

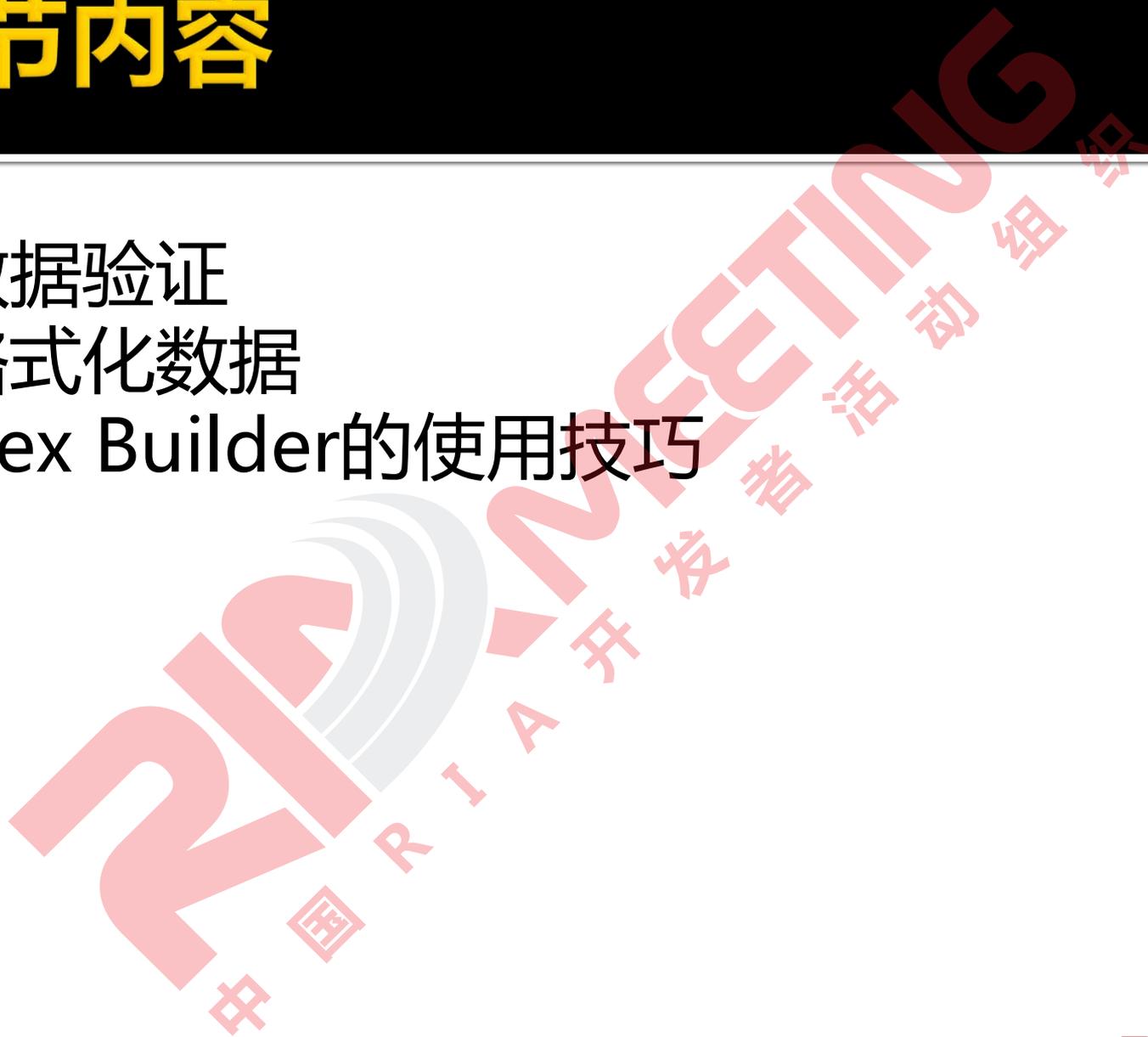
RIA 大讲堂

Flex 数据验证

RIA MEETING
中国 RIA 开发者活动组织

本节内容

- 数据验证
- 格式化数据
- Flex Builder的使用技巧



第一部分：数据验证



两个问题



什么是数据验证

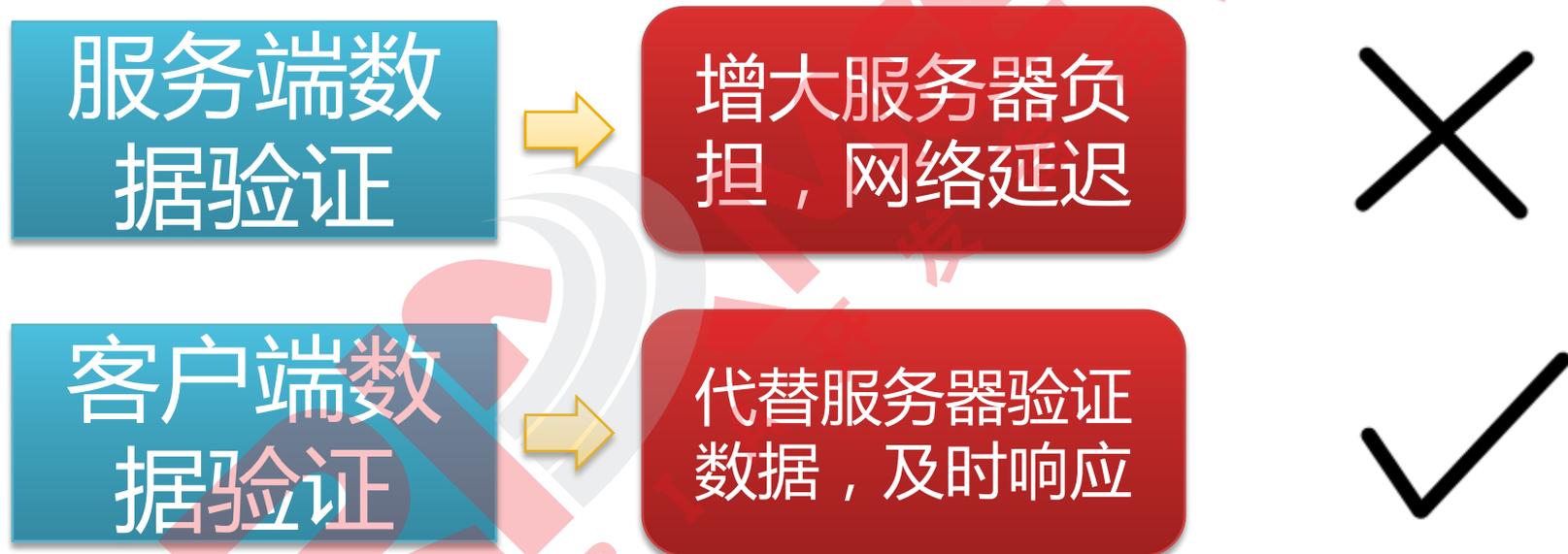
为什么使用数据验证

什么是数据验证

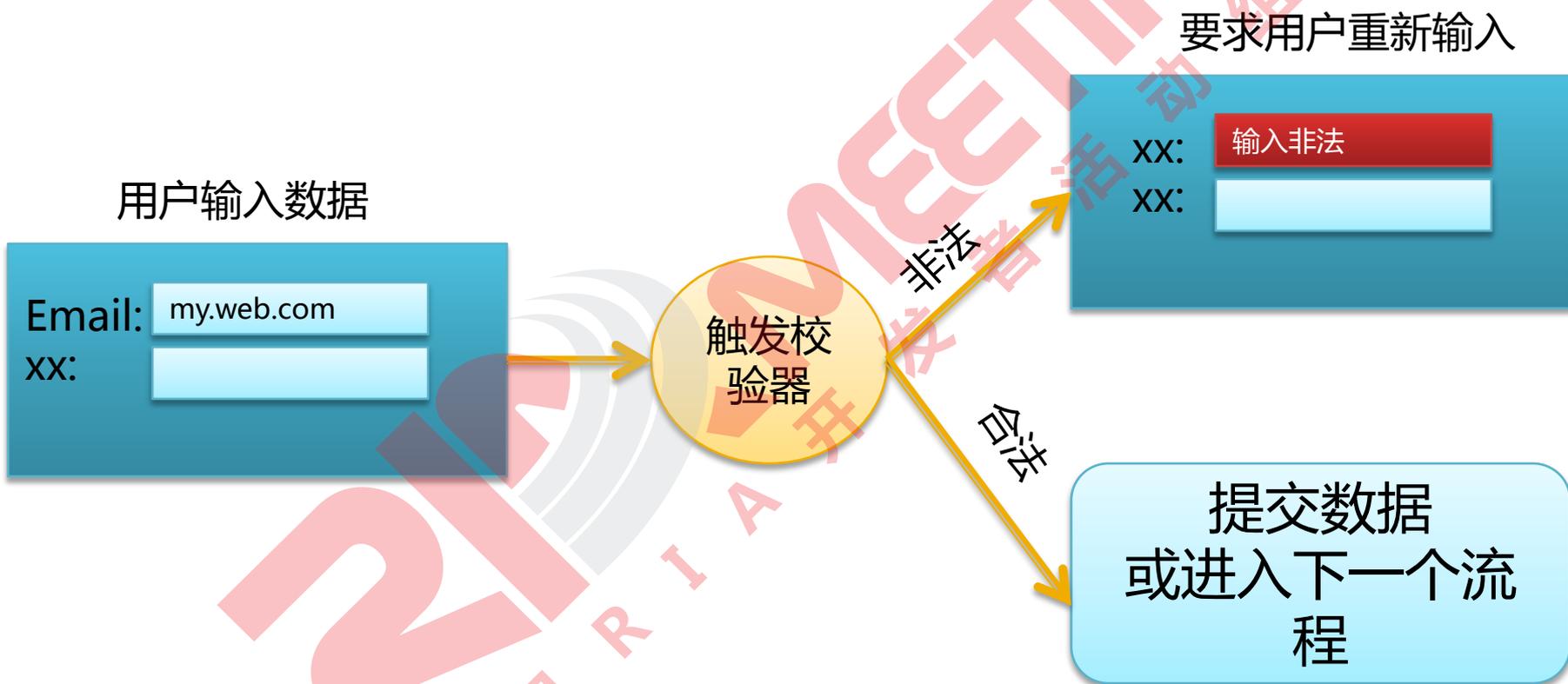
- 数据验证就是对用户输入的数据做格式检查
 - 类似于HTML，数据验证一般用于表单提交之前的格式检查
- 例如：
 - Email格式检测：
 - 合法格式：xxx@web.com
 - 非法格式：xxx.web.com

为什么使用数据验证

- 数据验证是必须的，两种方式：



校验的过程



Flex中的数据验证

校验器 Validator

Flex中的数据验证是有校验器来完成的

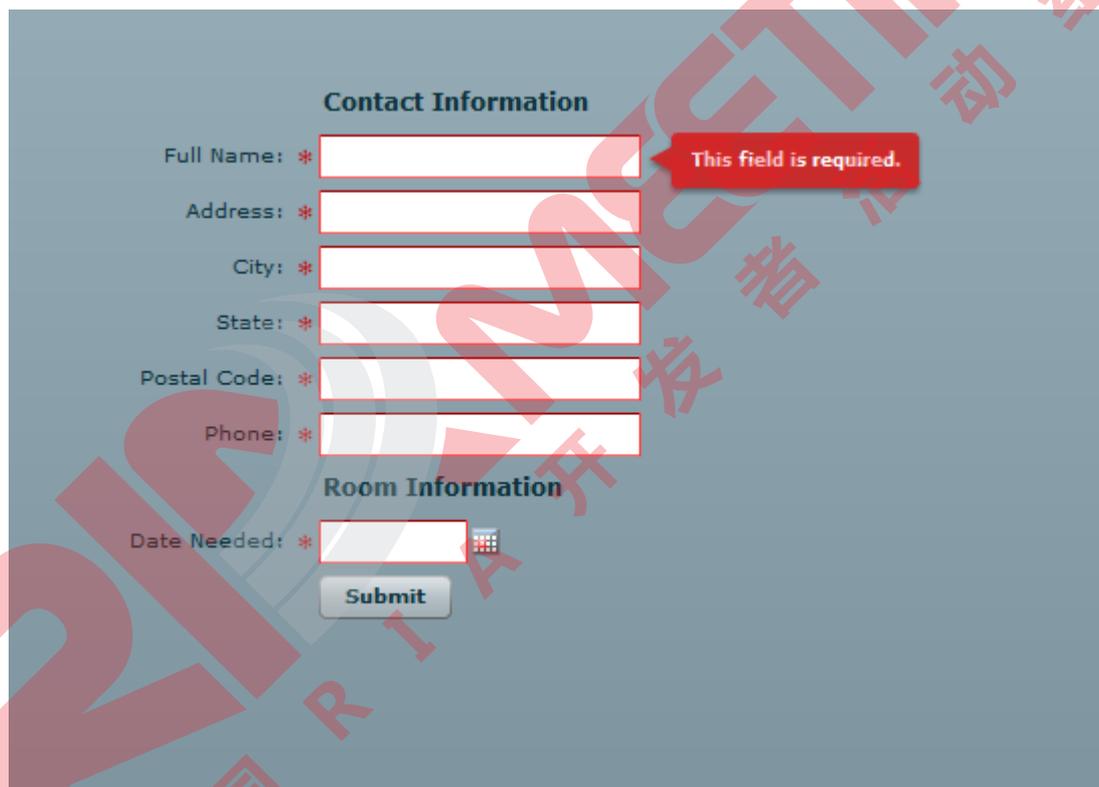
校验器类型A

- 验证信用卡号码
<mx:CreditCardValidator>
- 验证货币 <mx:Currency>
- 验证日期 <mx:DateValidator>
- 验证Email <mx:EmailValidator>
- 验证数字 <mx:NumberValidator>

校验器类型B

- 验证电话号码
<mx:PhoneNumberValidator>
- 验证正则表达式 <mx:RegExpValidator>
- 验证加密数字
<mx:SocialSecurityValidator>
- 验证字符串 <mx:StringValidator>
- 验证邮政编码 <mx:ZipCodeValidator>

校验器实际工作演示



The screenshot shows a web form with two sections: "Contact Information" and "Room Information". The "Contact Information" section contains six input fields: "Full Name", "Address", "City", "State", "Postal Code", and "Phone". Each field has a red asterisk to its left. A red tooltip with the text "This field is required." is positioned over the "Full Name" field. The "Room Information" section contains one input field: "Date Needed", also with a red asterisk to its left. A "Submit" button is located below the "Date Needed" field. A large, semi-transparent watermark "RIA MEETING 中国 RIA 开发者活动组织" is overlaid on the form.

Contact Information

Full Name: * This field is required.

Address: *

City: *

State: *

Postal Code: *

Phone: *

Room Information

Date Needed: *

必填项

```
<mx:FormItem label="City:"  
  required="true" >  
<mx:TextInput id="city" />  
</mx:FormItem>
```

注意这只是在显示的时候增加一个*号，而不会验证数据的格式

校验器的使用方式

```
<mx:EmailValidator id="emailCheck"  
    source="{email}"  
    property="text" />  
<mx:TextInput id="email" />
```

source属性需要对一个正在校验的对象实例名绑定

校验器的触发方式

```
<mx:StringValidator  
id="fullnameV"  
source="{fullname}"  
property="text"  
trigger="{button}"  
triggerEvent="click" />
```

默认：失去焦点则马上校验

使用trigger：当按钮事件触发再予以校验

使用代码创建校验器

```
private var phoneCheck:PhoneNumberValidator;  
private function createPhoneValidator():void  
{  
    phoneCheck = new PhoneNumberValidator();  
    phoneCheck.source = phone;  
    phoneCheck.property = "text";  
}
```

完整数据验证实例讲解

The image shows a web form with two sections: '联系信息' (Contact Information) and '住所信息' (Address Information). The '联系信息' section contains input fields for '姓名' (Name) with value 'name', '地址' (Address) with value 'di', '城市' (City) with value 'di', '州' (State) with value 'di', '邮政编码' (Zip Code) with value '90349', and '电话' (Phone) with value '290384343925'. The '住所信息' section contains a '联系时间' (Contact Time) field with value '01/25/2009' and a 'Submit' button. A modal dialog box is displayed in the foreground with the message '数据合法, 可以提交数据' (Data is valid, can submit data) and an 'OK' button.

联系信息

姓名: name

地址: di

城市: di

州: di

邮政编码: 90349

电话: 290384343925

住所信息

联系时间: 01/25/2009

Submit

数据合法, 可以提交数据

OK

第二部分：格式化数据



什么是格式化



格式化用于将数据转换为成特定格式的字符串

比如

日期：

Fri Jun 5 17:19:43 GMT+0800 2009



日期：

2009-6-5

了解格式化

- 格式化类都是mx.formatter.Formatter的子类
 - 包括：
 - CurrencyFormatter 货币格式化
 - DateFormatter 日期格式化
 - NumberFormatter 数字格式化
 - PhoneFormatter 电话号码格式化
 - ZipCodeFormatter 邮政编码格式化

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/258072042032006107>