

第十七章

Web测试

# 上一章回顾

* 安装测试验证点
* 运行测试验证点
* 卸载测试验证点

# 课堂提问

* 安装测试要点

# 本章学习目标

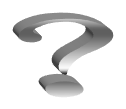
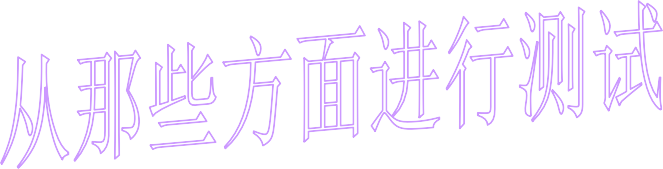
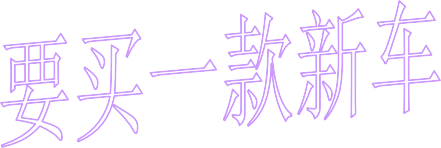
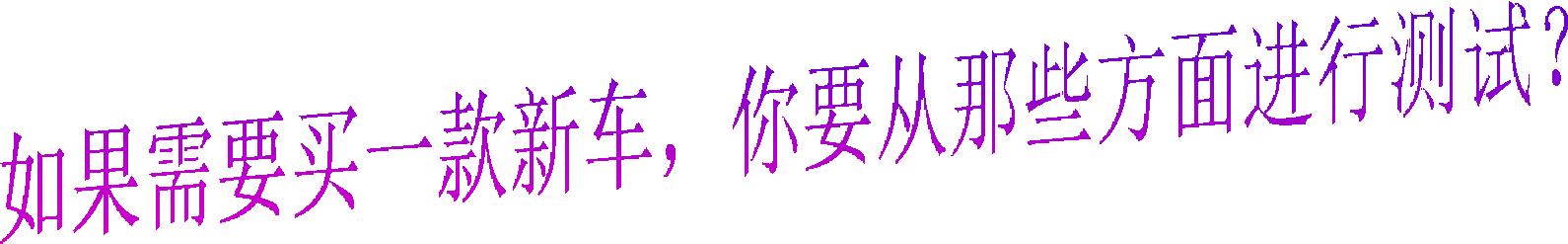
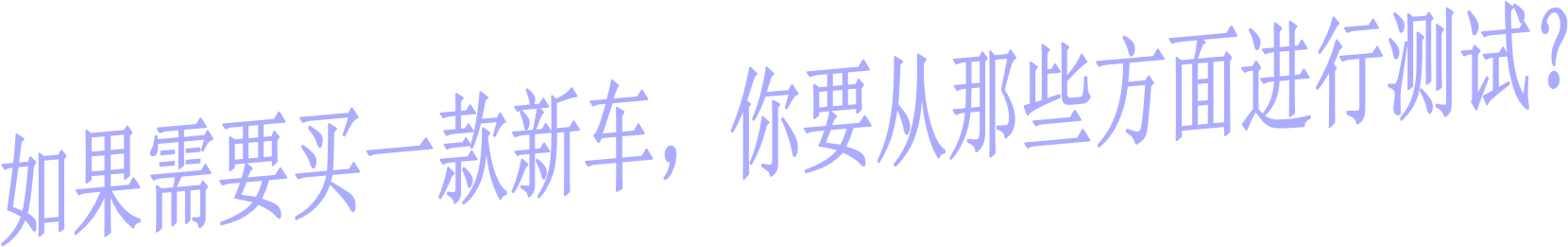
* 能够从事Web测试

# 本章学习方法

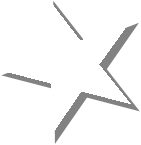
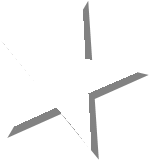
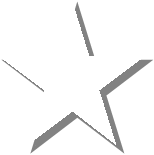
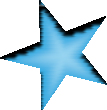
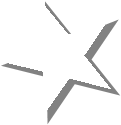
* 准备Web项目进行测试

# 课前思考









**美观测试**：外观造型、颜色、室内沙发、靠

1

背等配件位置和颜色搭配… …

**功能测试：**点火、启动、加速、行驶、转向 、

2

刹车、停车 … …

**性能测试:**启动加速到100千米/小时需要多少秒

3

、100公里耗油、最大时速… …

**兼容性测试：**配置是否为通用型、是否允许在高

4

速上行驶轮、胎是普通轮胎还是特制轮胎、… …

**安全测试：紧急刹车、抗撞击能力、安全气囊**

5

… …

# 本章主题

* Web功能测试
* Web性能测试
* Web界面测试
* Web兼容性测试
* Web安全性测试

# 功能测试

* 链接测试
* 表单测试
* Cookies测试
* 设计语言测试
* 数据库测试

# 链接测试

* 链接与页面对应
  + 测试所有链接是否按指示的那样确实链接到了该链接的页面
* 无空链接
  + 测试所链接的页面是否存在
* 没有孤立的页面
  + 孤立页面是指没有链接指向该页面，只有知道正确的URL地址才能访问

# 链接测试方法

* 链接测试执行的时间一般在集成测试阶段， 即在整个Web应用系统的所有页面开发完成后进行链接测试。
* 链接测试可以运行手工测试；也可以通过自动化测试工具测试。
* 手工测试方法：点击任何一个可能有链接的地方，看页面和链接是否对应，看是否有空链接，看是否存在孤立的页面。

# 链接测试工具

* Xenu Link Sleuth
* HTML Link Validator
* ACT
* Rational sitecheck
* Rational linkbot
* 演示Xenu Link Sleuth

# 链接测试案例

* + 案例演示：点击链接无反应
  + 错误现象：
  + 原因：

超链接地址有误

* + 解决办法

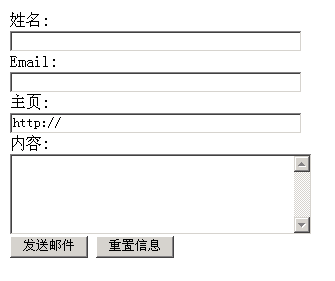
更正超链接地址

# 表单测试

* 当用户给Web应用系统管理员提交信息时， 就需要使用表单操作
* 必须测试提交操作的完整性，以校验提交给服务器的信息的正确性。

# 表单测试事例

* + 示例表单将检查如下功能：
    - 姓名是否为空
    - Email地址是否为空，是否包含“@”和“.”
    - 主页网址是否为空，是否包含“http://” “.”
    - 内容是否为空



# 表单测试事例

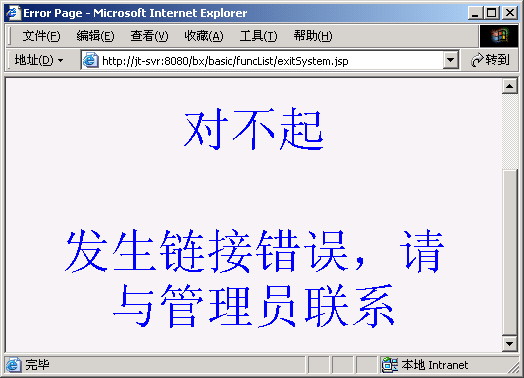
* + 案例演示：表单提交信息不完整
  + 错误现象及重现步骤：

# 表单测试事例



链接测试案例

* + 案例演示：链接页面不存在
  + 错误现象及重现步骤：
  + 原因：链接页面不存在



# Cookies测试

* Cookies通常用来存储用户信息和用户在某应用系统的操作，当一个用户使用Cookies 访问了某一个应用系统时，Web服务器将发送关于用户的信息，把该信息以Cookies的形式存储在客户端计算机上，这可用来创建动态和自定义页面或者存储登陆等信息
* 测试的内容可包括Cookies是否起作用，是否按预定的时间进行保存，刷新对Cookies 有什么影响

# 设计语言测试

* Java、JavaScript、 ActiveX、VBScript或Perl

等要进行验证

# 数据库测试

* 数据一致性错误
  + 数据一致性错误主要是由于用户提交的表单信息不正确而造成的
* 输出错误
  + 输出错误主要是由于网络速度或程序设计问题等引起的，针对这两种情况，可分别进行测试

# 功能测试小结

|  |  |
| --- | --- |
| **测试内容** | **功能测试** |
| 链接测试 | 测试所有链接是否按指示的那样确实链接到了该链接的页面 |
| 测试所链接的页面是否存在 |
| 保证Web应用系统上没有孤立的页面 |
| 表单测试 | 测试提交操作的完整性，以校验提交给服务器的信息的正确性 |
| Cookies测试 | Cookies是否起作用 |
| 是否按预定的时间进行保存(Cookies保存的时间长度) |
| 刷新对Cookies有什么影响(登录一个新的用户) |
| 设计语言测试 | HTML的多个版本的验证(IE的多个版本,火狐等) |
| 不同的脚本语言，例如Java、JavaScript、 ActiveX、VBScript或  Perl |
| 数据库测试 | 数据一致性错误,主要是由于用户提交的表单信息不正确而造成的 |
| 输出错误,主要是由于网络速度或程序设计问题等引起的 |

性能测试

* 连接速度测试
* 负载测试
* 压力测试

# 连接速度测试

* 用户连接到Web应用系统的速度根据上网方式的变化而变化，他们或许是电话拨号，或是宽带上网
* 如果Web系统响应时间太长（例如超过5秒钟），用户就会因没有耐心等待而离开
* 有些页面有超时的限制，如果响应速度太慢， 用户可能还没来得及浏览内容，就需要重新登陆了
* 连接速度太慢，还可能引起数据丢失，使用户得不到真实的页面

# 负载测试

* 根据场景设计测试用例
  + 100用户登录邮箱
  + 200用户查看新闻
  + 300用户查看科技

# 压力测试

* 并发测试
  + 500人同时登录邮箱

# 性能测试小结

|  |  |
| --- | --- |
| **测试内容** | **性能测试** |
| 连接速度测试 | 登入链接时间,页面刷新时间等 |
| 负载测试 | 在系统"满负荷"的情形下,测试系统是承受能力(能运行多长时间) |
| 压力测试 | 获取系统正确运行的极限. |

界面测试

* 导航测试
* 图形测试
* 内容测试
* 整体界面测试

# 导航测试

* 方便快捷的访问到用户需要的信息
* 在任何页面上都可以清楚地知道页面所处

Web应用系统中的位臵

* 页面逻辑结构清晰，层次分明
* 容易返回上一状态或主页面

# 图形测试

* 确保图形有明确的用途
* 图形无法显示
* 图形下载速度缓慢
* 放臵重要信息的图片丢失
* 所有页面字体的风格一致
* 背景颜色应该与字体颜色和前景颜色相搭配
* 图片的大小和质量影响性能

# 内容测试

* 检验Web应用系统提供信息的正确性、准确性和相关性

# 整体界面测试

* 设计风格一致性

# 界面测试小结

|  |  |
| --- | --- |
| **测试内容** | **界面测试** |
| 导航测试 | 是否易于导航 |
| 导航是否直观 |
| Web系统的主要部分是否可通过主页存取 |
| Web系统是否需要站点地图、搜索引擎或其他的导航帮助 |
| 图形测试 | 要确保图形有明确的用途，图片或动画不要胡乱地堆在一起，以免  浪费传输时间 |
| 验证所有页面字体的风格是否一致 |
| 背景颜色应该与字体颜色和前景颜色相搭配 |
| 图片的大小和质量也是一个很重要的因素，一般采用JPG或GIF压缩 |
| 内容测试 | 内容测试用来检验Web应用系统提供信息的正确性、准确性和相关性 |
| 整体界面测试 | 当用户浏览Web应用系统时是否感到舒适，是否凭直觉就知道要找的  信息在什么地方？ |
| 整个Web应用系统的设计风格是否一致？ |

兼容性测试

* 平台（操作系统）测试
* 浏览器测试
* 分辨率测试

# 安全性测试

* 先注册，后登陆
* 登录密码控制
* 用户登陆后在一定时间内（例如15分钟） 没有点击任何页面，是否需要重新登陆才能正常使用
* 日志
* 能够在服务器端放臵和编辑脚本

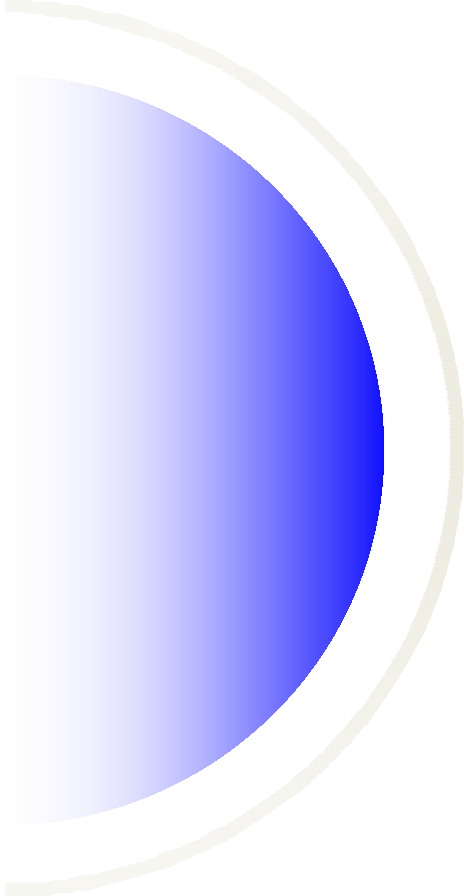
**兼容性、安全性测试小结**

|  |  |
| --- | --- |
| **测试内容** | **客户端兼容性测试** |
| 平台测试 | 在Web系统发布之前，需要在各种操作系统下对Web系统进行兼  容性测试 |
| 浏览器测试 | 测试浏览器兼容性的一个方法是创建一个兼容性矩阵。在这个矩阵中，测试不同厂商、不同版本的浏览器对某些构件和设臵的适应性 |
| 分辨率测试 | 在不同分辨率下，界面控件是否能正常显示 |
| **测试内容** | **安全性测试** |
| 安全性测试 | 用户登录 |
| Web应用系统是否有超时的限制 |
| 为了保证Web应用系统的安全性，日志文件是至关重要的 |
| 当使用了安全套接字时，还要测试加密是否正确，检查信息的  完整性 |
| 服务器端的脚本常常构成安全漏洞,要测试没有经过授权，就  不能在服务器端放臵和编辑脚本的问题 |

# 总结

* Web功能测试
* Web性能测试
* Web界面测试
* Web兼容性测试
* Web安全性测试
* 请预习第十八章

# 扩展一



**功能测试**

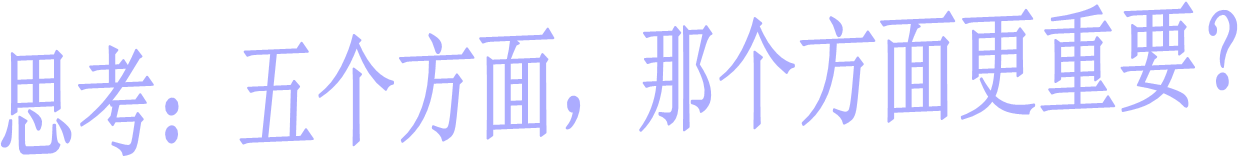
**性能测试**

Internet

**界面测试**

**兼容性测试**

**安全测试**



安全性测试术语介绍

1、什么是“易受攻击性”？

这是网络应用程序的一个软肋。造成这个“软肋”的原因，可能是程序中的

bug，一种注入（SQL/脚本代码）或者已存在的病毒。

2、什么是“URL处理”？

一些网络应用程序通过URL在客户端（浏览器）和服务器端之间进行额外信

息的传递。有时在URL中改变信息可能会导致服务器不可预期的结果。

3、什么是“SQL注入”？

是指通过网络应用的用户接口插入SQL指令到服务器所执行的查询中去的过

程。

4、什么是“XSS（跨站式脚本攻击）”？

当一个用户通过Web应用接口插入HTML代码，而这种嵌入其中的代码对于其

他用户是可见的，被称做XSS。

5、什么是“欺骗”？

完全仿造网站或电子邮件的赝品被称为欺骗

# 2011十大安全漏洞

对比2004,2007年十大安全漏洞,几乎每年的漏洞情况都是变动，随着Web的发展,

都会出现新的安全漏洞，或者原来的安全漏洞优先级靠前.

A1 注入式漏洞

A2 跨站脚本

A3 无效的验证和会话管理 A4 对不安全对象的直接引用A5 伪造的跨站请求

A6 安全配置错误

A7 加密存储方面的不安全因素

A8 不限制访问者的URL

A9 传输层面的保护力度不足

A10 未经验证的重新指向及转发

# 扩展二：怎样开展Web测试

* 获取测试需求，功能列表或测试矩阵
* 执行功能测试，同时完成界面测试、易用性测试，并将安全测试思想加入到功能测试中，进行简单的安全测试
* 兼容性测试
* 如有要求，深入执行安全测试
* 性能测试

# 上机课作业

* 针对当前的Web系统编写Web方面的测试用例，并进行测试

# 家庭作业

* 将本PPT各小结中的测试项和测试内容手工输入到EXCEL中，做成一个通用测试用例
* 查找B/S与C/S结构的区别