

一、课前准备

- Flutter环境

二、课堂目标

- 完成掘金社区APP首页

三、知识要点

- Flutter App项目的基本结构
- Scaffold
- AppBar
- Tabbar
- Row
- TextField
- Button (RaisedButton, FlatButton, TextButton)
- ListView
- 圆形或圆角图片的显示
- http请求
 - 内置方法
 - 第三方库的使用

- 本地图片，网络图片

四、总结

- Flutter框架在使用上和前端开发习惯还是有很大不同的，但理解规则后也不难上手

五、作业

- 使用ListView.builder实现两个ListView，一个ListView纵向滚动，第一个item为ListView，其余Item为文本，显示序号。一个ListView横向滚动，数量为10。



这些开源项目，让你轻松应对十大工作场景

532赞 · 33评论 · 阿宝哥



三分钟给女票写个“彩虹屁”bot | 🏆 技术专题第七期征文

128赞 · 76评论 · 义义又又



天明夜尽

CSS/GitHub

10 个 GitHub 上超火的 CSS 奇技淫巧项目，找到写 CSS 的灵感！

一般人没事的时候刷刷朋友圈、微博、电视剧、知乎，而有些人是没事的时候刷...



👍 51 💬 3



阿里巴巴TXD

前端/数据可视化

开眼界！阿里用这样的组件开发大屏（三）

Hi~ 我是前端学徒业枫 (@Malpor)，大



首页



沸点



发现



小册



我

Scaffold

App标准布局模板/脚手架

```
/// Creates a visual scaffold for material design widgets.
```

```
//主要参数
```

```
const Scaffold({  
  Key? key,  
  this.appBar, //顶部导航条  
  this.body, //页面内容区  
  this.floatingActionButton,  
  this.floatingActionButtonLocation,  
  this.floatingActionButtonAnimator,  
  this.persistentFooterButtons,  
  this.drawer, //左侧抽屉  
  this.endDrawer, //右侧抽屉  
  this.bottomNavigationBar, //底部tab导航  
  this.bottomSheet,  
  this.backgroundColor,  
  this.resizeToAvoidBottomPadding,  
  this.resizeToAvoidBottomInset,  
  this.primary = true,  
  this.drawerDragStartBehavior =  
DragStartBehavior.start,  
  this.extendBody = false,  
  this.extendBodyBehindAppBar = false,  
})
```

```
this.drawerScrimColor,  
this.drawerEdgeDragWidth,  
this.drawerEnableOpenDragGesture = true,//从屏幕边缘滑入手势  
this.endDrawerEnableOpenDragGesture = true,
```

? Scaffold 的 body 是 Scaffold

```
//drawer  
Container(width: 120,color:  
Colors.white,child:Center(child:Text("哈哈"))))
```

AppBar



```
this.leading,          //标题左侧按钮，比如返回。优先级高
this.title,            //标题栏
this.actions,          //标题右侧按钮
this.bottom,
this.shadowColor,
this.backgroundColor,
this.bottomOpacity = 1.0,
this.leadingWidth,
```

Tabbar + TabBarView

```
TabController _tabBarController;

List<Widget> tabs=[];
List<Widget> tabViews=[];

@override
void initState() {
  super.initState();

  List tabTitles = ["推荐", "热榜", "前端", "后端", "Android", "iOS", "人工智能", "开发工具", "代码人生", "阅读"];
  for(int i = 0 ;i<tabTitles.length;i++){
    tabs.add(Tab(text: tabTitles[i]));

    tabViews.add(Center(child:Text(tabTitles[i])));
  }
}
```

```
}

    _tabBarController = TabController( initialIndex:
0,length: tabTitles.length,vsync: this);
}
```

TabController

- with SingleTickerProviderStateMixin

更简单的方式

```
DefaultTabController(
  length:10,
  child:Scalffold
)
```

Row & Column

Row 组件是不可以滚动的，所以在 Row 组件中一般不会放置过多子组件，如果需要滚动的话应该考虑使用 ListView。

如果需要垂直展示，应该考虑 Column 组件。

如果只有一个子组件的话，应该考虑使用使用 Align 或者 Container 组件来定位这个子组件。

`crossAxisAlignment`: 子组件沿着 Cross 轴（在 Row 中是纵轴）如何摆放，其实就是子组件对齐方式，可选值有：

`CrossAxisAlignment.start`: 子组件在 Row 中顶部对齐。

`CrossAxisAlignment.end`: 子组件在 Row 中底部对齐。

`CrossAxisAlignment.center`: 子组件在 Row 中居中对齐。

`CrossAxisAlignment.stretch`: 拉伸填充满父布局。

`CrossAxisAlignment.baseline`: 在 Row 组件中会报错。

`mainAxisAlignment`: 子组件沿着 Main 轴（在 Row 中是横轴）如何摆放，其实就是子组件排列方式，可选值有：

`MainAxisAlignment.start`: 靠左排列。

`MainAxisAlignment.end`: 靠右排列。

`MainAxisAlignment.center`: 居中排列。

`MainAxisAlignment.spaceAround`: 每个子组件左右间隔相等，也就是 `margin` 相等。

`MainAxisAlignment.spaceBetween`: 两端对齐，也就是第一个子组件靠左，最后一个子组件靠右，剩余组件在中间平均分散排列。

`MainAxisAlignment.spaceEvenly`: 每个子组件平均分散排列，也就是宽度相等。

`children`: 子组件，值是一个 `Widget` 数组。

Expanded（占满剩余空间）

```
Row(children: [Text("哈哈"),Expanded(child:Text("嘻嘻"),textAlign: TextAlign.center,))])
```

TextField

```
controller: controller,  
maxLength: 30, //最大长度，设置此项会让TextField右下角有一个输入数量的统计字符串  
maxLines: 1, //最大行数  
autocorrect: true, //是否自动更正  
autofocus: true, //是否自动对焦  
obscureText: true, //是否是密码  
textAlign: TextAlign.center, //文本对齐方式  
style: TextStyle(fontSize: 30.0, color: Colors.blue), //输入文本的样式  
inputFormatters: [WhitelistingTextInputFormatter.digitsOnly], //允许的输入格式  
onChanged: (text) { //内容改变的回调  
  print('change $text');  
},  
onSubmitted: (text) { //内容提交(按回车)的回调
```

```
print('submit $text');  
},  
enabled: true, //是否禁用
```

TextField

Button (RaisedButton, FlatButton, TextButton)

InputDecoration

```
decoration: InputDecoration(  
  counterText: "", //计数文字  
  hintText: "搜索掘金", //placeholder  
  prefixIcon: Icon(Icons.search), //前面的图表  
  border: OutlineInputBorder(  
    borderRadius:  
      BorderRadius.circular(15), //圆角  
    borderSide: BorderSide.none),  
  fillColor: Color.fromARGB(255, 238, 241,  
    244), //背景色  
  filled: true  
)
```

ListView

这是我们会经常打交道的一个控件

```
Axis scrollDirection = Axis.vertical,  
bool reverse = false,  
ScrollController? controller,  
bool? primary,  
ScrollPhysics? physics,  
bool shrinkWrap = false,  
EdgeInsetsGeometry? padding,  
this.itemExtent,  
bool addAutomaticKeepAlives = true,  
bool addRepaintBoundaries = true,  
bool addSemanticIndexes = true,  
double? cacheExtent,  
List<Widget> children = const <Widget>[],  
int? semanticChildCount,  
DragStartBehavior dragStartBehavior =  
DragStartBehavior.start,  
ScrollViewKeyboardDismissBehavior  
keyboardDismissBehavior =  
ScrollViewKeyboardDismissBehavior.manual,  
String? restorationId,  
Clip clipBehavior = Clip.hardEdge,
```

构建ListView的三种方式

- ListView() 使用场景：少量数据
- ListView.build() 使用
- ListView.separated 自带分割线

圆形或圆角图片的显示

```
Container(  
  decoration: ShapeDecoration(  
    image: DecorationImage(  
      image: AssetImage("images/landscape1.jpeg"),  
      fit: BoxFit.fitWidth),  
    shape: RoundedRectangleBorder(  
      borderRadius:  
BorderRadiusDirectional.circular(20))),  
  width: double.maxFinite,  
  height: 300,  
  child: Align(  
    child: Padding(  
      padding: const EdgeInsets.all(8.0),  
      child: Text(  
        "Container decoration实现圆角(radius =  
20)",  
        style: TextStyle(color: Colors.white),
```

```
    ),  
    ),  
    alignment: Alignment.bottomCenter,  
  ),  
)
```

```
ClipRRect(  
  borderRadius: BorderRadius.circular(20),  
  child: Image.asset("images/landscape2.jpeg"),  
),
```

HttpClient

```
var url =  
  'https://apinew.juejin.im/recommend_api/v1/article  
  /recommend_all_feed';  
  
var httpClient = new HttpClient();  
var request = await  
httpClient.postUrl(Uri.parse(url));  
  
//post传参  
Map postData = {  
  'client_type': '2608',  
  'cursor': '0',  
  'id_type': '2',  
  'limit': '20',
```

```
        'sort_type': '200'  
    };  
  
    request.add(utf8.encode(json.encode(postData)));  
  
    var response = await request.close();  
    var responseBody = await  
response.transform(utf8.decoder).join();  
    //转json  
    listData = jsonDecode(responseBody);
```

作业

使用ListView.builder实现两个ListView，一个ListView纵向滚动，第一个item为ListView，其余Item为文本，显示序号。一个ListView横向滚动，数量为10。

暗号：面向工资编程