



# 跨浏览器JavaScript开发

罗宇翔

Google软件工程师

# 内容

- JavaScript解释器和布局引擎
- 常见兼容性问题和应对策略
- 几点建议

# JavaScript解释器和布局引擎

新一代  
JavaScript  
解释器

Chakra

TraceMonkey

V8

SquirrelFish

Carakan

JavaScript  
解释器

JScript

SpiderMonkey

JavaScriptCore

Futhark

布局引擎

Trident

Gecko

WebKit

Presto

浏览器

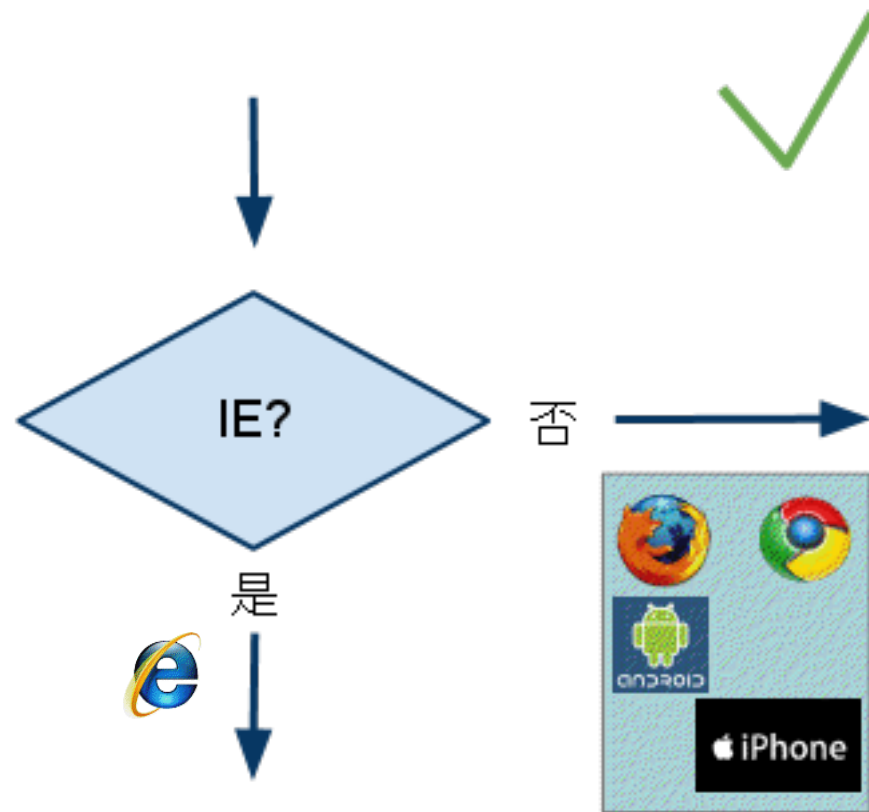
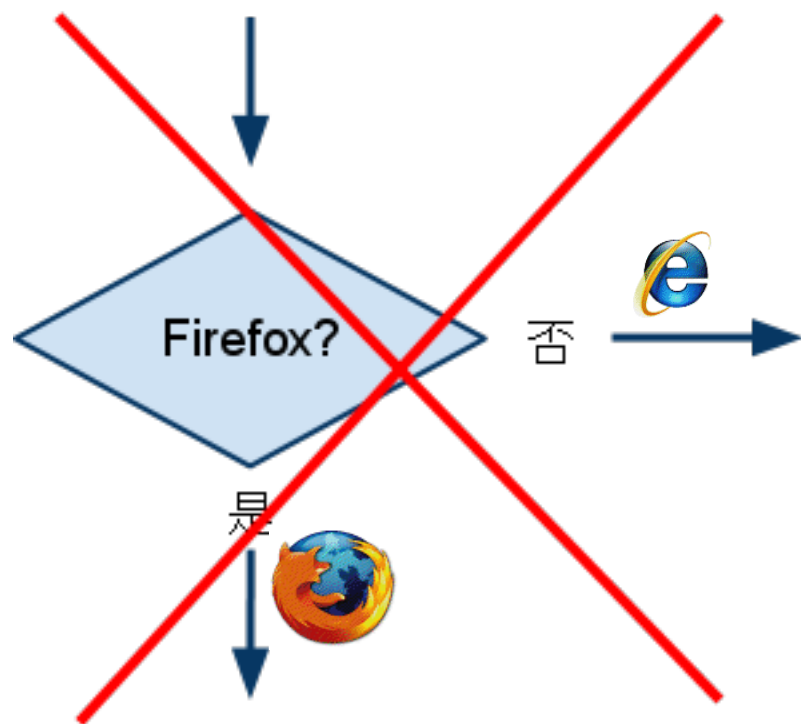


用户在使用什么浏览器？

---



# 浏览器判断



# User Agent 检查

**Mozilla/5.0** (Macintosh; U; Intel Mac OS X 10\_5\_8; en-US)


Apple**WebKit/534.1** (**KHTML, like Gecko**) **Chrome/6.0.437.3** **Safari/534.1**

- 正确分析navigator.userAgent不容易
- 模拟其它浏览器的userAgent，测试判断逻辑是否正确

## 更好的方法——判断对象是否存在



```
if (navigator.userAgent.indexOf('Gecko') != -1)
  ...
else (navigator.userAgent.indexOf('MSIE') != -1)
  ...
else (navigator.userAgent.indexOf('AppleWebKit') != -1)
  ...
else
  ...
```



```
if (document.querySelectorAll) {
  // Not all browser versions implement this function
  var a = document.querySelectorAll(':hover');
}
```

# 浏览器名称 VS 布局引擎

~~if (navigator.userAgent.indexOf('Chrome') != -1)~~



```
var isWebKit = navigator.userAgent.indexOf('AppleWebKit') > -1;
```

```
var WebKitVersion =  
    parseFloat(navigator.userAgent.split('AppleWebKit')[1]) || undefined;  
if (isWebKit && WebKitVersion > 500) {  
    // user expected WebKit feature here  
}
```



# 浏览器检测——总结

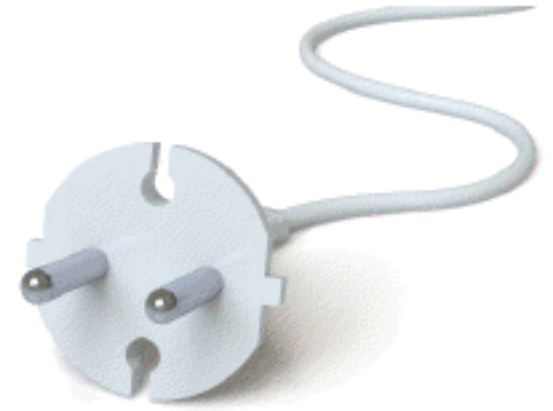
- 特殊对待IE而不是其他浏览器
- User Agent分析容易出错
- 尽量检测对象是否存在，而不是User Agent
- 针对布局引擎（如WebKit），不是浏览器（如Safari）

# 视频和插件

---



# 浏览器插件



- 插件：增强浏览器功能的外部程序
- 如果使用某个插件，确保它被主流浏览器支持
- ActiveX插件只被IE支持，限制于Windows平台
- 是否真的需要插件？

# 视频



- Flash是当前最流行的网络视频格式
- 如何嵌入Flash视频？

```
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"  
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,40,0"  
width="550" height="400" id="myMovie">  
  <param name="movie" value="myFlashMovie.swf">  
  <embed src="/support/flash/ts/documents/myFlashMovie.swf"  
    width="550" height="400"  
    name="myMovieName" type="application/x-shockwave-flash"  
    pluginspage="http://www.macromedia.com/go/getflashplayer">  
</embed>  
</object>
```

- 缺点：
  - 缺乏版本检测
  - embed非标准标签

# SWFObject——更好的选择

- 使用SWFObject嵌入Flash
  - <http://code.google.com/p/swfobject> (MIT协议)

`<script type="text/javascript" src="swfobject.js"></script>`

`<div id="flashcontent">动画</div>`

`<script type="text/javascript">`

`swfobject.embedSWF("movie.swf", "flashcontent", "200", "100", "9.0.0",  
expressInstall.swf);`

`</script>`

- 优点：
  - 版本检测
  - 封装, 写起来更简单
  - 避免使用非标准的embed标签

# HTML5 Video

- 优点
  - 跨浏览器，跨操作系统平台
  - 开放标准，不依赖于第三方插件
- 从现在开始试用HTML5 <Video> tag
- 使用Flash作为fallback

# 异步加载脚本

---



# 什么是异步加载的脚本？

- 被异步加载的脚本
  - 改变已有script元素的'src'属性
  - 用createElement创建script元素，使用'src'属性
- 优点
  - 需要时加载
  - 加快页面内容的呈现



# 加载时机

```
<script type='text/javascript'>  
var js = document.createElement('script');  
js.src = 'http://code.jquery.com/jquery-1.4.2.js';  
document.getElementsByTagName('head')[0].appendChild(js);  
</script>
```

```
<script type = 'text/javascript'>  
alert($)  
</script>
```

Firefox, Opera	显示\$函数
IE	未定义
Chrome, Safari	未定义

## 加载时机（2）

```
<script type='text/javascript'>  
var js = document.createElement('script');  
document.getElementsByTagName('head')[0].appendChild(js);  
js.src = 'http://code.jquery.com/jquery-1.4.2.js';  
</script>
```

```
<script type = 'text/javascript'>  
alert($)  
</script>
```


Firefox, Opera	显示\$函数
IE	显示\$函数
Chrome, Safari	未定义

# 监听加载完成事件

- 同时监听onload和onreadystatechange事件以兼容各主流浏览器

```
var script = document.createElement('script');
script.type = 'text/javascript';
script.src = scriptFile;
script.onload = onLoad;
script.onreadystatechange = function() {
    if (script.readyState == 'loaded' || script.readyState == 'complete')
        onLoad();
};
function onLoad() {
    if (script.loaded) return;
    script.loaded = true;
    // onload code goes here
}
```


# 通过innerHTML载入动态内容？

  
`var x = document.getElementById('div');  
x.innerHTML = AjaxResponse;`

不安全来源

  
`x.textContent / innerText = AjaxResponse;`

纯文本内容

`var script = document.createElement('script');  
script.src = scriptFile  
or  
script.appendChild(document.createTextNode(scriptText));`  


可控来源

# 语言特性

---



# 函数声明与函数表达式

~~var a = function b() {};  
alert(b); ???~~

b到底有没有定义？能不能被外部访问？



var a = function() {};

函数表达式用匿名函数

## 函数声明与函数表达式(2)

```
function test() {  
  if (true) {  
    function boo() { ... }  
  } else {  
    function boo() { ... }  
  }  
  boo();  
}
```

在不同逻辑块内用相同函数名声明函数

```
function test() {  
  if (true) {  
    function boo1() { ... }  
  } else {  
    function boo2() { ... }  
  }  
  ...  
}
```

避免函数名重名

```
function test() {  
  if (true) {  
    var boo = function() { ... }  
  } else {  
    var boo = function() { ... }  
  }  
  boo();  
}
```

使用函数表达式

## 函数声明与函数表达式(3)

~~function Object.method() { }~~

✓  
var Object = {  
 method: function() { }  
};

✓  
Object.method = function() { }




# DOM实现差异


---



# 事件监听



```
window.attachEvent('onload', onLoadHandler);  
window.detachEvent('onload', onLoadHandler);
```



```
if (window.attachEvent)  
    window.attachEvent('onload', onLoadHandler);  
else if (window.addEventListener)  
    window.addEventListener('load',  
onLoadHandler);  
else  
    ...
```

# 获取元素对象

`<div id='divID' name='divName'></div>`

~~`divID.style.display = 'none';`  
`divName.style.display = 'none';`  
`document.divID.style.display = 'none'`  
`document.divName.style.display = 'none'`  
`document.all['divID']`  
`document.getElementById('divID');`~~



`document.getElementById('divID');`

## 对象的native方法



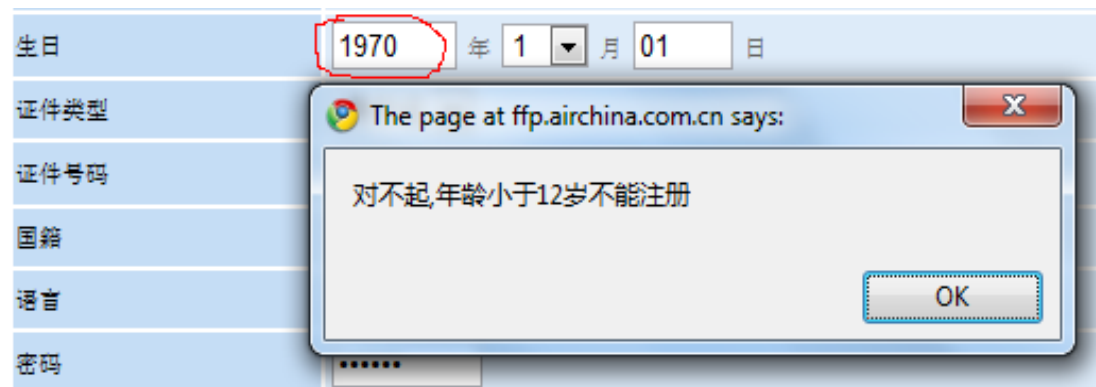
```
var getID = document.getElementById;  
var x = getID('xID');
```



```
var $getID = function(id) {  
    return document.getElementById(id);  
};
```

```
var x = $getID('xID');
```

# 系统时间



```
gd2011d.jsp
67  today = new Date();
68  if ( today.getFullYear() - document.submit_application.year.value < 12){
69      alert("对不起,年龄小于12岁不能注册");
70      document.submit_application.year.focus();
71      return false;
72  }
```

~~new Date().getYear()~~

new Date().  
getFullYear()

# 区别HTML Attribute和DOM Object Property

```
<div id='myDiv' class='divClass' style='float:left;width:200px'></div>
```

```
var div = document.getElementById('myDiv');
```

HTML Attribute	DOM Object Property
div.getAttribute('id') = 'myDiv'	div.id = 'myDiv'
div.getAttribute('class') = 'divClass'	div.className = 'divClass'
div.getAttribute('style') = 'float:left;width:200px'	div.style = { CSSStyleDeclaration }

## 区别HTML Attribute和DOM Object Property (2)

- **名称不等价**: 多数情况下HTML Attribute和DOM Object Property 共享同样的名称, 如id, style等属性, 但也有如下原因导致有些属性不能采用相同的名称:
  - 避免和脚本语言中的关键字重名, 如class, for等属性
  - 在脚本语言中不是合法的标识符, 如包含'-'等字符的属性
- **类型不等价**: HTML Attribute的值类型是字符串, DOM Object Property则可能是不同的JavaScript变量类型
- **值相关联却不相等**: DOM Object Property基于HTML Attribute计算出来, 其中某一个值的改变通常会影响到另一个值。

# IE6/IE7/IE8(Q)的行为

```
<div id='myDiv' class='divClass' style='float:left;width:100px'></div>  
var div = document.getElementById('myDiv');
```

## 名称等价：

```
div.getAttribute('class') ==> null  
div.getAttribute('className') == div.className ==> 'divClass'
```

## 类型和值等价：

```
div.getAttribute('style') == div.style ==> { CSSStyleDeclaration }
```

## 没有区分HTML Attribute和DOM Object Property

- element.getAttribute('xxx') <==> element.xxx
- element.setAttribute('xxx', value) <==> element.xxx = value



# 正确方式

- 严格区别HTML Attribute和DOM Object Property

```
<div id='myDiv' country='China'>中国</div>
```

```
var div = document.getElementById('myDiv');  
div.city = { ... };
```

div.id <==> div.getAttribute('id') ==> 标准HTML Attribute

div.getAttribute('country'); ==> 自定义HTML Attribute

div.city['Beijing']; ==> 自定义DOM Object Property

div.innerHTML; ==> 内置DOM Object Property

# 还有很多...

- CSS Filter
- *DHTML Behavior*
- clipboardData
- 设置首页收藏
- 获取元素的计算后样式
- createElement
- scrollLeft/scrollTop
- Node和Element对象的属性和方法
- ...

# 总结

- 浏览器检测
- 视频和插件
- 异步加载脚本
- 语言特性
- DOM实现差异

# 几点建议

---

# 建议

- 在不同平台上测试
- 有疑问时看标准
- 第三方跨平台库
  - Google AJAX Library API
  - Closure, SWFObject
  - jQuery, Mootools, Dojo, Ext JS
- 各浏览器的Debugger Tool

# 相关资料

- 跨浏览器开发论坛  
<http://forum.csdn.net/SList/CrossBrowser/>
- w3help主页  
<http://www.w3help.org>
- W3C（包括DOM和CSS等相关的标准）  
<http://www.w3.org>
- ECMAScript  
<http://www.ecmascript.org/>
- Google AJAX Library  
<http://code.google.com/apis/libraries/>
- SWFObject  
<http://code.google.com/p/swfobject>
- Closure  
<http://code.google.com/closure/>

谢谢

---

Q&A

