

Google  
Developer  
Day 2009

# iGoogle Gadgets 开发: 界面设计和性能优化

梁宇

2009年6月4日

Google  
Developer  
Day2009

# iGoogle - 我的个性化首页

The screenshot shows the iGoogle homepage with the following components:

- Header:** iGoogle logo, search bar, and navigation links like "Google 搜索" and "手气不错".
- Navigation:** "主页" and "添加标签页" buttons.
- 谷歌资讯 (Google News):** A list of news items with categories like "焦点", "财经", "科技", "体育", "娱乐".
- 谷歌中文热榜 (Google Chinese Hot Topics):** A table of trending topics in Chinese, including "张悬", "赵又廷", "张悬 绯闻", "九孔模仿小沈阳", and "畅享人生年金保险产品".
- 谷歌网站导航 (Google Website Navigation):** A list of popular websites like "新浪", "淘宝网", "腾讯", "雅虎", "网易", "百度贴吧", "天涯社区", etc.
- 工具箱 (Toolbox):** A section with "Google 工具" and "待办事项" buttons, and a list of popular services like "大学搜索", "地图", "工具栏", "快讯", "热搜".

# 议程 – 界面设计

- 受用户欢迎的Gadget的7条设计原则
- Gadget界面和风格样式的细节

# 原则1：使用标准高度

- Gadget的默认高度是200。
- 用户喜欢将相同高度的gadgets对齐。



- 如果gadget内容太长，请使用标签页(tabs)。



## 原则2：让用户参与


- 一般的feed gadget只是单向的把信息传递给用户。
- 通过让用户参与，你将不再是简单的把信息提供给用户，而是让用户从中能获得参与的乐趣。



## 原则3： 保持内容每天更新

- 用户喜欢把经常更新的gadgets放在自己的iGoogle主页上。
- 最流行的gadgets中，很多的标题都是“每天一个……”。例如，“每天一个笑话”，“每天一句格言”，“每天一个单词”，等等。

### Daily Horoscope for Aries [edit](#)

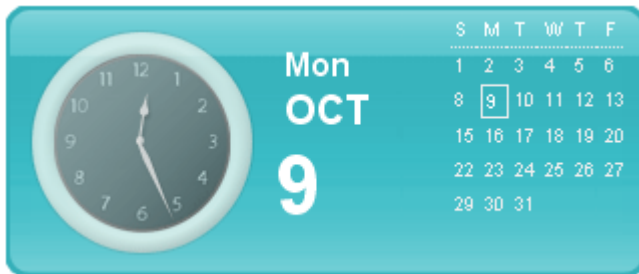


You must find ways to demonstrate your creative side today, for the Moon is in your 5th House of Self-Expression. You might not be able to concentrate at work, for you would much rather be playing. It can be quite a challenge to suppress your desires now, for they come to the surface so easily. Even still, pleasure will be more enjoyable once you've fulfilled your obligations.

Monday, August 21, 2006  
[more from tarot.com »](#)

## 原则4：风格多样，充满趣味性

- 在风格上做一些有趣的变化更容易得到用户的青睐。





## 原则5：做一个好公民

- 每个gadget在界面设计上有自己的独立性，但是请给用户持续而一致的体验。
- 记住：gadget是用户iGoogle主页的一部分。请考虑整个页面的用户体验。
- 用户厌恶那些太商业的gadget。例如：显示很大的logo，很刺眼的背景色，大片的广告，或是千方百计诱使用户点击链接来到自己的网站。

## 原则6：避免获取个人信息

- 用户出于自我保护，通常会删掉那些自己不熟悉的同时又要求填入个人信息的gadget。
- 如果一定需要用户输入个人信息，请考虑以下几点：
  - 让你的gadget在有或没有用户个人信息的情况下都能正常使用。
  - 明确的告诉用户，输入个人信息对使用gadget有何帮助。
  - 让人信任你的gadget：注明你的公司或gadget的目的。

## 原则7：节省空间

- 用户iGoogle主页上的空间是非常宝贵的。
- 用户不喜欢占据大量空间的gadget。
- 尽可能的减小没必要的空白空间。
- 考虑使用动态高度（dynamic height library）。

# 议程 – 界面设计

- 受用户欢迎的Gadget的7条设计原则
- Gadget界面和风格样式的细节

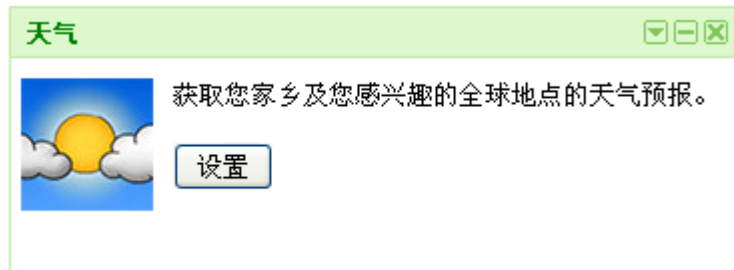
# 宽度及不同的屏幕大小

- 屏幕可能从800x600变化到1600x1200甚至更高。
- Gadget必须要适应从250px到500px的宽度变化（用firefox插件来测试不同宽度）。
  - 对于输入框或表格这样的页面元素，使用width=100%，不要使用固定宽度。这样，它们能适应不同的分辨率和窗口大小。
  - 将页面内容居中
  - 小心的使用表格能使内容不会因为窗口大小变化而换行。



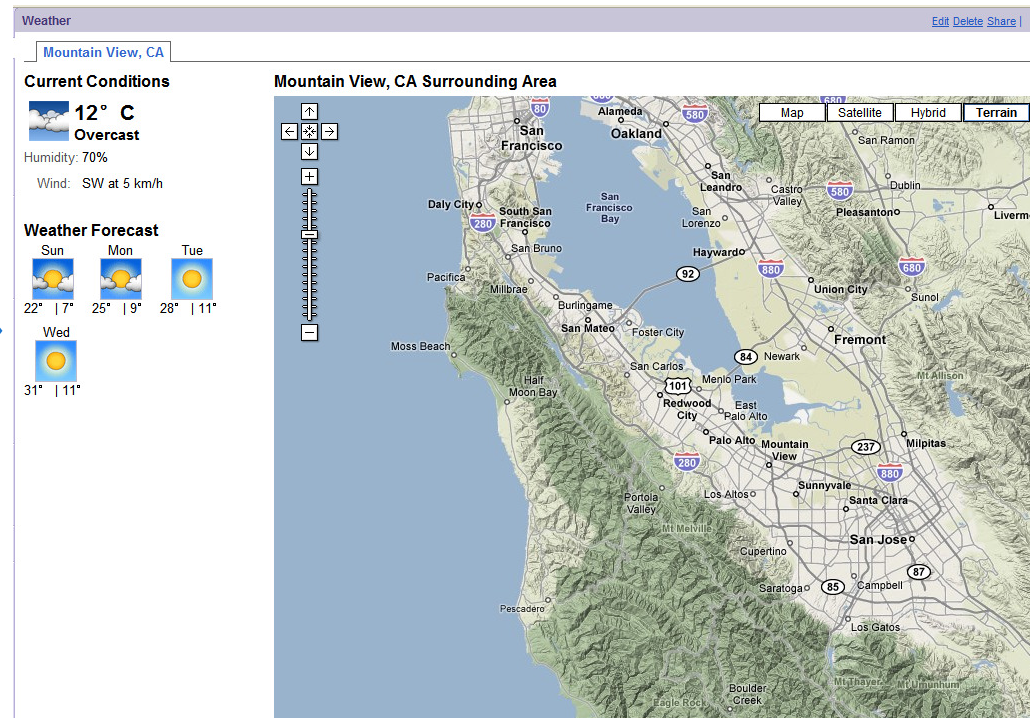
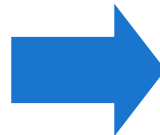
# 图片

- 图片虽然慢，但是能比文字传达更丰富的意思。
- 设置图片的高度和宽度，使得gadget在图片没有load完时，也能正常显示。
- 放一个小图标在欢迎模式的左边，告诉用户他需要输入一些信息。



# 信息的呈现

- 通常要呈现的信息太多，而gadget的空间有限。
- 两种常用的方法：
  - 让用户打开新窗口到另外一个页面（Wikipedia gadget）。
  - 在gadget内动态的生成和显示信息（Google Map Search）。
- Home View vs Canvas View



## 设置和修改用户选项

- 如果你的gadget有用户选项是必须填的，使用欢迎界面。



- 让用户可以控制信息的隐藏和显示。



# Gadget 性能优化

---

# 议程 — 性能优化

- 网站性能优化技巧
- Gadget常用性能优化技巧
- Gadget特有性能优化技巧

# 网站性能优化技巧



- 《 **High Performance Web Sites** 》  
作者: Steve Souders
- 《 **Even Faster Web Sites** 》  
作者: Steve Souders
- Best Practices for Speeding Up Your Web Site  
– <http://developer.yahoo.com/performance/rules.html>

# 议程 — 性能优化

- 网站性能优化技巧
- **Gadget**常用性能优化技巧
- Gadget特有性能优化技巧

# 减小传输大小 1

- 减小Javascript, CSS和HTML的大小 (量变 -> 质变)
  - Gzip压缩
    - 压缩后, 通常可以减小70%的大小。
  - 去掉不必要的字符 (空格, 换行, 引号, 分号等) 和注释。
  - 编译优化Javascript
    - YUI Compressor: <http://developer.yahoo.com/yui/compressor/>

```
<script>
var height = 3;
Var width = 5;
var area = height * width;
</script>
```

```
<script>var a=3,b=5,c=a*b;</script>
```

## 减小传输大小 2

- 优化图片
  - ImageMagick: <http://www.imagemagick.org/>
    - 查看gif图片颜色
    - 把gif图片转成png。值得一试!
  - Pngcrush: <http://pmt.sourceforge.net/pngcrush/>
    - 优化png图片
  - Jpegtran: <http://jpegclub.org/>
    - 优化jpg图片

## 减小传输大小 3

- 不要把大图片缩小显示在gadget里面。
  - 把400x200的图片直接按200x100的大小显示。糟糕!
  - 把400x200的图片先压缩成200x100大小，再按200x100的大小显示在gadget里面。棒极了!

## 减小传输大小 4

- 把Javascript移到外部引用文件中
  - 适用于Javascript 代码比较多的情况。
  - 确保Javascript外部引用文件能缓存在浏览器上。



## 避免不必要的HTTP请求

- 尽可能的合并HTTP请求
  - 合并外部Javascript文件
  - 合并外部CSS文件
  - 用CSS Sprite合并多个图片



## 延缓不必要的下载

- 把用户不会立即看到的数据放到最后加载。例如：
  - Google主页会在页面底部加载Google搜索建议的Javascript代码。
  - Gadget里面，等用户点击其它标签之后再加载里面内容。
  - 可以先加载并显示一小部分内容，等用户希望看到更多时，再加载其余部分内容。

## 并行化下载

- 节省建立连接的时间。
- 把Javascript外部引用文件放在页面底部，因为它会阻碍浏览器并行下载其它的资源。
- 将所有请求分配到2~4个Hostname上
  - IE和Firefox默认只能对每个Hostname支持2个并行请求。

## 加快页面渲染

- 避免在gadget里使用外部CSS文件。
  - Gadget里的外部CSS只能放在HTML body里。
  - HTML body里的外部CSS会阻碍页面逐步渲染显示 (progressive rendering)。
- 指定图片的高和宽。
- 避免使用table来控制整个gadget的页面布局。
- 避免多个<script>和<style>模块的交替。

# 议程 — 性能优化

- 网站性能优化技巧
- Gadget常用性能优化技巧
- Gadget特有性能优化技巧

## 缓存静态内容

- `_IG_GetCachedUrl(url, {refreshInterval:num})`
  - 由iGoogle作为proxy将静态内容返回给用户，减少了第三方服务器的负担。
  - 指定缓存时间，减少用户反复下载相同静态内容。

```
var image = document.createElement("image");
image.src = _IG_GetCachedUrl(
    "http://www.example.com/my_image.jpg",
    {refreshInterval: 3600});
_gel("container").appendChild(image);
```

## 为远程内容设置适当的缓存时间

- `_IG_FetchContent(url, func, {refreshInterval: num})`
  - 远程内容会缓存在用户的浏览器上，时间由`refreshInterval`指定。
  - `refreshInterval`的默认值是3600 (1小时)。
  - 如果你的远程内容会在一个小时内更新多次，可以减小`refreshInterval`的值，例如：`{ refreshInterval: 1200 }`

## 避免使用Google Analytics

- Google Analytics会增加额外的请求。
- 建议使用 <http://www.google.com/ig/directory?url=<gadget-url>> 来查看gadget每周的使用情况。



## 使用 `prefs.getString("xxx")` 代替 `__MSG_xxx__`

- `__MSG_xxx__` 使用消息变量内容直接替换掉自身，如果 `xxx` 消息变量的内容很长，会增加不必要的字节。

### Message bundle file:

```
<messagebundle>
  <msg name="xxx">abcdefghijklmnopqrstuvxyz</msg>
</messagebundle>
```

### Gadget Spec:

```
<script>
  var msg1 = "__MSG_xxx__";
  var msg2 = prefs.getMsg("xxx");
</script>
```

### Gadget iframe source code:

```
<script>
  var msg1 = "abcdefghijklmnopqrstuvxyz";
  var msg2 = prefs.getMsg("xxx");
</script>
```

## 使用<preload>提前加载数据内容

- <preload>使获取远程内容的API (\_IG\_Fetch...)不再使用AJAX发送异步请求来获取数据内容，而是把数据嵌入到gadget的html里面。
- 对获取远程内容的API (\_IG\_Fetch.....)是透明的。
- 优点：节省一个额外的异步数据请求，可以直接获得数据。
- 缺点：数据无法被缓存。

### Gadget Spec:

```
<ModulePrefs title="xxx" ...>
  <preload href="http://www.example.com/data" />
</ModulePrefs>

.....
<script>
  _IG_FetchContent("http://www.example.com/data",
    function(response) { ..... },
    { refreshInterval: 3600 });
</script>
```

## 不要提前加载其他标签的内容

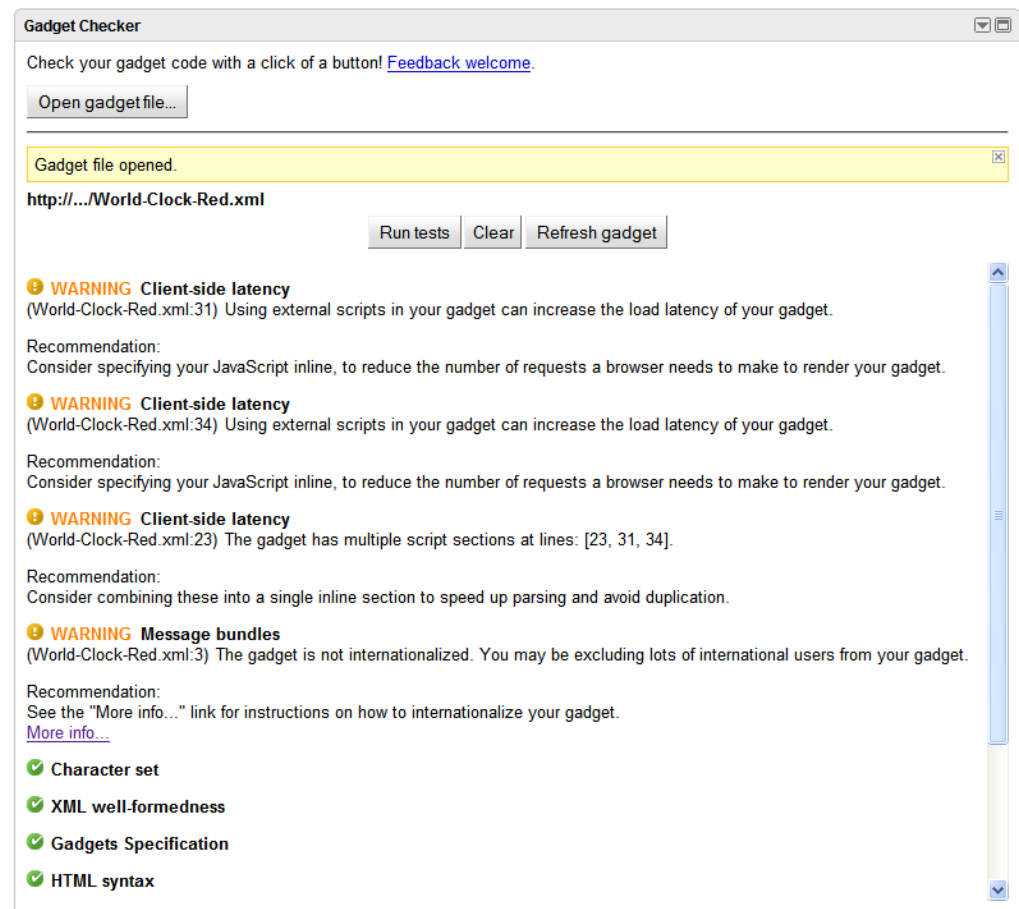
- 当用户点击了他想看的标签时，再去加载和显示标签的内容。

### Gadget Spec:

```
<script>
.....
var tabs = new _IG_Tabs(__MODULE_ID__, "Two");
tabs.addTab("One", "one_id", callbackFunction);
tabs.addTab("Two", {
    contentContainer: _gel("two_id"),
    callback: callbackFunction
});
function callbackFunction(tabID) {
    // Fetch content
    // Render content
    _gel(tabID).innerHTML = renderedContent;
}
</script>
```

# Gadget Checker

- <http://www.google.com/ig/modules/codechecker/codechecker.xml>
- 检查本地gadget文件， gadget URL和GGE里的gadget。
- 检查要点：
  - Unused features
  - Character set
  - XML well-formedness
  - Gadgets Specification
  - HTML syntax
  - Client-side latency
  - Message bundles
  - Internationalized metadata
  - Referenced files available
  - Internationalized HTML



想了解更多信息？访问[code.google.com](http://code.google.com)

---

谢谢!

---