

实验二：添加组件

1、代码情况

```
//Index页面
import router from '@ohos.router'; // 导入路由模块

@Entry // 标记为入口组件
@Component // 声明为自定义组件
struct Index { // 定义Index组件
  @State message: string = 'Index页面'; // 响应式状态变量

  build() { // 构建UI方法
    Row() { // 横向布局容器开始
      Column() { // 纵向布局容器开始
        Text(this.message) // 显示文本组件
          .fontSize(50) // 设置字体大小
          .fontWeight(Fontweight.Bold) // 设置字体加粗

        Button('跳转') // 创建按钮组件
          .fontSize(40) // 设置按钮字号
          .onClick(() => { // 绑定点击事件
            router.pushUrl({ // 调用路由跳转
              url: 'pages/Second', // 目标页面路径
              params: { // 传递参数对象
                src: 'Index页面传来的数据' // 自定义参数值
              } // 参数对象结束
            }); // pushUrl调用结束
          }) // onClick事件结束
      } // Column纵向布局结束
    } // Row横向布局结束
  } // build方法结束
} // Index组件定义结束
```

```
// Second页面
import router from '@ohos.router';

interface PageParams { src: string; }

@Entry
@Component
struct Second {
  @State message: string = '功能页面';
  @State src: string = (router.getParams() as PageParams).src;

  // 时间组件状态
  @State currentTime: string = new Date().toLocaleTimeString();

  // 记事本状态
  @State noteText: string = '';
  @State savedNotes: string[] = [];
```

```

// 计时器状态
@State isTiming: boolean = false;
@State elapsedSeconds: number = 0;
private timerID: number = 0;

aboutToAppear(): void {
    setInterval(() => {
        this.currentTime = new Date().toLocaleTimeString();
    }, 1000);
}

// 计时器控制方法
private toggleTimer(): void {
    if (this.isTiming) {
        clearInterval(this.timerID);
    } else {
        this.timerID = setInterval(() => {
            this.elapsedSeconds += 1;
        }, 1000);
    }
    this.isTiming = !this.isTiming;
}

build() {
    scroll() {
        column() {
            Text(this.src)
                .fontSize(25)
                .margin(15);

            Text(`当前时间: ${this.currentTime}`)
                .fontSize(25)
                .margin(20);

            // 布局
            TextInput({ text: this.noteText })
                .onChange((value: string) => {
                    this.noteText = value;
                })
                .fontSize(28)
                .margin(20)
                .width('80%');

            Button('保存笔记')
                .onClick(() => {
                    this.savedNotes = this.savedNotes.concat([this.noteText]);
                    this.noteText = '';
                })
                .fontSize(28)
                .margin(5);

            column() {
                ForEach(
                    this.savedNotes,
                    (item: string) => {

```

```

        Text(`• ${item}`)
            .fontSize(20)
            .margin(5);
    },
    (item: string) => item
);
}
.margin(15);

Text(`已计时: ${Math.floor(this.elapsedSeconds/60)}分
${this.elapsedSeconds%60}秒`)
    .fontSize(28)
    .margin(10);
Button(this.isTiming ? '停止' : '开始')
    .onClick(() => this.toggleTimer())
    .fontSize(28)
    .margin(10)
    .width(100);

Button('返回Index页面')
    .fontSize(35)
    .onClick(() => router.back())
    .margin(20);

// 每日心语
Text('今天也是元气满满的一天! ')
    .fontSize(28)
    .fontColor(0x666666)
    .margin({ top: 20 })
}
.width('100%');
}
}
}

```

2、截图运行页面





3、介绍组件

(1) 时间组件

能够实时显示当前时间



当前时间: 上午12:18:29



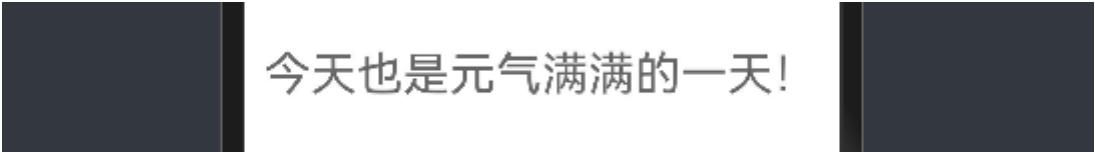
(2) 记事本组件
可以在上面记下想记录的事情或者提醒自己需要做的事情



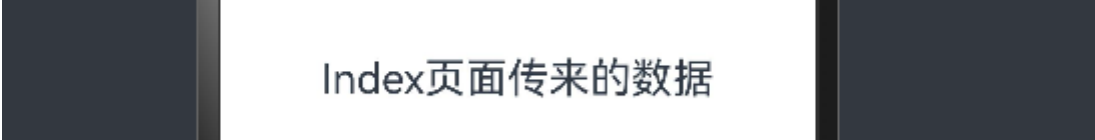
(3) 计时器组件
能够进行计时



(4) 每日心语组件
写下一句话给自己, 让自己拥有一天好心情



(5) 传递参数组件
实验一做过的保留下来的组件



(6) 跳转返回按钮
实验一做过的保留下来的组件

