

Advanced web technology

# 高级Web技术

*RIA技术—概述*

# Rich Internet Application技术

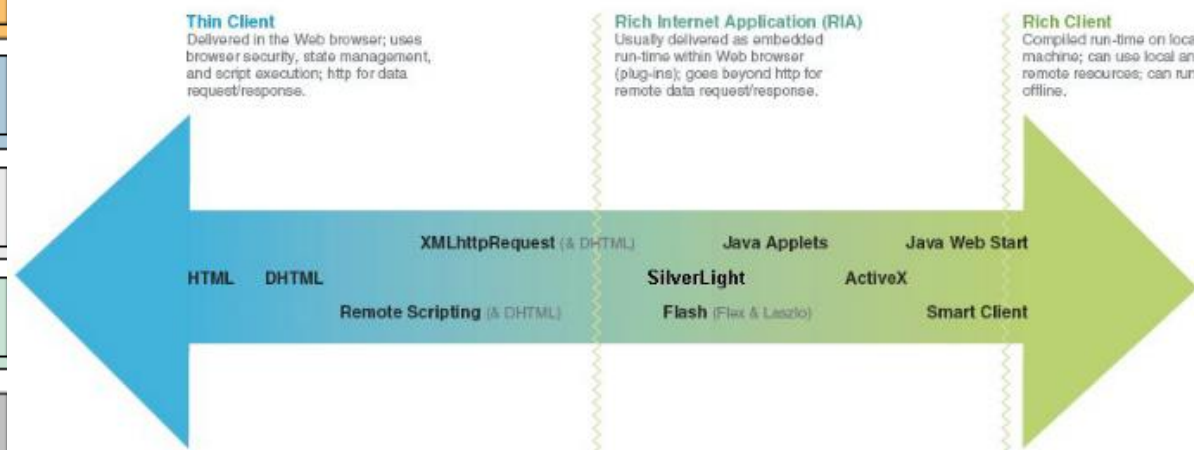
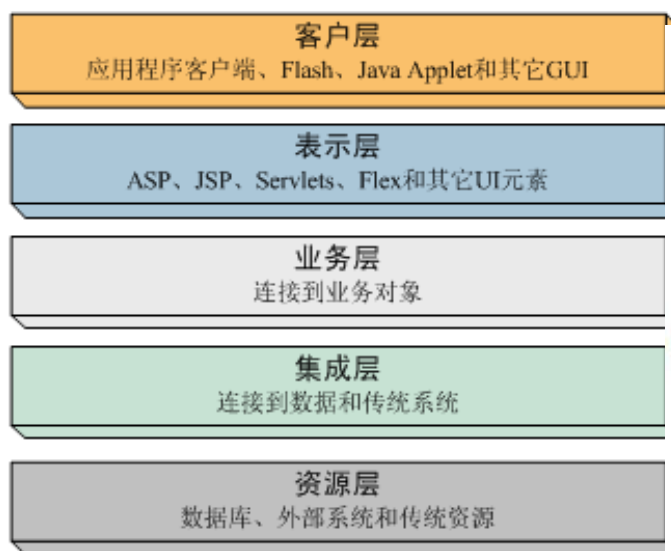
## ■ Web的不足

- 现在的**Web**应用程序对完成复杂应用方面却始终跟不上步伐
- **Web**模型是基于页面的模型，缺少客户端智能机制
- 几乎无法完成复杂的用户交互（如传统的**C/S**应用程序和桌面应用程序中的用户交互）



# RIA的应用程序模型

- RIA中的 Rich Client（丰富客户端）提供可承载已编译客户端应用程序（以文件形式，用HTTP传递）的运行环境，客户端应用程序使用异步客户/服务器架构连接现有的后端应用服务器



# Rich Internet Application技术

## ■ RIA的优势

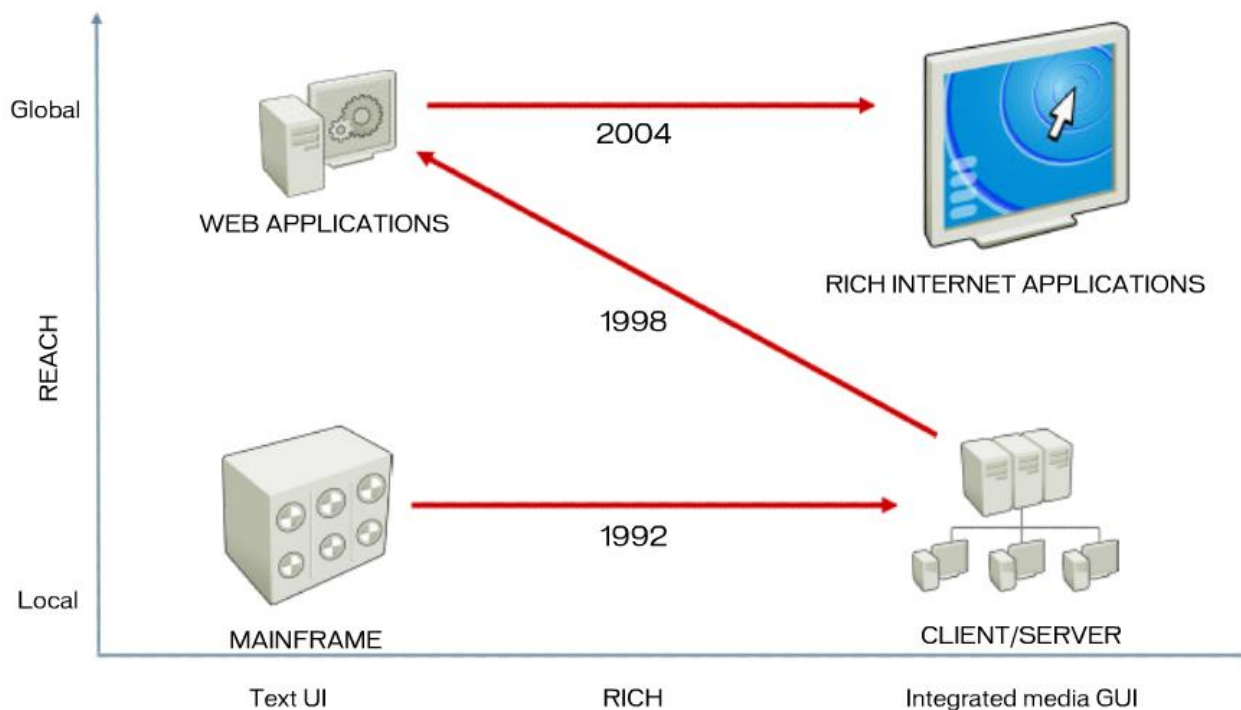
- **数据模型的丰富**
  - 用户界面可以显示和操作更为复杂的嵌入在客户端的数据模型
- **用户界面的丰富**
  - 提供了灵活多样的界面控制元素，这些控制元素可以很好的与数据模型相结合
- **最好的通讯模式**
  - 无刷新页面之下提供快捷的界面响应时间
  - 双向互动声音和图像。



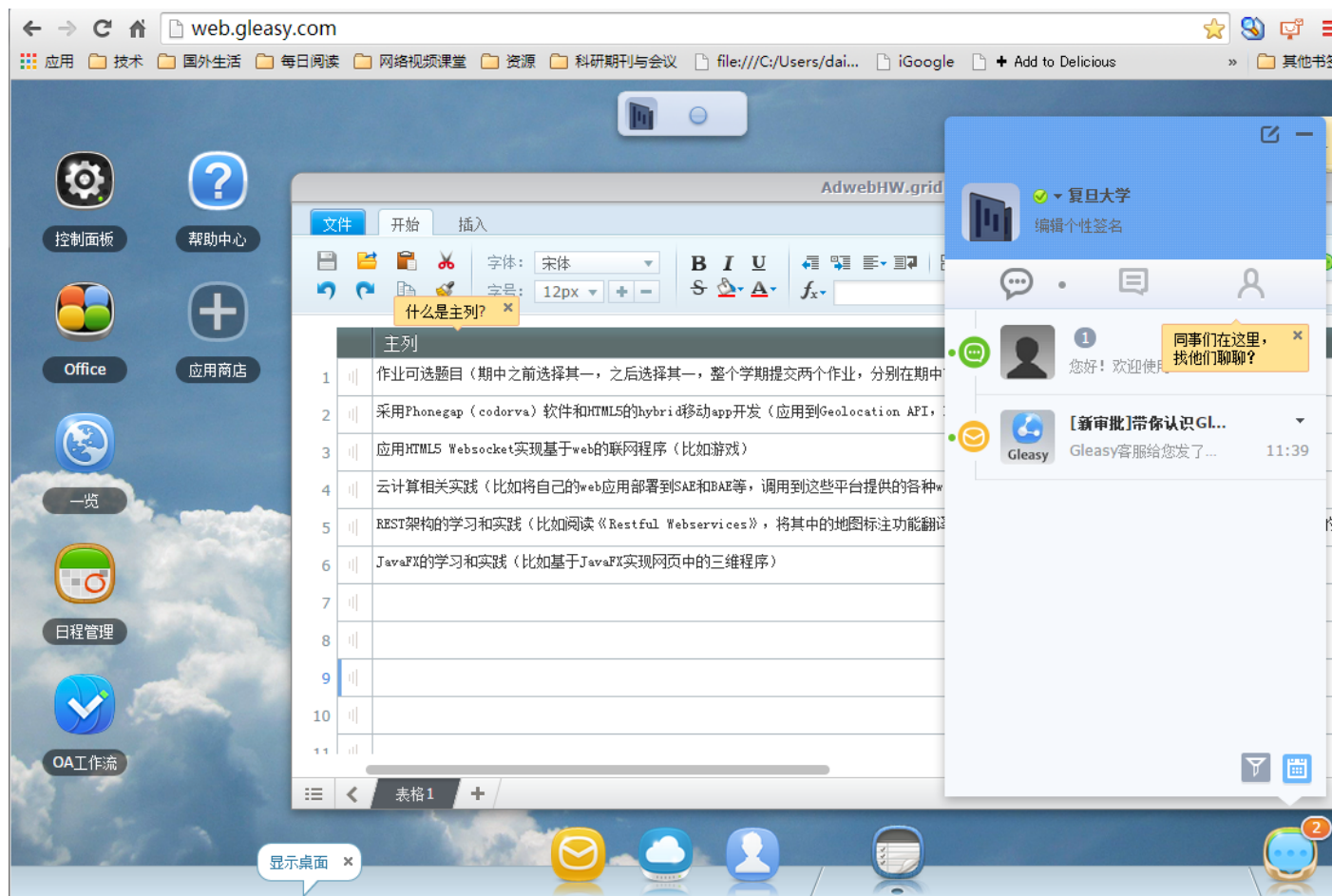
# Rich Internet Application技术

## ■ Rich Internet Applications

- 结合了桌面应用程序的反应快、交互性强的优点
- Web应用程序的传播范围广及容易传播，低成本部署的特性
- RIA简化并改进了Web应用程序的用户交互。这样，用户开发的应用程序可以提供更丰富、更具有交互性和响应性的用户体验



## RIA例子

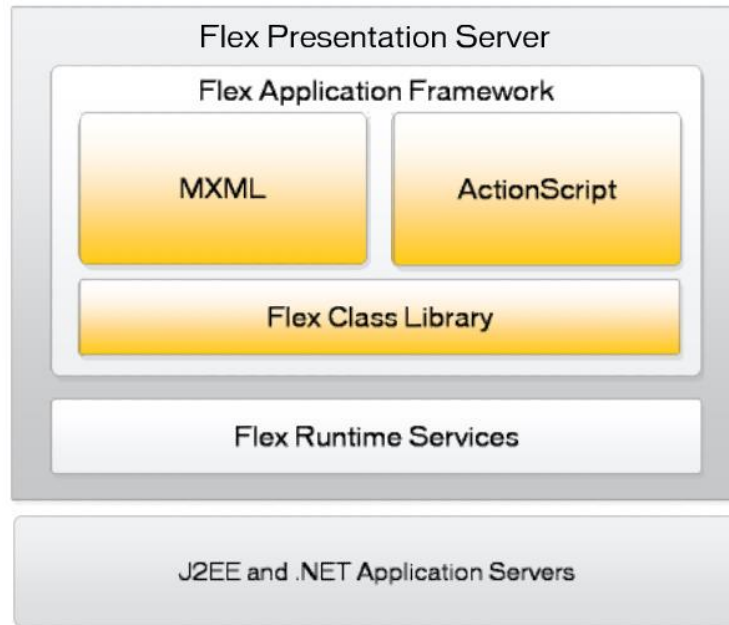


<http://web.gleasy.com/> ( admin@fudan2015, adweb2015)

# Rich Internet Application技术

## ■ FLEX

- 将MXML(the Macromedia Flex Markup Language)文件，编译成SWF文件，然后显示在浏览器中,并利用Web Service技术和服务器通信
- Flex是为满足希望开发 RIA的企业级程序员的需求而推出的表示服务器和应用程序框架，它可以运行于J2EE和.NET平台
- <http://www.adobe.com/cn/products/flex.html>

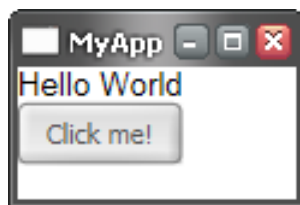


# Rich Internet Application技术

## ■ Microsoft SilverLight

- 基于NET Framework , 在浏览器内和桌面操作系统获得一致体验
- 基于XAML (eXtensible Application Markup Language) , 可以像HTML构建WEB页面一样构建应用程序的界面, XAML基于XML

```
<FlowPanel xmlns="http://schemas.microsoft.com/2003/xaml">  
  <Text>Hello World</Text>  
  <Button>Click me!</Button>  
</FlowPanel>
```



- 遵循WPF (Windows Presentation Foundation)



Microsoft®  
**Silverlight™**



# Rich Internet Application技术

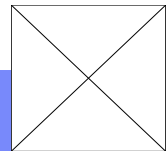
---

## ■ XUL

- XML用户接口语言，（XML User interface Language）
- 一个基于XML的用户界面语言，它来自于Mozilla的开放源码项目，XUL描述引擎都非常小（100K以下），
- XUL是一种用于交换数据的标准，它主要用于描述用户接口，具有独立于平台的特点，十分灵活，可以和XML等现在流行的语言相兼容
- XUL最大的优点在于它与Gecko引擎的集成（打开了通向大量Web标准的大门）

## ■ Bindows

- Bindow 是用Javascript和DHTML开发的Web窗体框架。Javascript用于客户端界面的显示和处理，XMLHTTP用于客户端与服务器的信息传输
- Bindows的一个主要的缺点是它采用一次全部载入的方式来实现脚本库
- 没有考虑到非IE的浏览器，限制了Bindows的流行



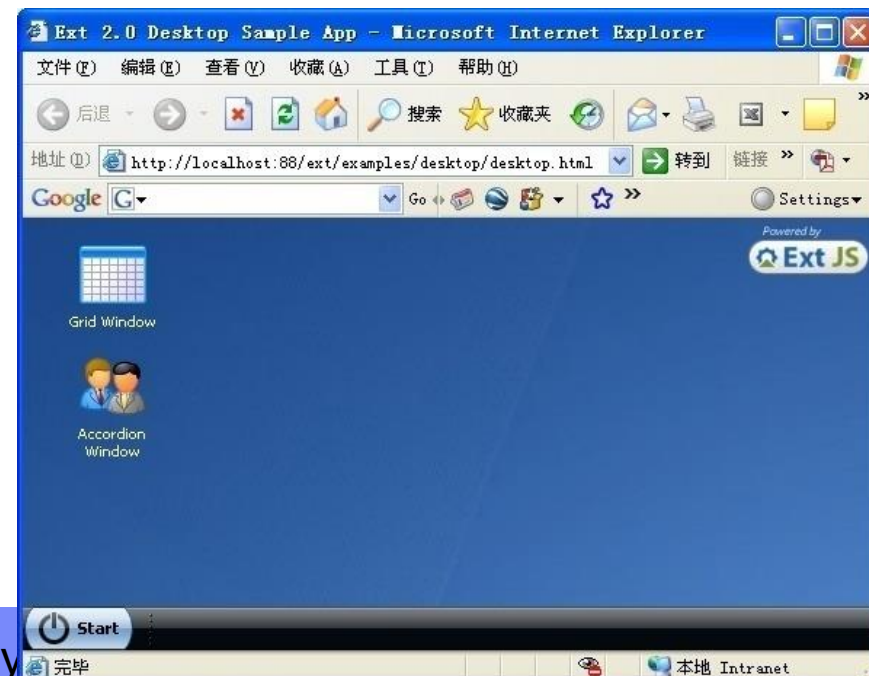
# Rich Internet Application技术

## ■ 开源的RIA开发环境Laszlo

- 编写名为LZX的描述语言（其中整合了XML和JavaScript），运行在J2EE应用服务器上的Laszlo平台会将其编译成FLASH文件并传输给客户端展示
- <http://www.openlaszlo.org>

## ■ Ext.

- JS库框架
- 内建卓越的Ajax支持
- <http://extjs.com/downloads>



Demo (in IIS)

<http://localhost:88/ext/examples/samples.html>

<http://localhost:88/ext/examples/portal/portal.html>

# Rich Internet Application技术

## ■ Java WebStart和JavaFX– sun

- 从版本 **1.3** 开始, **JRE** 就包含了 **Java Webstart**
- 简单但功能强大且灵活的将应用程序部署到任何平台的方法
- **Webstart** 允许应用程序的用户从他们的浏览器、电子邮件或桌面启动和管理应用程序 (只需在网页上点击一个超级链接就能运行一个Java桌面应用)
- 直接从服务端发布到客户端
- 检查用户是否在运行最新版本的应用程序

## ■ JavaFX

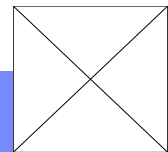
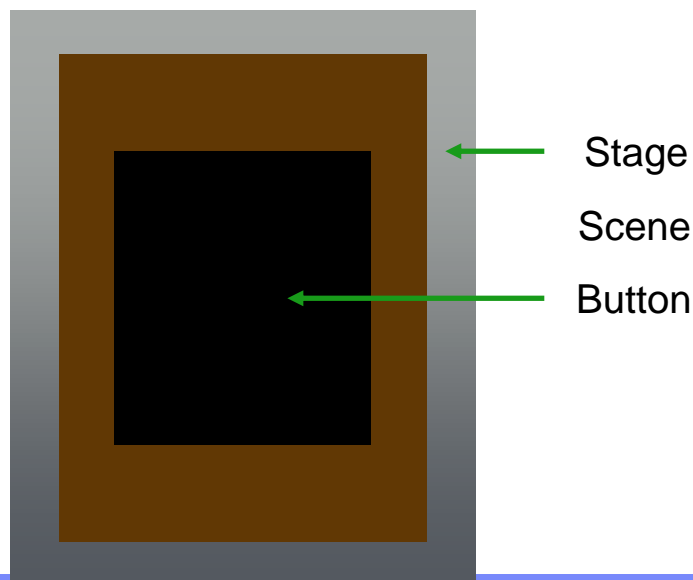
- **JAVA8.0**正式采用的一种**RIA**技术



# Rich Internet Application技术

## ■ **JavaFX**

- 支持桌面程序和浏览器应用的开发，可用于开发RIA
- 支持多点触控设备（如，手机，平板）
- 具有内建的2D，3D，动画支持，以及视频和音频的回放功能
- 基本程序结构
  - 舞台是一个支持场景的平台，节点如同在场景中演出的演员



# Rich Internet Application技术

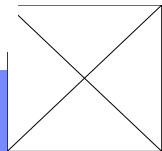
## ■ *JavaFX*

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.stage.Stage;

public class HelloWorld extends Application { // 每一个JavaFX类都继承自Application

    @Override
    public void start(Stage primaryStage) throws Exception { // 重写start方法
        Button btn = new Button("Hello World!");
        Scene scene = new Scene(btn, 200, 250); // 创建一个Scene来放置btn
        primaryStage.setTitle("Hello World");
        primaryStage.setScene(scene); // 将scene加入Stage
        primaryStage.show();
    }

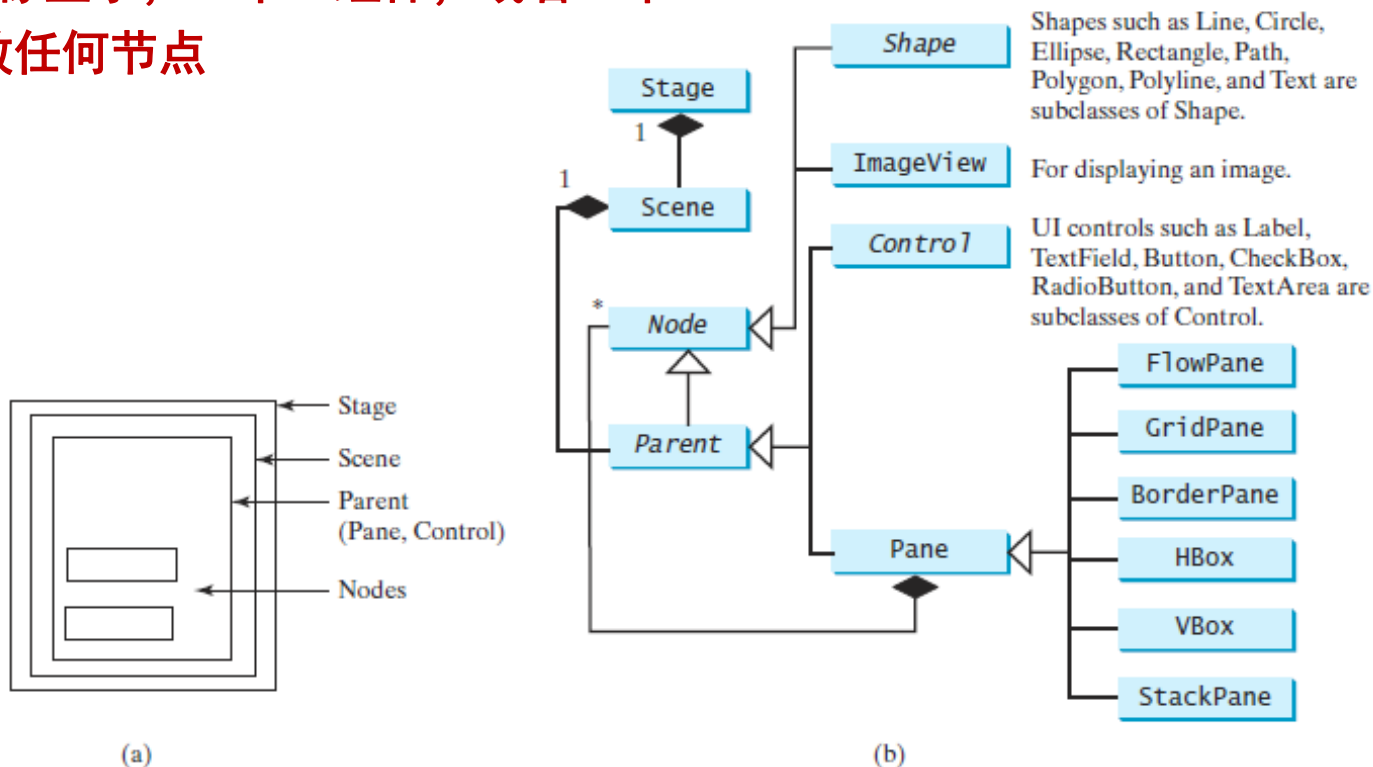
    public static void main(String[] args) { // 在部分支持JavaFX的IDE中需要
        launch();
    }
}
```



# Rich Internet Application技术

## ■ JavaFX

- 可以有多个Stage，Stage用来放Scene
- Scene可以包含Control或者Pane，节点是一个可视化的组件，比如一个形状，一个图像显示，一个UI组件，或者一个Pane
- Pane可以放任何节点



(a) Panes are used to hold nodes. (b) Nodes can be shapes, image views, UI controls, and panes.

# Rich Internet Application技术

## ■ JavaFX

- **属性绑定：** 将一个目标对象绑定到源对象中。源对象的修改将自动反映到目标对象中。

Target.bind(source)

```
public class ShowCircleCentered extends Application {
    @Override // Override the start method in the Application class
    public void start(Stage primaryStage) {
        // Create a pane to hold the circle
        Pane pane = new Pane();

        // Create a circle and set its properties
        Circle circle = new Circle();
        circle.centerXProperty().bind(pane.widthProperty().divide(2));
        circle.centerYProperty().bind(pane.heightProperty().divide(2));
        circle.setRadius(50);
        circle.setStroke(Color.BLACK);
        circle.setFill(Color.WHITE);
        pane.getChildren().add(circle); // Add circle to the pane

        // Create a scene and place it in the stage
        Scene scene = new Scene(pane, 200, 200);
        primaryStage.setTitle("ShowCircleCentered"); // Set the stage title
        primaryStage.setScene(scene); // Place the scene in the stage
        primaryStage.show(); // Display the stage
    }
}
```

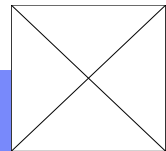
# Rich Internet Application技术

## ■ *JavaFX*

### – 属性的样式表CSS描述:

```
Button btOK = new Button("OK");  
btOK.setStyle("-fx-border-color: blue;");  
pane.getChildren().add(btOK);
```

```
pane.setRotate(45);  
pane.setStyle(  
    "-fx-border-color: red; -fx-background-color: lightgray;");
```



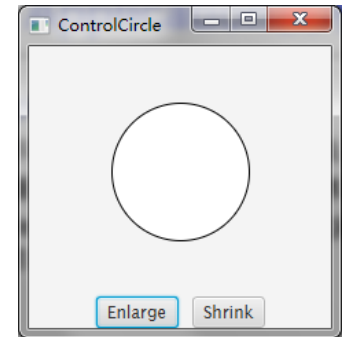


# Rich Internet Application技术

## ■ JavaFX

### – 事件驱动编程

- Handler类实现一个接口： `EventHandler<T extends Event>`
- 调用 `source.setOnAction(handler)` 登记 handler
- Handler类中的方法 handler 实现功能



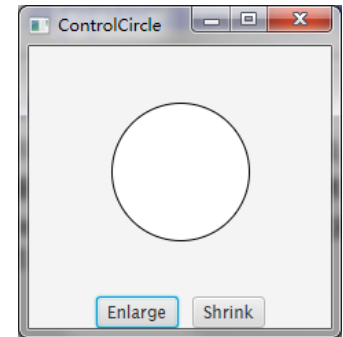
<i>User Action</i>	<i>Source Object</i>	<i>Event Type Fired</i>	<i>Event Registration Method</i>
Click a button	Button	ActionEvent	<code>setOnAction(EventHandler&lt;ActionEvent&gt;)</code>
Press Enter in a text field	TextField	ActionEvent	<code>setOnAction(EventHandler&lt;ActionEvent&gt;)</code>
Check or uncheck	RadioButton	ActionEvent	<code>setOnAction(EventHandler&lt;ActionEvent&gt;)</code>
Check or uncheck	CheckBox	ActionEvent	<code>setOnAction(EventHandler&lt;ActionEvent&gt;)</code>
Select a new item	ComboBox	ActionEvent	<code>setOnAction(EventHandler&lt;ActionEvent&gt;)</code>
Mouse pressed	Node, Scene	MouseEvent	<code>setOnMousePressed(EventHandler&lt;MouseEvent&gt;)</code>
Mouse released			<code>setOnMouseReleased(EventHandler&lt;MouseEvent&gt;)</code>
Mouse clicked			<code>setOnMouseClicked(EventHandler&lt;MouseEvent&gt;)</code>
Mouse entered			<code>setOnMouseEntered(EventHandler&lt;MouseEvent&gt;)</code>
Mouse exited			<code>setOnMouseExited(EventHandler&lt;MouseEvent&gt;)</code>
Mouse moved			<code>setOnMouseMoved(EventHandler&lt;MouseEvent&gt;)</code>
Mouse dragged			<code>setOnMouseDragged(EventHandler&lt;MouseEvent&gt;)</code>
Key pressed	Node, Scene	KeyEvent	<code>setOnKeyPressed(EventHandler&lt;KeyEvent&gt;)</code>
Key released			<code>setOnKeyReleased(EventHandler&lt;KeyEvent&gt;)</code>
Key typed			<code>setOnKeyTyped(EventHandler&lt;KeyEvent&gt;)</code>

# Rich Internet Application技术

## ■ JavaFX

### – 事件驱动编程

- Handler类实现一个接口： `EventHandler<T extends Event>`
- 调用 `source.setOnAction(handler)` 登记 handler
- Handler类中的方法 handler 实现功能



<i>User Action</i>	<i>Source Object</i>	<i>Event Type Fired</i>	<i>Event Registration Method</i>
Click a button	Button	ActionEvent	<code>setOnAction(EventHandler&lt;ActionEvent&gt;)</code>
Press Enter in a text field	TextField	ActionEvent	<code>setOnAction(EventHandler&lt;ActionEvent&gt;)</code>
Check or uncheck	RadioButton	ActionEvent	<code>setOnAction(EventHandler&lt;ActionEvent&gt;)</code>
Check or uncheck	CheckBox	ActionEvent	<code>setOnAction(EventHandler&lt;ActionEvent&gt;)</code>
Select a new item	ComboBox	ActionEvent	<code>setOnAction(EventHandler&lt;ActionEvent&gt;)</code>
Mouse pressed	Node, Scene	MouseEvent	<code>setOnMousePressed(EventHandler&lt;MouseEvent&gt;)</code>
Mouse released			<code>setOnMouseReleased(EventHandler&lt;MouseEvent&gt;)</code>
Mouse clicked			<code>setOnMouseClicked(EventHandler&lt;MouseEvent&gt;)</code>
Mouse entered			<code>setOnMouseEntered(EventHandler&lt;MouseEvent&gt;)</code>
Mouse exited			<code>setOnMouseExited(EventHandler&lt;MouseEvent&gt;)</code>
Mouse moved			<code>setOnMouseMoved(EventHandler&lt;MouseEvent&gt;)</code>
Mouse dragged			<code>setOnMouseDragged(EventHandler&lt;MouseEvent&gt;)</code>
Key pressed	Node, Scene	KeyEvent	<code>setOnKeyPressed(EventHandler&lt;KeyEvent&gt;)</code>
Key released			<code>setOnKeyReleased(EventHandler&lt;KeyEvent&gt;)</code>
Key typed			<code>setOnKeyTyped(EventHandler&lt;KeyEvent&gt;)</code>

# Rich Internet Application技术

## ■ JavaFX

### – Lambda表达式

- 针对Single Abstract Method(SAM), Lambda表达式可简化编程

```
btn.setOnAction(new EventHandler<ActionEvent>() {  
    public void handle(ActionEvent e) {  
        System.out.println("Clicked");  
        btn.setStyle("-fx-border-color: " + color[pointer] + ";");  
        pointer++;  
        if (pointer >= color.length)  
            pointer = 0;  
    }  
}); // 用匿名类代码仍然显得冗余
```

```
btn.setOnAction(e -> {  
    System.out.println("Clicked");  
    btn.setStyle("-fx-border-color: " + color[pointer] + ";");  
    pointer++;  
    if (pointer >= color.length)  
        pointer = 0;  
}); // 通过使用Lambda表达式代码显得更加简洁
```

代码简化了

# Rich Internet Application技术

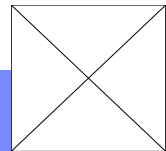
## ■ **JavaFX**

### - 动画制作

- PathTransition
- FadeTransition
- Timeline

```
// Create an animation for alternating text
Timeline animation = new Timeline(
    new KeyFrame(Duration.millis(500), eventHandler));
animation.setCycleCount(Timeline.INDEFINITE);
animation.play(); // Start animation

// Pause and resume animation
text.setOnMouseClicked(e -> {
    if (animation.getStatus() == Animation.Status.PAUSED) {
        animation.play();
    }
    else {
        animation.pause();
    }
});
```



# Rich Internet Application技术

## ■ Native Client of Google

Native Client is an open-source technology for running native code in web applications, with the goal of maintaining the browser neutrality, OS portability and safety that people expect from web apps. We believe that Native Client technology will help web developers to create richer and more dynamic browser-based applications.



*Quake in the browser*