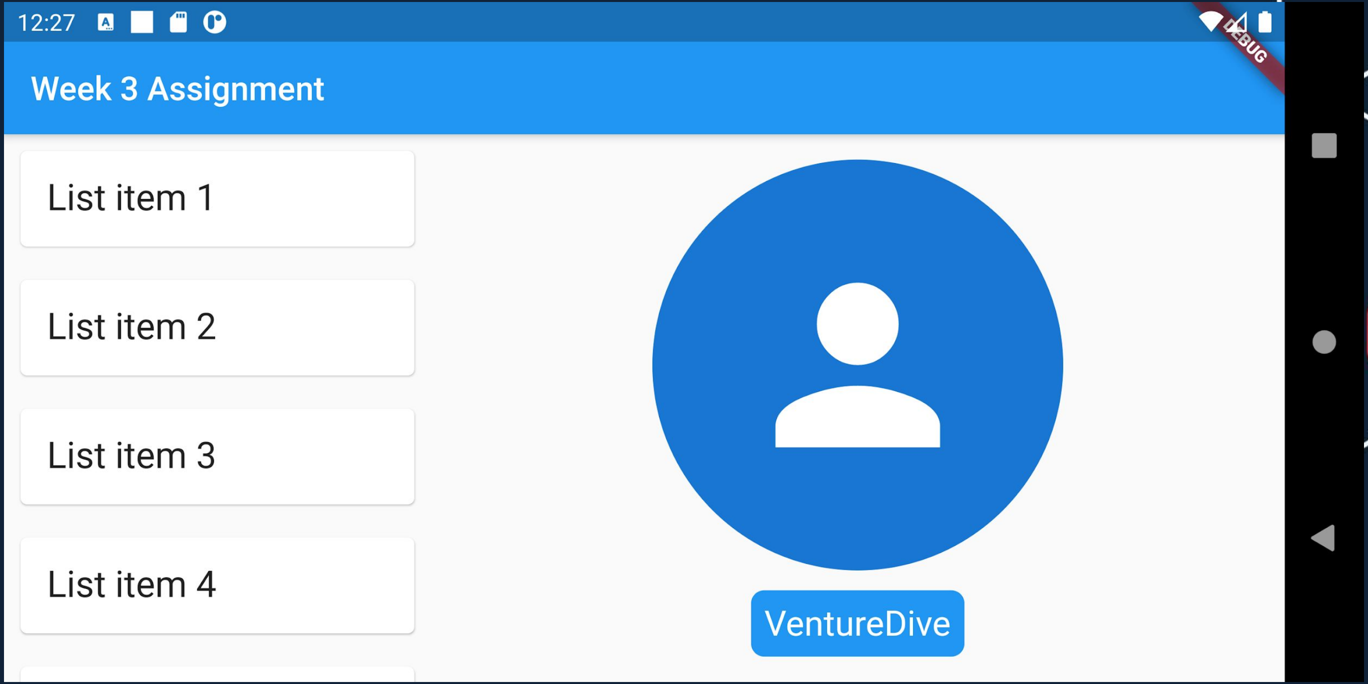
- Expanded
  - Helping you fill Rows and Columns.
  - Must be used under the **Flex** widget i.e. Row, Column, and Flex.
  - Use **flex** attribute to define to proportion of space the child should contain.
  - Gets the space after inflexible widgets are laid out







Take Home Assignment



Take Home Assignment

Assignment submission

Upload your code on github and submit it’s link on the Google chat group.

VentureDive

Thank you

Upcoming: Network Requests