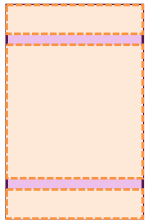
# Most Important Widgets

App / Page Setup

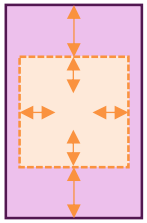
MaterialApp / CupertinoApp

Scaffold / CupertinoPageScaffold

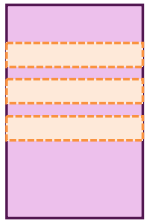


Layout

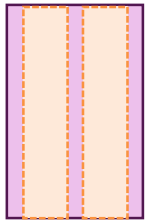
Container



Row



Column



ACADEMIND

# Details

## MaterialApp / CupertinoApp

- Typically the root widget in your app
- Does a lot of “behind-the-scenes” setup work for your app
- Allows you to configure a global theme for your app
- Sets up navigation behavior (e.g. animations) for your app

## Scaffold / CupertinoPageScaffold

- Typically used as a frame for a page in your app
- Provides a background, app bar, navigation tabs, etc
- Only use one scaffold per page!

ACADEMIND

# Details

## Container

- Extremely versatile widget!
- Can be sized (width, height, maxWidth, maxHeight), styled (border, color, shape, ...) and more
- Can take a child (but doesn't have to) which you also can align in different ways
- You'll use this widget quite often

## Row / Column

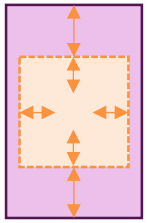
- Must-use if you need multiple widgets sit next to each other horizontally or vertically
- Limited styling options => Wrap with a Container (or wrap child widgets) to apply styling
- Children can be aligned along main-axis and cross-axis (see separate cheat sheet)

ACADEMIND

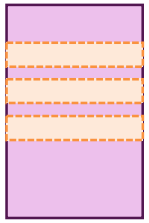
# Most Important Widgets

## Layout

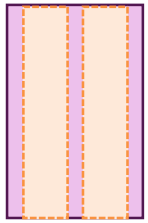
- Container



- Row

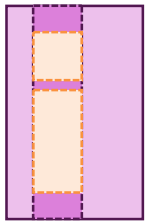


- Column

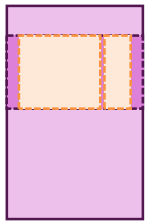


## Row / Column Children

- Flexible



- Expanded



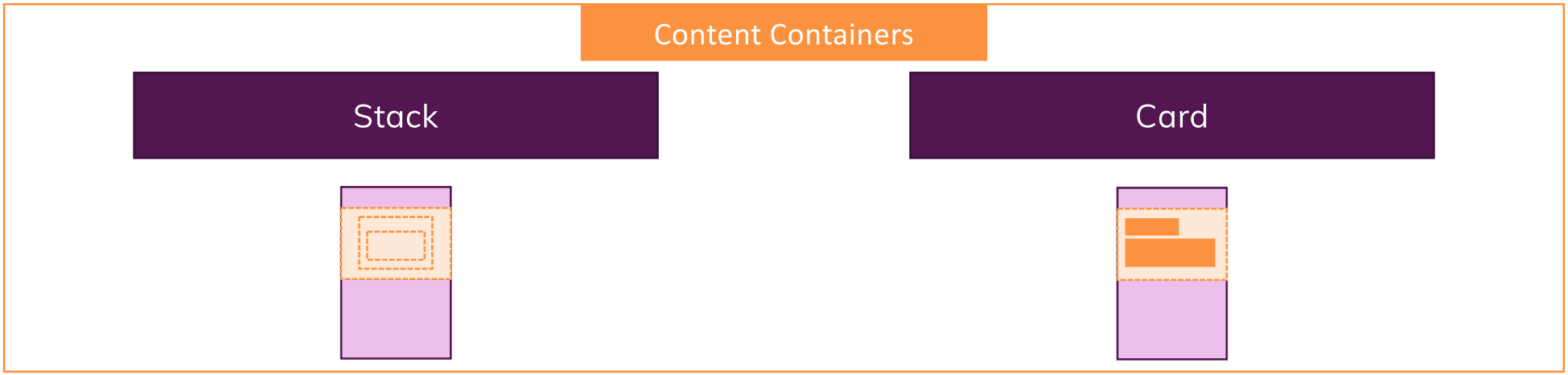
ACADEMIND

# Details

## Flexible / Expanded

- Also see separate cheat sheet attached to lecture about Row / Column / Flexible / Expanded
- Helps you size the child widgets of Row / Column

# Most Important Widgets



ACADEMIND

# Details

## Stack

- Used to position items on top of each other (along the Z axis)
- Widgets can overlap
- You can position items in absolute space (i.e. in a coordinate space) via the Positioned() widget

## Card

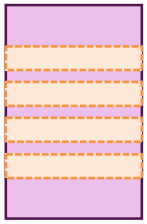
- A container with some default styling (shadow, background color, rounded corners)
- Can take one child (can be anything)
- Typically used to output a single piece / group of information

ACADEMIND

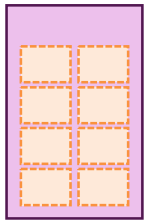
# Most Important Widgets

## Repeat Elements

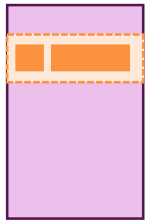
ListView



GridView

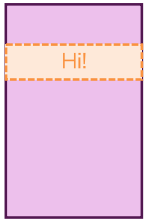


ListTile

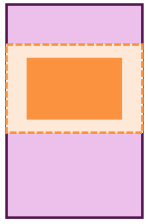


## Content Types

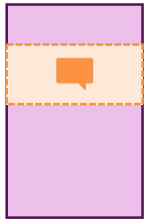
Text




Image



Icon



# Details

## ListView / GridView

- Used to output lists (or grids) of items
- Like a Column() but scrollable (Column is not)
- Can be laid out vertically (default) and horizontally
- Use ListView.builder() to get optimized item rendering for very long lists

## ListTile

- A pre-styled container / Row() that allows you to achieve a typical “list-item look”
- Offers various slots for widgets (e.g. at the beginning, a title, at the end)
- Not a must-use but can be handy for a default list-item look

ACADEMIND

# Details

## Text

- A widget that simply outputs some text on the screen
- Text can be styled (font family, font weight, font size etc)
- Text behavior can be controlled (e.g. clipping if it's too long)

## Image

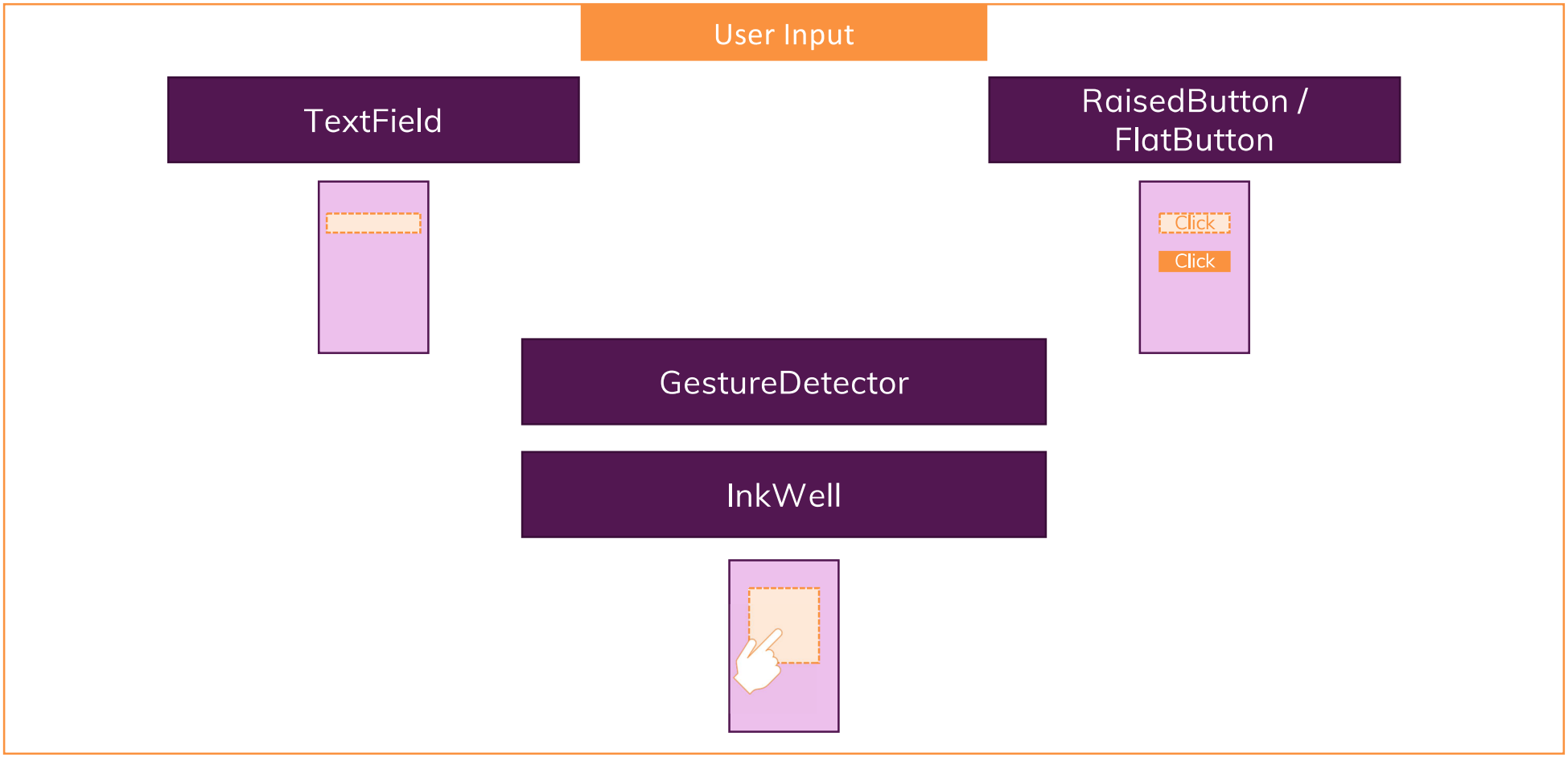
- Used to render an image on the screen
- Supports different sources (included in app, web image, ...)
- Can be configured to size itself in different ways into a wrapping container

## Icon

- Renders an Icon onto the screen
- Flutter ships with many default Material (Android) and iOS icons which you can use
- There also is an IconButton() widget in case you need a button with an icon

ACADEMIND

# Most Important Widgets



# Details

## TextField

- Renders an editable text field where the user can enter (type) information
- Many, many configuration options (e.g. autocorrection, error messages, labels, styles)
- Supports different kinds of keyboard (email, number, normal text, ...)

## RaisedButton / FlatButton / IconButton

- Differently styled buttons that handle user taps
- A custom function that should execute upon a tap can (and has to be) provided
- Can be styled / customized

## GestureDetector / InkWell

- GestureDetector allows you to wrap ANY widget with touch listeners (e.g. double tap, long tap)
- InkWell does the same but adds a visual ripple effect upon touches (effect can be configured)
- You can build your own buttons / touchable widgets with these widgets