

Widget Layout

...

Layout

- Trong Flutter, chúng ta chỉ có Widget và sắp xếp chúng
- Tất cả các Widget cần được bọc bởi một Widget lớn hơn có tính tổ chức
- Layout trong Flutter chính là bao gồm một hệ thống phân cấp các Widget và ràng buộc chúng với nhau

Điểm chung của các loại Layout

- Tất cả các Layout sẽ chứa các Widget con trong tham số `child` ở constructor đối với các layout chỉ chứa một Widget duy nhất, như Container. Hay `children` đối với các Layout có thể chứa nhiều Widget con như Column và Row.
- Đối với `children`, thì giá trị của nó là một danh sách các Widget

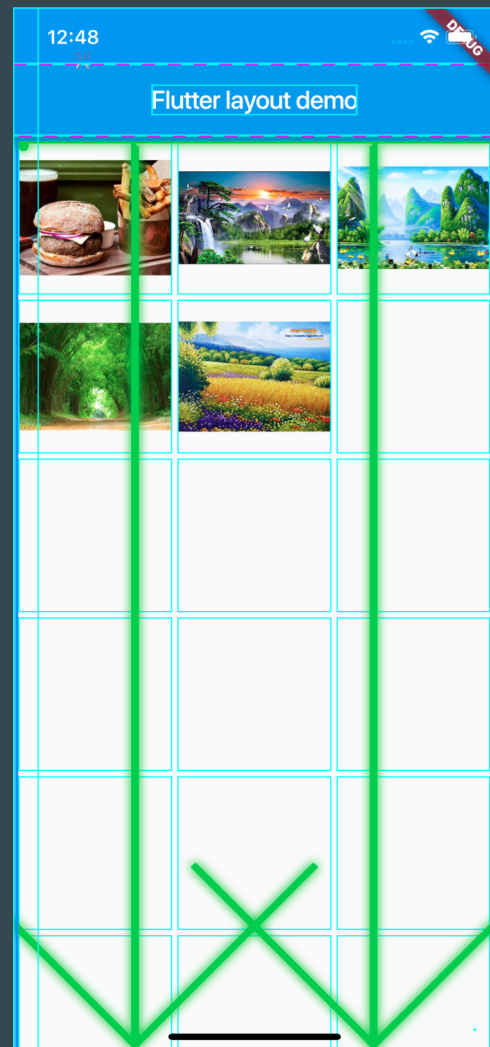
Nguyên tắc layout

Trước khi dựng một màn hình cần tính toán và dựng sơ đồ layout, chúng ta phải chia nhỏ được các thành phần:

- Xác định hàng và cột
- Layout có bao gồm dạng grid không?
- Có chứa các yếu tố chồng chéo không?
- Lưu ý các khu vực cần căn chỉnh, khoảng cách và đường viền

Virtual Rendering

```
debugPaintSizeEnabled = true
```



Xây dựng bố cục đơn giản

Container

- Có thể coi **container** là một Layout đơn giản, chỉ chứa đựng đúng **một Widget con**.
- Container đang là widget duy nhất có các thuộc tính margin, padding, constraint, width, height được chỉ định trong constructor

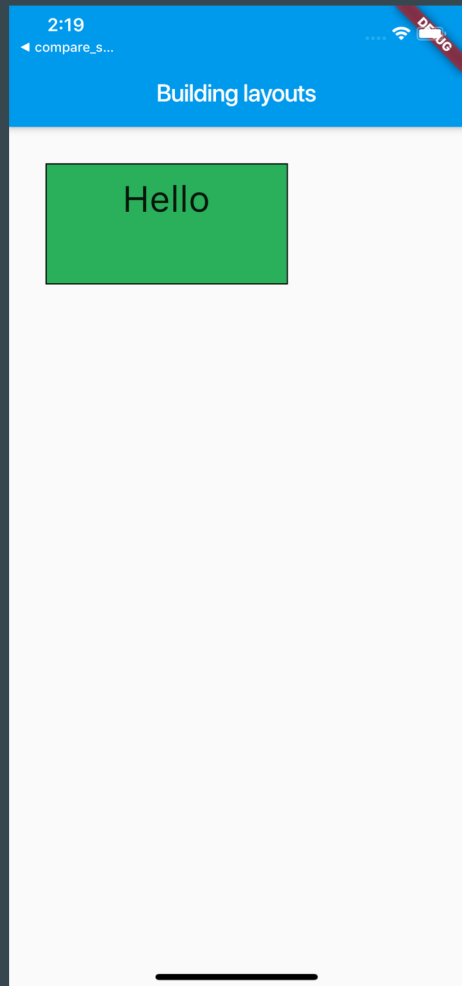
Lưu ý với Container

- Lưu ý:
- + Nếu container không có các thuộc tính định vị với bên ngoài như margin, constraint hay bên trong như padding, và cũng không được định width, height thì khi render, kích thước của nó sẽ nhỏ nhất có thể
- + Nếu container có thuộc tính constraint theo parent Widget chứa nó, thì nó sẽ dẫn ra cho khớp constraints được định.
- + Nếu sử dụng quá nhiều container có thể gây cảm giác giật, lag trên thiết bị có cấu hình yếu, vì nó sẽ mất khá nhiều thời gian tính toán so với các loại layout khác


```
Container(  
  color: Colors.green,  
),
```



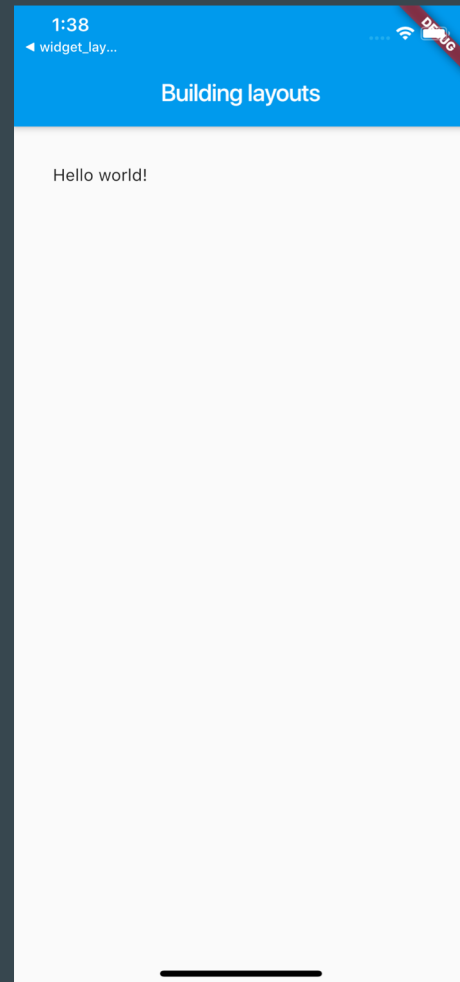
```
Widget myLayoutWidget() {  
  return Container(  
    margin: EdgeInsets.all(30.0),  
    padding: EdgeInsets.all(10.0),  
    alignment: Alignment.topCenter,  
    width: 200,  
    height: 100,  
    decoration: BoxDecoration(  
      color: Colors.green,  
      border: Border.all(),  
    ),  
    child: Text("Hello", style: TextStyle(fontSize: 30)),  
  );  
}
```



Padding

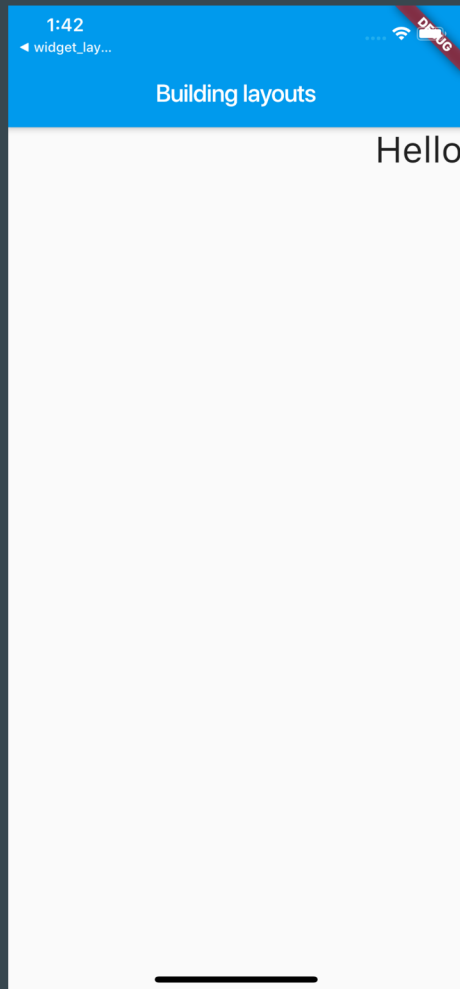
- Padding (có thể được hiểu là một dạng thu gọn của Container) với chỉ có một thuộc tính `padding`.
- Param của padding là một instance của `EdgeInsetsGeometry`
- Padding sẽ định nghĩa ra `paddingLeft`, `paddingTop`, `paddingRight`, `paddingBottom`

```
Widget myLayoutWidget() {  
  return Padding(  
    padding: EdgeInsets.all(32.0),  
    child: Text("Hello world!"),  
  );  
}
```



Align

```
Widget myLayoutWidget() {  
  return Align(  
    alignment: Alignment.topRight,  
    child: Text(  
      "Hello",  
      style: TextStyle(fontSize: 30),  
    ),  
  );  
}
```



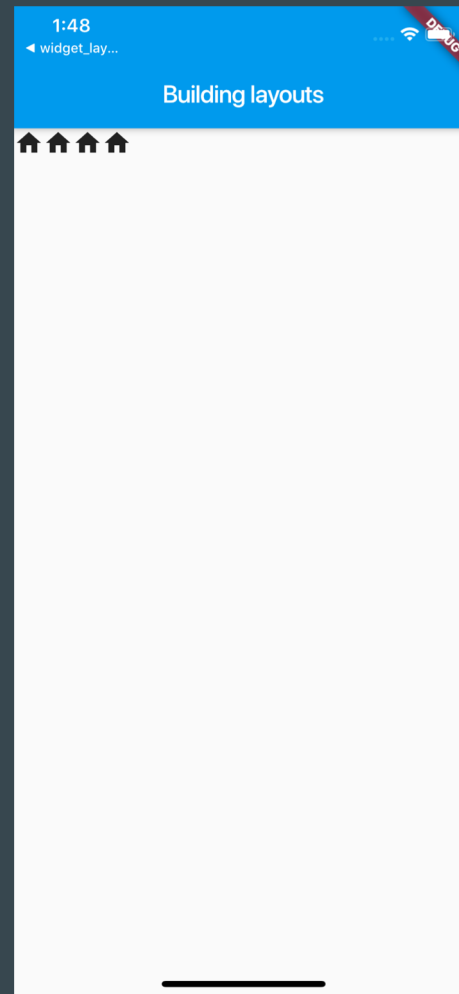
Transform

- Transform có nhiệm vụ thay đổi transform của Widget con theo chỉ định, nó có thể thay đổi vị trí theo tọa độ (translate), xoay (rotate), co giãn (scale).

```
Container(  
  color: Colors.black,  
  child: Transform(  
    alignment: Alignment.topRight,  
    transform: Matrix4.skewY(0.3)..rotateZ(-math.pi / 12.0),  
    child: Container(  
      padding: const EdgeInsets.all(8.0),  
      color: const Color(0xFFE8581C),  
      child: const Text('Apartment for rent!'),  
    ),  
  ),  
)
```

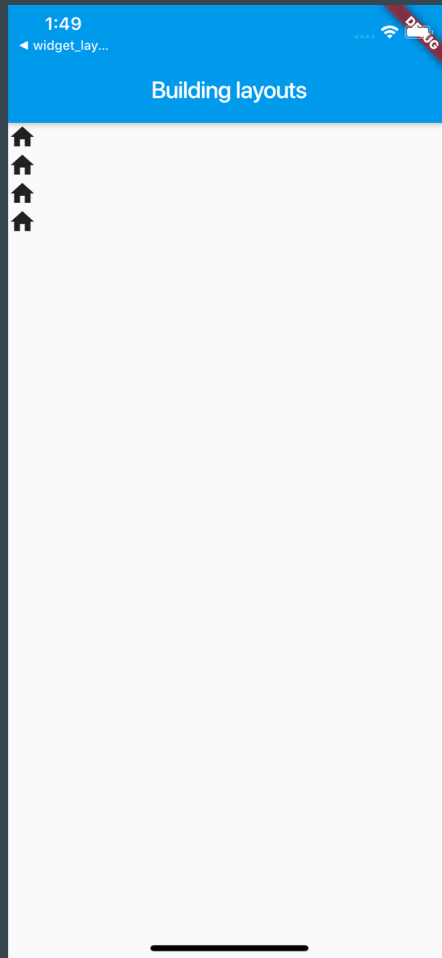
Row

```
Widget myLayoutWidget() {  
  return Row(  
    children: [  
      Icon(Icons.home),  
      Icon(Icons.home),  
      Icon(Icons.home),  
      Icon(Icons.home),  
    ],  
  );  
}
```



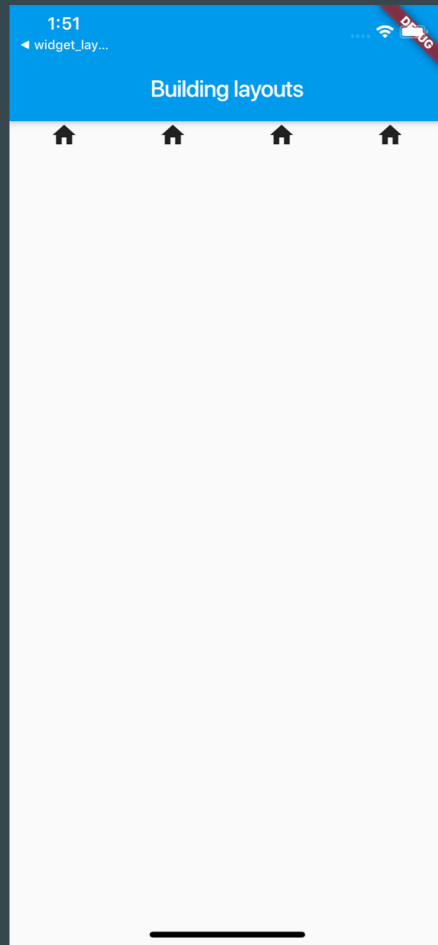
Column

```
Widget myLayoutWidget() {  
  return Column(  
    children: [  
      Icon(Icons.home),  
      Icon(Icons.home),  
      Icon(Icons.home),  
      Icon(Icons.home),  
    ],  
  );  
}
```



Expanded

```
Widget myLayoutWidget() {  
  return Row(  
    children: [  
      Expanded(child: Icon(Icons.home)),  
      Expanded(child: Icon(Icons.home)),  
      Expanded(child: Icon(Icons.home)),  
      Expanded(child: Icon(Icons.home)),  
    ],  
  );  
}
```

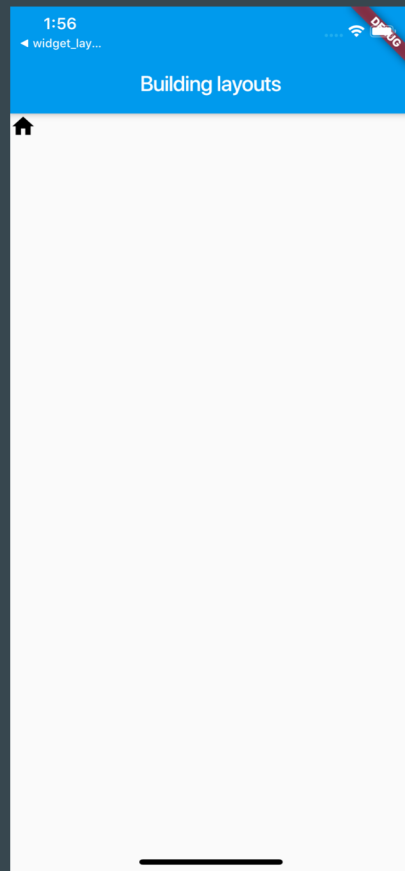


```
Widget myLayoutWidget() {  
  return Row(  
    children: [  
      Expanded(  
        flex: 7,  
        child: Container(  
          color: Colors.green,  
        ),  
      ),  
      Expanded(  
        flex: 3,  
        child: Container(  
          color: Colors.yellow,  
        ),  
      ),  
    ],  
  );  
}
```

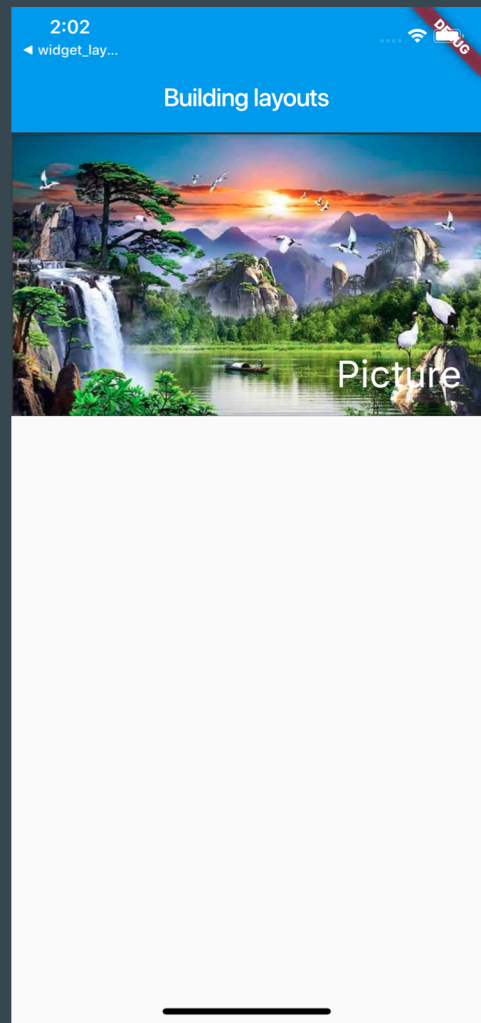


Stacks

```
Widget myLayoutWidget() {  
  return Stack(  
    children: [  
      Icon(Icons.home),  
      Icon(Icons.home),  
      Icon(Icons.home),  
      Icon(Icons.home),  
    ],  
  );  
}
```



```
Widget myLayoutWidget() {  
  return Stack(  
    alignment: Alignment.bottomRight,  
    children: [  
      Image.asset('images/1.jpg'),  
      Padding(  
        padding: EdgeInsets.all(16.0),  
        child: Text(  
          'Picture',  
          style: TextStyle(fontSize: 30, color:  
Colors.white),  
        ),  
      ),  
    ],  
  );  
}
```



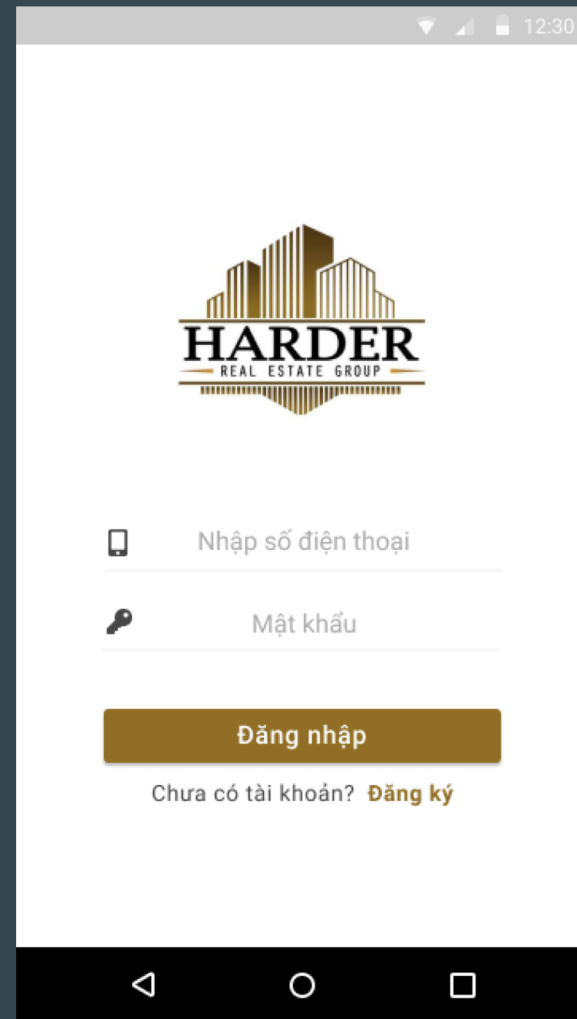
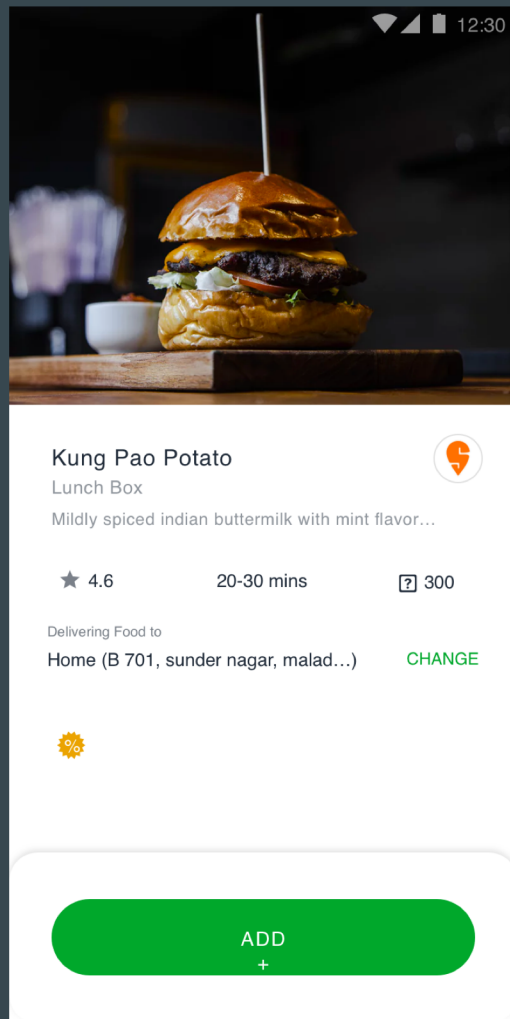
Các widget layout khác

`ListView`: Widget này cuộn các hàng hoặc cột với nội dung lớn hơn kích thước vật lý của màn hình

`GridView`: Widget này cuộn nội dung được đặt trong một lưới các hàng và cột.


`Scaffold`: Widget này được cung cấp bởi `material package`. Nó cung cấp một cách dễ dàng để thêm `AppBar`, `FloatingActionButton`, `Drawer`, `bottomNavigationBar`, `SnackBar`, v.v.

Bài tập:



Bài tập về nhà:

12:30



Họ và tên

Số điện thoại

Mật khẩu

Nhập lại mật khẩu


Đăng ký

Đã có tài khoản? **Đăng nhập**

12:30

Hồ sơ

Cập nhật



Đổi ảnh đại diện

Họ và tên

Hàn Duy

Số điện thoại

0902 209 011

Địa chỉ

14 gõ 6 Phạm Tuấn Tài - Quận Tân Phú - Thành phố Hồ Chí Minh