

Intro to Flutter

Flutter Layout Widget Definition

Layout widgets are used to control how other widgets appear visually. Single-child layout widgets can affect one child widget while multi-child layout widgets can affect many.

Flutter Center Widget

A Center widget can take a child widget and have it appear centered in the available space.

```
Center (  
  child: Text(  
    'Hello, world!'  
  )  
)
```

Flutter Align Widget

An Align widget is a more flexible version of the Center widget. Its alignment property can be used to align its child widget in multiple ways.

```
Align(  
  alignment: Alignment.topRight,  
  child: Text("In the top right")  
)
```

Flutter Row Widget

A Row widget is a multi-child layout widget designed to organize content in a horizontal row.

```
Row(  
  children: const [  
    Text('Left'),  
    Text('Right!')  
  ]  
)
```

Flutter Makeup

Flutter apps are defined by combinations of widgets, the core piece of content in an application. Widgets are created by calling their constructor with parameters defining the widget content.

Flutter Main Function

The base of a Flutter app typically has a `main` function with a call to `runApp` inside of it. The primary widget of the application, the root widget, is usually a `MaterialApp` with a `Scaffold` inside of it.

```
void main {  
  runApp(  
    const MaterialApp(  
      home: Scaffold()  
    )  
  );  
}
```

Flutter Text Widget

A Text widget displays text in an application, with the text to be displayed passed into the widget's constructor.

```
Text('Hello, world!')
```

Flutter Icon Widget

An Icon widget displays an image from a list of [Google provided options](#).

```
const myIcon = Icon(  
  Icons.check,  
  color: Colors.green,  
  size: 100.0,  
);
```

Flutter Supported Platforms

Flutter is a framework that allows for the creation of applications on many platforms such as:

- iOS
- Android
- Web
- Desktop

Flutter Benefits

Flutter has many benefits such as:

- Being free and open source
- Having fast reload times
- Cross platform capabilities
- Community resources

Flutter Container Widget

Container widgets paint a container box around a child widget. They have properties for specifying how the container box looks: what borders it has, what color it is, and how much spacing.

```
Container(  
  child: const Icon(Icons.image, size: 50)  
)
```

Container Decorater Parameter

The `decorator` parameter of the Container widget is able to control the border and colors of the `child` widget.

```
Container(  
  decoration: myDecoration,  
  child: const Icon(Icons.image, size: 50)  
)
```

Container Margin Parameter

Margins are used to specify the amount of empty space around a Container.

```
Container(  
  margin: const EdgeInsets.all(50)  
)
```

Container Padding Parameter

Padding is used to specify the amount of empty space around the child widgets within a Container.

```
Container(  
  child: Text('Hello There')  
  padding: const EdgeInsets.all(50)  
)
```

Container Transform Parameter

Transformations are used to visually transform a Container and its child widget(s), adding effects like rotation or skewing.

```
Container(child: const Text('Hello World'),  
  transform: Matrix4.skewX(0.3))
```