

Container Decoration

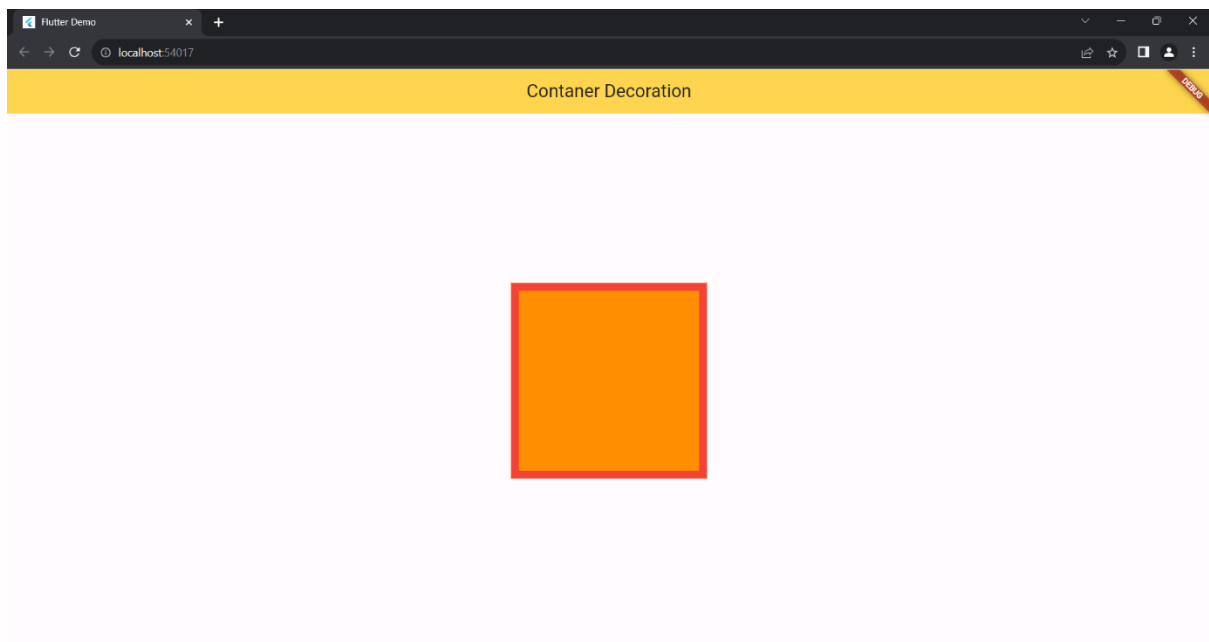
decoration:-

decoration property is used to apply visual styling to a widget, typically a Container. It takes a BoxDecoration object that defines various visual elements such as color, border

border:

```
Container(  
  height: 250,  
  width: 250,  
  decoration: BoxDecoration(  
    color: Colors.amber.shade800,  
    border: Border.all(color: Colors.red, width: 10)), // BoxDecoration  
), // Container
```

Output:

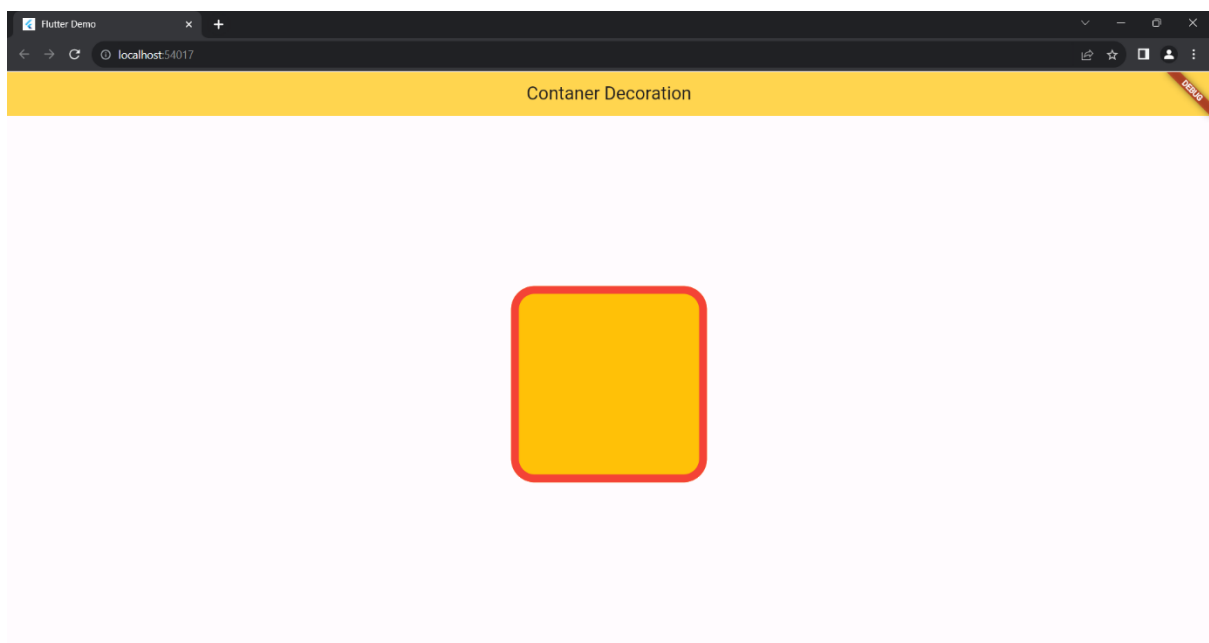


borderRadius:

To set the border to the Container we can use `Border.all()`, `Border.symmetric()`, `Border()`. We can set the border color, and border width using the `Border` class. To give the border to a particular side we can use the `Border()`.

```
ElevatedButton(  
  onPressed: () {  
    setState(() {  
      choiceAnsIndex = 0;  
    });  
  },  
  style: ElevatedButton.styleFrom(  
    backgroundColor: getColor(0), fixedSize: const Size(200, 30)),  
  child: Text(  
    allQuestions[questionIndex]["Options"][0],  
    style: const TextStyle(  
      fontSize: 17,  
      color: Colors.black,  
      fontWeight: FontWeight.w400), // TextStyle  
  )), // Text // ElevatedButton
```

Output:

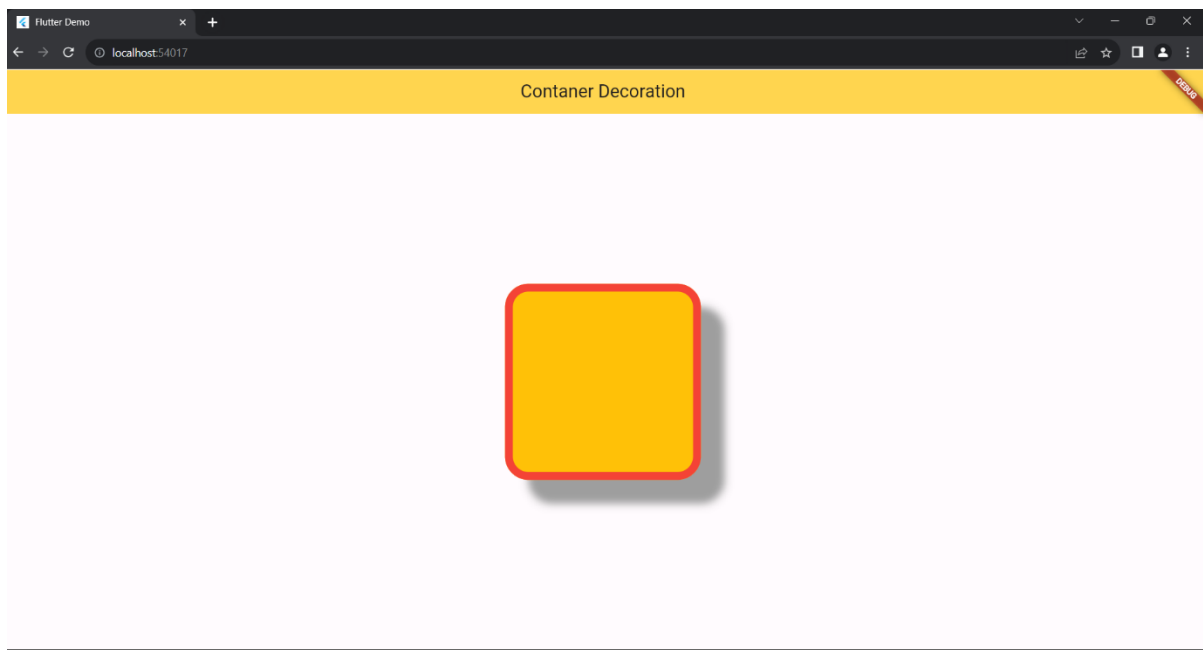


boxShadow:

It takes a List of BoxShadow. Using this we can display the shadow behind the container. We can specify the color, offset(Position) and blurRadius in the BoxShadow.

```
Container(  
  height: 250,  
  width: 250,  
  decoration: BoxDecoration(  
    color: Colors.amber,  
    borderRadius: const BorderRadius.all(  
      Radius.circular(30),  
    ), // BorderRadius.all  
    border: Border.all(color: Colors.red, width: 10),  
    boxShadow: const [  
      BoxShadow(  
        color: Colors.grey,  
        offset: Offset(30, 30),  
        blurRadius: 8), // BoxShadow  
    ],  
  ), // BoxDecoration  
), // Container
```

Output:



gradient:

Gradient refers to the transition between two or more colors. We can give a gradient to the Container using LinearGradient(), SweepGradient(), RadialGradient. In the Linear Gradient, we need to specify the List of colors to be used to display the gradient. We can also specify the position to place the color using the stops parameter. The values in the stop parameter range between 0.0 to 1.0. The values in the stops list must be in ascending order. If a value in the [stops] list is less than an earlier value in the list, then its value is assumed to equal the previous value.

```
Container(  
  height: 250,  
  width: 250,  
  decoration: BoxDecoration(  
    color: Colors.amber,  
    border: Border.all(color: Colors.black, width: 10),  
    gradient: const LinearGradient(  
      stops: [0.4, 0.6],  
      colors: [Colors.orange, Colors.green]  
    ) // LinearGradient  
  ), // BoxDecoration  
), // Container
```

Output:

