Flutter:Stack and Positioned Widget:

The stack is a widget in Flutter that contains a list of widgets and positions them on top of the other. In other words, the stack allows developers ***to overlap multiple widgets into a single screen*** and renders them from bottom to top. Hence, the **first widget** is the **bottommost** item, and the **last widget** is the **topmost** item

Key Points Related to Stack Widget

The following are the key points of the Flutter stack widget:

- The child widget in a stack can be either **positioned** or **non-positioned**.
- Positioned items are wrapped in Positioned widget and must have a one non-null property
- The non-positioned child widgets are aligned itself. It displays on the screen based on the stack's alignment. The default position of the children is in the top left corner.
- We can use the alignment attribute to change the alignment of the widgets.
- Stack places the children widgets in order with the first child being at the bottom and the last child being at the top. If we want to reorder the children's widget, it is required to rebuild the stack in the new order. By default, the **first widget of each stack has the maximum size** compared to other widgets.

How to use a stack widget in Flutter?

The below example helps to understand the use of stack widget quickly that contains three containers of shrinking size:

```
1. Stack(
2.   children: <Widget>[
3.     // Max Size
4.     Container(
5.       color: Colors.green,
6.     ),
```

```
7.      Container(
8.        color: Colors.blue,
9.      ),
10.     Container(
11.       color: Colors.yellow,
12.     )
13.   ],
14.),
```

Properties of the Stack Widget

The following are the properties used with the stack widget:

alignment: It determines how the children widgets are positioned in the stack. It can be top, bottom, center, center-right, etc.

```
1. Stack(
2.   alignment: Alignment.topCenter, // Center of Top
3.   children: <Widget>[ ]
4. )
```

textDirection: It determines the text direction. It can draw the text either ltr (left to right) or rtl (right to the left).

```
1. Stack(
2.   textDirection: TextDirection.rtl, // Right to Left
3.   children: <Widget>[ ]
4. )
```

fit: It will control the size of non-positioned children widgets in the stack. **It has three types**: loose, expand and passthrough. The **loose** used to set the child widget small, the **expand** attribute makes the child widget as large as possible, and the **passthrough** set the child widget depending on its parent widget.

```
1. Stack(
2.   fit: StackFit.passthrough,
3.   children: <Widget>[ ]
4. )
```

overflow: It controls the children widgets, whether visible or clipped, when it's content overflowing outside the stack.

```
Stack(
  overflow: Overflow.clip, // Clip the Content
  children: <Widget>[ ]
)
```

clipBehavior: It determines whether the content will be clipped or not.

Positioned Widget:

It is not the stack parameter but can be used in the stack to locate the children widgets. The following are the constructor of the positioned stack:

```
const Positioned({
Key key,
this.left,
this.top,
this.right,
this.bottom,
this.width,
this.height,
@required Widget child,
```

```dart
Widget build(BuildContext context) {
  return Scaffold(
    body: SafeArea(
      child: Center(
        child: Stack(
          children: [
            Container(
              width: 400,
              height: 200,
              color: Colors.teal,
              child: const Center(
                child: Text('First widget'),
              ), // Center
            ), // Container
            Positioned(
              left: 20,
              top: 30,
              width: 150,
              height: 50,
              child: Container(
                // width: 100,
                // height: 50,
                color: Colors.yellow,
                child: const Center(child: Text('Second widget')),
              ), // Container
            ), // Positioned
            Positioned(
              width: 150,
              height: 50,
              left: 230,
              top: 30,
              child: Container(
                color: Colors.blue,
                child: const Center(child: Text('Third widget')),
              ), // Container
            ), // Positioned
          ],
        ), // Stack
```