Web前端基础进阶

骆勤

百度资深研发工程师

课程提纲

- Ajax
- HTML5
- WebSocket
- 应用进阶

一门"大杂烩"技术

Ajax

Baidu 百度

Ajax

- 什么是 Ajax
  - 即Asynchronous JavaScript and XML（异步的JavaScript与XML技术），指的是一套综合了多项技术的浏览器端网页开发技术（后续XML逐渐进化为JSON）
  - 无须刷新页面而从服务器取得数据
  - 核心XMLHttpRequest对象（XHR）

Ajax

- 为何而生
  - 服务器处理每一个用户请求都需要重新加载网页，导致过多的冗余加载
  - 刷新后所有的页面状态都会消失，再重新载入后，即使只是一部分页面元素改变也重新载入了整个页面，不仅要刷新改变的部分，连没有变化的部分也要刷新，加重了服务器的负担

- 先看几个著名Case：
  - Gmail
  - 百度音乐人
  - Google Instant

Ajax

◆ 怎么用起来

通过挂在Window下的XMLHttpRequest对象：

IE5中首先引入，通过MSXML库中的ActiveX对象实现

IE7+、FF、Chrome、Opera、Safari支持原生的XHR对象

```
var xhr = new XMLHttpRequest();
```

```
function createXHR(){
        if (typeof XMLHttpRequest != "undefined"){
                return new XMLHttpRequest();
        } else {
                throw new Error("No XHR object available.");
        }
}
```

Ajax

◆ XHR的接口调用

open()

参数：

1. 要发送的请求类型(get/post)
2. 请求url
3. 是否异步请求(Boolean)，通常建议默认为异步(jQuery等)

```
var xhr = createXHR();
xhr.open('GET', 'ajaxHandler.php', true);
```

open()方法不会真正发送请求，而是启动一个请求待发送

Ajax

◆ XHR的接口调用

send()

参数：

请求中要发送的数据，若无则传入null

xhr.send(null);

此时将请求发送到服务器，收到服务器响应后，数据会自动填充XHR对象的的属性：

- responseText
- responseXML
- status
- statusText

Ajax

◆ XHR的接口调用

收到响应后：

1. 检查status属性，来确定相应已经成功返回HTTP状态码

2. 检测readyState属性，表示请求/响应过程的当前活劢阶段，可取值如下：

0 —— 未初始化，未调用open()
1 —— 启劢，已调用open()，未调用send()
2 —— 发送，已调用send()，未收到响应
3 —— 接收，收到部分数据
4 —— 完成，数据接收完成

readyState值变化, 触发readystatechange事件

Ajax

◆ XHR的接口调用

响应处理示例：

```
var xhr = createXHR();
xhr.onreadystatechange = function(event){
    if (xhr.readyState == 4){
        if ((xhr.status >= 200 && xhr.status < 300) || xhr.status
        == 304){
            alert(xhr.responseText);
        } else {
            alert("Request was unsuccessful: " + xhr.status);
        }
    }
};
xhr.open("GET", "ajaxHandler.php", true);
xhr.send(null);
```

Ajax

- 特点
  - AJAX应用可以仅向服务器发送并取回必须的数据，服务器和浏览器之间交换的数据大量减少（约原来的5%）
  - AJAX不是指一种单一的技术，而是有机地利用了一系列相关的技术。虽然其名称包含XML，但实际上数据格式可以由JSON代替，进一步减少数据量，形成所谓的AJAJ
  - 在不更新整个页面的前提下维护数据
  - Ajax不需要任何浏览器插件，JavaScript在浏览器上执行

Ajax

优势:

1）被主流浏览器广泛支持，无需插件, jQuery，Zepto等简化封装
2）优秀的用户体验
3）提高web性能
4）减轻服务器和带宽负担

不足:

1）浏览器兼容需要处理(IE7+)
2）对搜索引擎支持不足（Spider已支持）
3）页面的前进后退按钮失效(HTML5 PushState)
4）不便于url对外传播

Silver Bullet?

Ajax

- 应用场景？
- Live Coding?

http://labs.music.baidu.com/demo/fe8899/backforward/

Ajax

- Reference
  - https://zh.wikipedia.org/zh-cn/AJAX
  - http://www.cnblogs.com/xmphoenix/archive/2011/11/21/2257651.html
  - http://api.jquery.com/jQuery.ajax/
  - https://mail.google.com/
  - http://y.baidu.com/

一种"古老"的新发明

HTML5

Baidu 百度

HTML 5

■ 进化史

1991HTML 1996CSS 1 + JS 1998CSS2 2002Tableless 2008HTML5

91			94		96	97	98		00		02			05				09

1994HTML 2 1997HTML 4 2000XHTML1 2005AJAX

HTML 5

- 广义论HTML5时，实际指的是包括HTML、CSS和JavaScript在内的一套技术组合

- 对于HTML4改进点：
  - 强化了Web 网页的表现性能。
  - 追加本地数据库等 Web 应用的功能
  - 减少浏览器对插件的依赖

HTML 5

◆ 与HTML4不一样的地方：

文件类型声明仅需：<!DOCTYPE HTML>

新元素：section, video, progress, nav, meter, time, aside, canvas, command, datalist, details, embed, figcaption, figure, footer, header, hgroup, keygen, mark, output, rp, rt, ruby, source, summary, wbr

input元素的新类型：date, email, url等

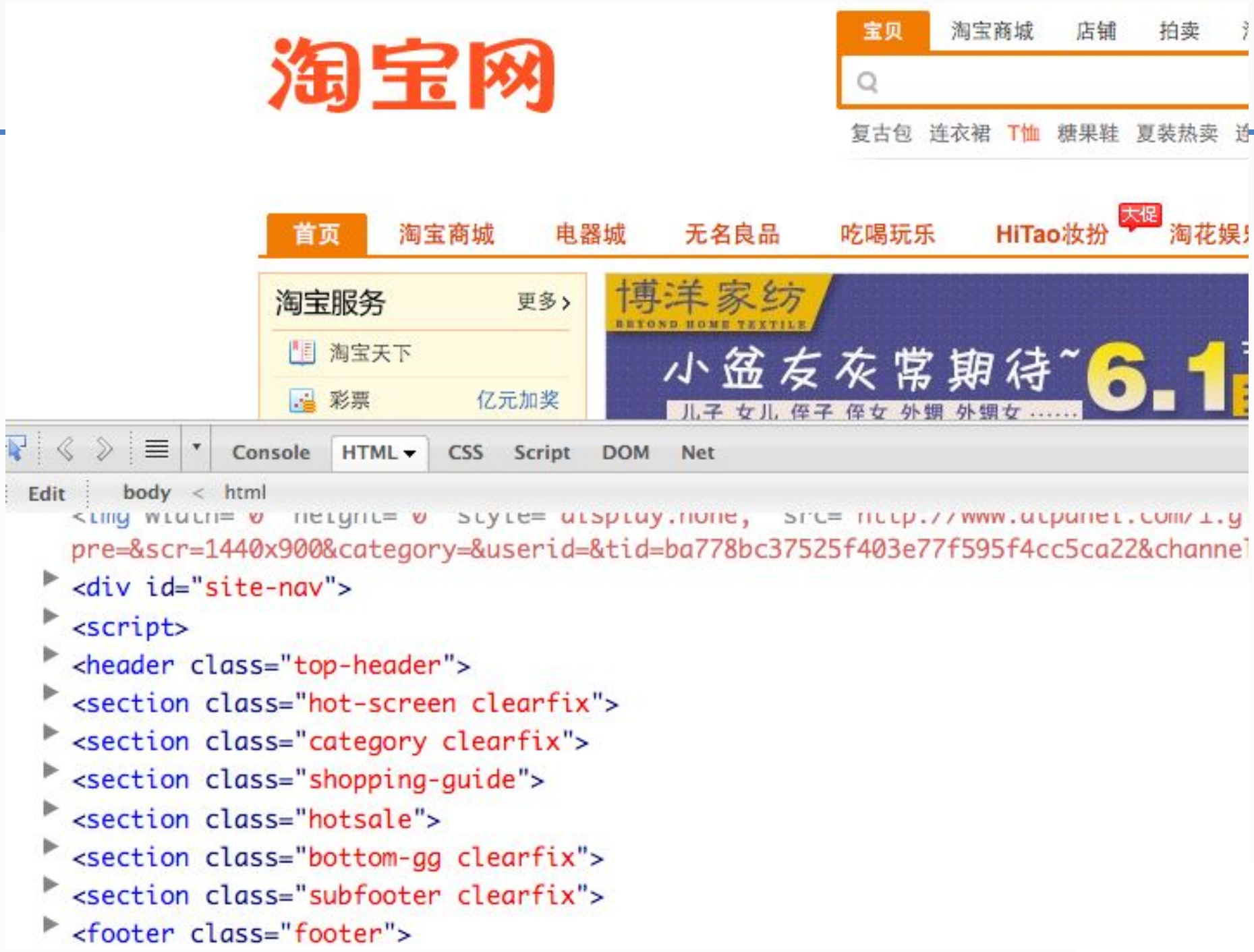
新属性：ping（用于a与area）, charset（用于meta）, async（用于script）
全域属性：id, tabindex, repeat
新的全域属性：contenteditable, contextmenu, draggable, dropzone, hidden, spellcheck

移除元素：acronym, applet, basefont, big, center, dir, font, frame, frameset, isindex, noframes, strike, tt

HTML5
SEMANTICS

Semantics

```
<article>
  <header>
    <h1>文章标题</h1>
  </header>
  <p>文章内容</p>
  <footer>
    <a href="?comments=1">评论</a>
  </footer>
</article>
```


淘宝网
宝贝
淘宝商城
店铺
拍卖
复古包
连衣裙
T恤
糖果鞋
夏装热卖
首页
淘宝商城
电器城
无名良品
吃喝玩乐
HiTao妆扮
大促
淘花娱
淘宝服务
更多
淘宝天下
彩票
亿元加奖
博洋家纺
BEYOND HOME TEXTILE
小盆友们常期待~6.1
儿子 女儿 侄子 侄女 外甥 外甥女 .....
Console
HTML
CSS
Script
DOM
Net
Edit
body < html
pre=&scr=1440x900&category=&userid=&tid=ba778bc37525f403e77f595f4cc5ca22&channel

<script>
<header class="top-header">
<section class="hot-screen clearfix">
<section class="category clearfix">
<section class="shopping-guide">
<section class="hotsale">
<section class="bottom-gg clearfix">
<section class="subfooter clearfix">
<footer class="footer">

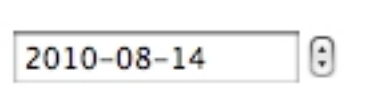
```
<meter min="0" max="100" low="40" high="90" optimum="100"
value="91">A+</meter>
```


```
<progress>正在处理</progress>
```


```
<progress value="75" max="100”>完成75%</progress>
```

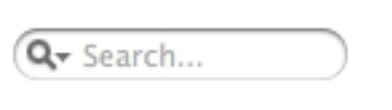
```
<input type="email" value="some@email.com" />
```


```
<input type="date" min="2010-08-14" max="2011-08-14"
value="2010-08-14"/>
```




```
<input type="range" min="0" max="50" value="10" />
```

```
<input type="search" results="10" placeholder="Search..." />
```



type="email"

type="tel"


OFFLINE &
STORAGE

Once upon a time…

Web Storage

```
localStorage.setItem('lng', map.getCenter().lng);
localStorage.setItem('lat', map.getCenter().lat);
localStorage.setItem('mapZoom', map.getZoom());

var lng = localStorage.getItem('lng');
```

Web SQL Database

```
var db = window.openDatabase('MyDB', '1.0', 'my database', 5*1024*1024);
db.transaction(function(tx){
  tx.executeSql("CREATE TABLE IF NOT EXISTS test (id unique, text)");
  tx.executeSql("INSERT INTO test (id, text) VALUES (1, 'my data')");
  tx.executeSql("SELECT * FROM test", [], successCallback);
});
```

Application Cache


```
<!DOCTYPE html>
<html manifest="a.manifest">
.
.
</html>
```

```
#comments begin with the hash symbol
CACHE MANIFEST
# files to cache
html5.css
index.html
lake-tahoe.JPG
#do not cache signup page
NETWORK
signup.html
FALLBACK
signup.html offline.html
/app/ajax/ default.html
```

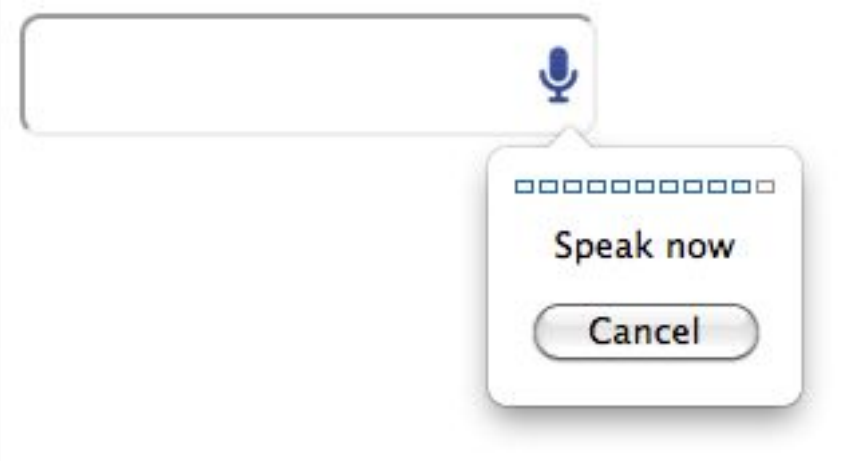

DEVICE
ACCESS

Geolocation

```
if (navigator.geolocation) {
  navigator.geolocation.getCurrentPosition(function(pos){
    alert(pos.coords.latitude + ',' + pos.coords.longitude);
  })
}
```

Speech Input

```
<input type="text" x-webkit-speech />
```



Device Orientation

```
window.addEventListener("deviceorientation", function(e){
  var a = e.alpha;
  var b = e.beta;
  var g = e.gamma;
}, false);
```

CONNECTIVITY

WebSocket

WebSocket v2.00

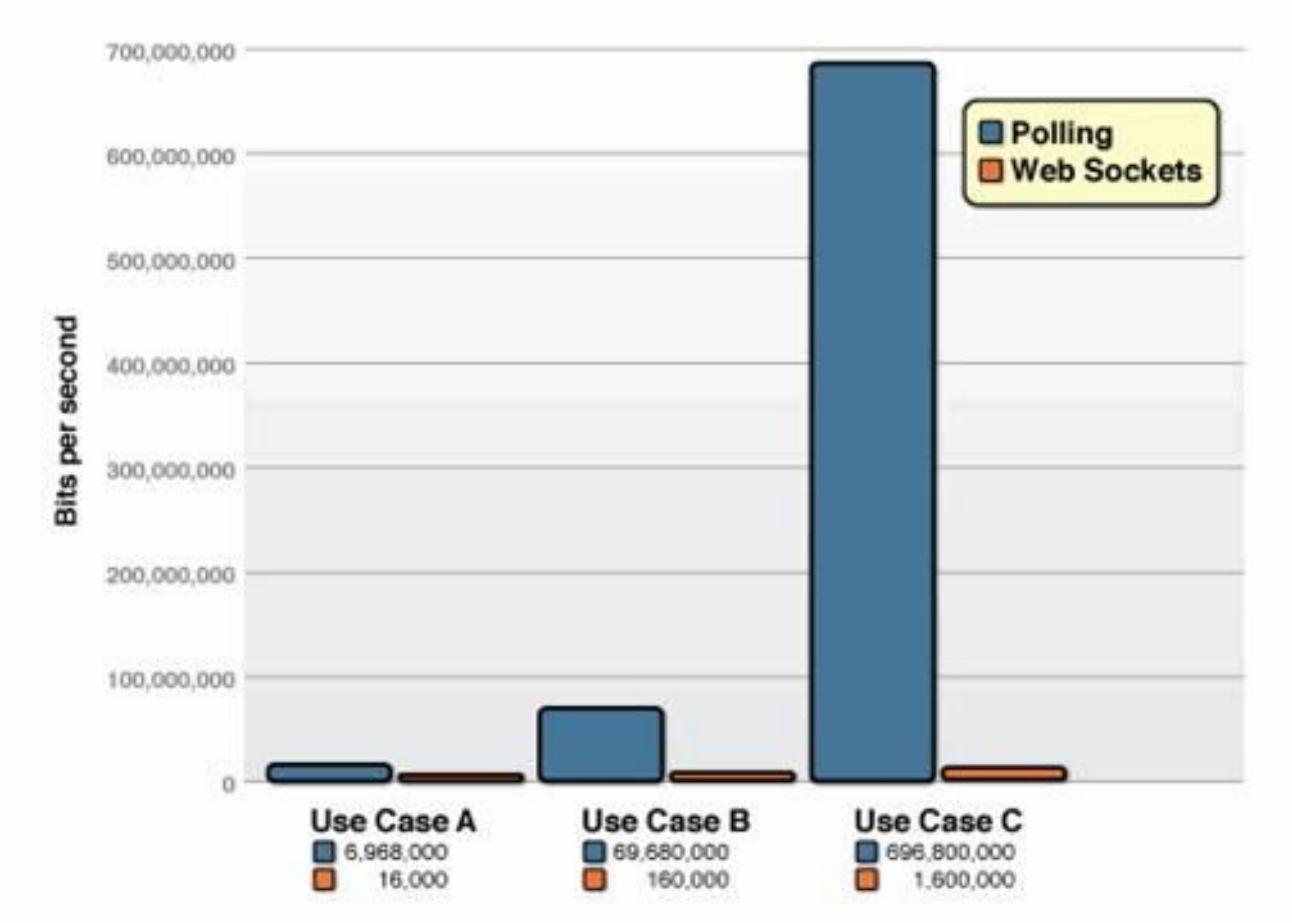
WebSocket - status 0
Welcome - status 1
Sent: hello
Received: hello human
Sent: hi
Received: zup human

Send Quit

Commands: hello, hi, name, age, date, time, thanks, bye

后续课程会着重介绍.

WebSocket


700,000,000
600,000,000
500,000,000
400,000,000
300,000,000
200,000,000
100,000,000
0
Bits per second
Polling
Web Sockets
Use Case A
6,968,000
16,000
Use Case B
69,680,000
160,000
Use Case C
696,800,000
1,600,000

MULTIMEDIA

Audio & Video

<audio />

<video />

Audio

```
<audio src="vincent.mp3" controls="controls" />
```

Video

```
<video src="mapapi.mp4" controls="controls" />
```

3D, GRAPHICS,
EFFECTS

Pixel Manipulate in Canvas

jiazheng.baidu.com/Study/html5/canvas/pixel_manipulate.html

Grayscale

Black&White input threshold: 0.5

Ordered Dithering choose patterns: 1

FloydSteinburg

Restore

Baidu 百度

Pixel Manipulate in Canvas

jiazheng.baidu.com/Study/html5/canvas/pixel_manipulate.html

Grayscale

Black&White input threshold: 0.5

Ordered Dithering choose patterns: 1

FloydSteinburg

Restore

Baidu 百度


HelloRacer™ WebGL
helloracer.com/webgl/
57 FPS (34-60)
HelloRacer™ WebGL — Created by HelloEnjoy™ — Powered by three.js
Santander
acer
AMD
BRIDGESTONE
POTENZA
Use WASD or cursor keys to drive, space for hand brake and enter to change camera.

http://www.webhek.com/misc/webgl-jellyfish

More WebGL： http://www.chromeexperiments.com/

CSS3
STYLING

Text stroke

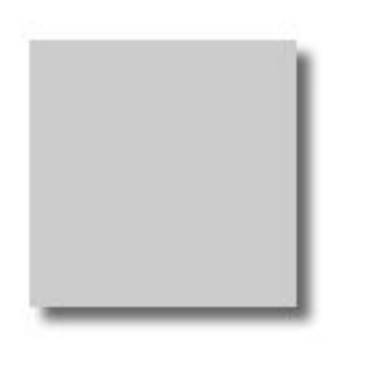
```
div {
  -webkit-text-fill-color: white;
  -webkit-text-stroke-color: black;
  -webkit-text-stroke-width: 1px;
}
```

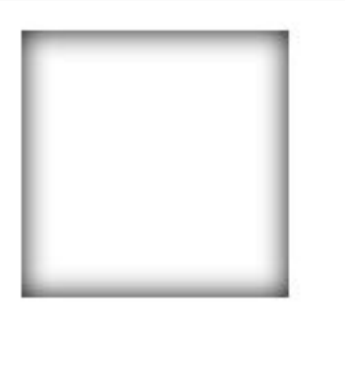
Some Text Here

阴影

```
text-shadow: 4px 4px 1px #aaa;
box-shadow: 1px 1px 2px #fff;
box-shadow: inset 0 0 10px #000;
```

STROKE OF GENIUS





Gradients

```
border-radius: 10px
background-image: -moz-linear-gradient(#fff, #000);
background-image: -moz-linear-gradient(30deg, #000, #fff 75%, #000);
```

Masks

- Box mask
- Gradient mask
- Border radius mask
- Svg mask

Masks

```
<img src="kate.png" style="-webkit-mask-box-image: url(mask.png) 75
stretch;">
```

The result looks like this:

Masks

```
<img src="kate.png" style="-webkit-mask-image: -webkit-gradient(linear, left top, left bottom, from(rgba(0,0,0,1)), to(rgba(0,0,0,0)))">
```

Masks

```
<img src="kate.png" style="-webkit-border-radius: 10px; -webkit-mask-image:
-webkit-gradient(linear, left top, left bottom, from(rgba(0,0,0,1)),
to(rgba(0,0,0,0)))">
```

Masks

can be applied as a mask like so:

```
<img src="kate.png" style="-webkit-mask-image: url(circle.svg)">
```

The end result is shown below:

Reflections

```
<img src="ship.jpg" style="-webkit-box-reflect:below 3px
    -webkit-gradient(linear, left top, left bottom, from(transparent),
    color-stop(0.5, transparent), to(white));">
```

Transitions

- implicit animations
- -webkit-transition-property(none | all | ,,)
- -webkit-transition-duration
- -webkit-transition-delay
- -webkit-transition-timing-function(cubic-bezier)

Baidu 百度

HTML 5

◆ Live code， 更多示例:

git clone https://github.com/remy/html5demos.git

Filter demos: canvas classlist contenteditable dataset dnd events file-api file geolocation getUserMedia hidden history manifest offline postMessage sql-database storage svg video websocket workers xhr2

Demo	Support	Technology
Stream video and filter with canvas		getUserMedia canvas
Stream video to the browser Also works on Opera Mobile 12		getUserMedia
Drag and drop and XHR upload		file dnd xhr2
Hidden property		hidden
Simple class manipulation		classlist
Storage events		storage
dataset (data-* attributes)		dataset
History API using pushState		history
Browser based file reading Not part of HTML5		file-api

HTML 5

- Reference
  - Wiki: https://zh.wikipedia.org/zh-cn/HTML5
  - Demos：http://html5demos.com/
  - 源码：https://github.com/remy/html5demos
  - Canvas: http://javascript.ruanyifeng.com/htmlapi/canvas.html
  - WebGL水母: http://www.webhek.com/misc/webgl-jellyfish
  - Ray Cast 3D Algorithm：http://www.permadi.com/tutorial/raycast/rayc11.html
  - https://dev.opera.com/articles/3d-games-with-canvas-and-raycasting-part-1/

有意思的东东

WebSocket

Baidu 百度

Websocket

- 如何实现Web端的消息推送？
  - 模拟长连接？
  - 基于Flash AS/Java Applet插件?
- 基于前面所述的Ajax如何达到实时推送效果
  - 轮询？

Baidu 百度

Websocket

◆ **背景**

- **轮询的资源消耗**

在特定的的时间间隔，由浏览器对服务器发出HTTP请求，服务器返回最新的数据给浏览器，缺点浏览器需要不断的向服务器发出请求

- **Comet长连接形式并发数的影响**

基于AJAX，但这种技术虽然可达到双向通信，但依然需要发出请求，而且在Comet普遍采用长链接，会大量消耗服务器带宽和资源

Websocket

- WebSocket是HTML5开始提供的一种在单个 TCP 连接上进行全双工通讯的协议，该协议包含一个握手和一个基本消息分帧、分层通过TCP

- 在WebSocket API中，浏览器和服务器只需要做一个握手(handshake)的动作，两者之间形成了一条快速安全的信息通道，两者之间就直接可以数据互相传送

```
+---------+    1.发送Sec-WebSocket-Key        +----------+
|         | ---------------------------------> |          |
|         |    2.加密返回Sec-WebSocket-Accept  |          |
| client | <--------------------------------- | server |
|         |    3.本地校验                      |          |
|         | ---------------------------------> |          |
+---------+                                    +---------+
```

Websocket

◆ 抓包示例（Chrome）：

浏览器请求

```
GET / HTTP/1.1
Upgrade: websocket
Connection: Upgrade
Host: example.com
Origin: null
Sec-WebSocket-Key: sN9cRrP/n9NdMgdcy2VJFQ==
Sec-WebSocket-Version: 13
```

服务器回应

```
HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: fFBooB7FAkLlXgRSz0BT3v4hq5s=
Sec-WebSocket-Origin: null
Sec-WebSocket-Location: ws://example.com/
```

Websocket

1. Server 返回 Sec-WebSocket-Accept 这个头，头内容是通过一定的算法生成的：

```
mask   = "258EAFA5-E914-47DA-95CA-C5AB0DC85B11";  // 这是算法中要用到的
accept = base64( sha1( key + mask ) );
```

2. key 和 mask 串接之后经过 SHA-1 处理，处理后的数据再经过一次 Base64 加密

1). t = "GhlIHNhbXBsZSBub25jZQ==" + "258EAFA5-E914-47DA-95CA-C5AB0DC
-> "GhlIHNhbXBsZSBub25jZQ==258EAFA5-E914-47DA-95CA-C5AB0DC85B11

2). s = sha1(t)
-> 0xb3 0x7a 0x4f 0x2c 0xc0 0x62 0x4f 0x16 0x90 0xf6
0x46 0x06 0xcf 0x38 0x59 0x45 0xb2 0xbe 0xc4 0xea

3). base64(s)
-> "s3pPLMBiTxaQ9kYGzzhZRbK+xOo="

Websocket

◆ 客户端(浏览器)发起请求示例：

```
var ws = new WebSocket("ws://localhost:8001");
ws.onopen = function(){
    console.log("Handshake success");
};
ws.onerror = function(){
    console.log("Got error");
};
```

```
Request Method: GET
Status Code: 101 Switching Protocols
▼ Request Headers    view source
  Accept-Encoding: gzip, deflate, sdch
  Accept-Language: en-US,en;q=0.8
  Cache-Control: no-cache
  Connection: Upgrade
  Cookie: BAIDUID=[redacted]d4c7c906a933623c7db2
  _lpvt_[redacted]
  Host: labs.music.baidu.com:8899
  Origin: http://labs.music.baidu.com
  Pragma: no-cache
  Sec-WebSocket-Extensions: permessage-deflate; client_max_window_bits
  Sec-WebSocket-Key: KY2A94nNxZrdWvCgFZF25A==
  Sec-WebSocket-Version: 13
  Upgrade: websocket
  User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0
  7.36
▼ Response Headers    view source
  Connection: Upgrade
  Sec-WebSocket-Accept: F8BA1r7ZKJBxsCzbcyECHyrP9uY=
  Upgrade: websocket
```

Websocket

◆ 服务端接收握手请求示例(基于NodeJS)：

```
var crypto = require('crypto');
require('net').createServer(function(o){
    var key;
    o.on('data',function(e){
        if(!key){
            console.log(e.toString()); // 握手
        }
    });
}).listen(8001);
```

```
GET / HTTP/1.1
Upgrade: websocket
Connection: Upgrade
Host: 127.0.0.1:8000
Origin: null
Pragma: no-cache
Cache-Control: no-cache
Sec-WebSocket-Key: /gfKkKCKmAfg8zt3+bALag==
Sec-WebSocket-Version: 13
Sec-WebSocket-Extensions: x-webkit-deflate-frame
User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36
Chrome/31.0.1650.63 Safari/537.36
```

Websocket

◆ 服务端回应握手示例：

```
var crypto = require('crypto');
var code= '258EAFA5-E914-47DA-95CA-C5AB0DC85B11';
require('net').createServer(function(o){
    var key;
    o.on('data',function(e){
        if(!key){ // 握手
            key = e.toString().match(/Sec-WebSocket-Key: (.+)/)[1];
            key = crypto.createHash('sha1').update(key + code).digest('base64');
            o.write('HTTP/1.1 101 Switching Protocols\r\n');
            o.write('Upgrade: websocket\r\n');
            o.write('Connection: Upgrade\r\n');
            o.write('Sec-WebSocket-Accept: ' + key + '\r\n'); o.write('\r\n');
        }else{
            console.log(e);
        };
    });
}).listen(8001);
```

Websocket

◆ 握手成功后的数据传输：

```
<Buffer 81 8c 2b 5d 5b fe cd d2 fa 18 a2 d6 bd 76 bb b8 d1 61>
<Buffer 8a 86 c0 6a b4 e9 26 ff 04 0f 4d c4>
```

◆ 数据帧格式：

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-------+-+-------------+-------------------------------+
|F|R|R|R| opcode|M| Payload len |    Extended payload length    |
|I|S|S|S|  (4)  |A|     (7)     |             (16/64)           |
|N|V|V|V|       |S|             |   (if payload len==126/127)   |
| |1|2|3|       |K|             |                               |
+-+-+-+-+-------+-+-------------+ - - - - - - - - - - - - - - - +
|     Extended payload length continued, if payload len == 127  |
+ - - - - - - - - - - - - - - - +-------------------------------+
|                               |Masking-key, if MASK set to 1  |
+-------------------------------+-------------------------------+
| Masking-key (continued)       |          Payload Data         |
+-------------------------------- - - - - - - - - - - - - - - - +
:                     Payload Data continued ...                :
+ - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - +
|                     Payload Data continued ...                |
+---------------------------------------------------------------+
```

```
FIN            1bit 表示信息的最后一帧，flag，也就是标记符
RSV 1-3        1bit each 以后备用的 默认都为 0
Opcode         4bit 帧类型，
Mask           1bit 掩码，是否加密数据
Payload        7bit 数据的长度
Masking-key        1 or 4 bit 掩码
Payload data       (x + y) bytes 数据
Extension data     x bytes  扩展数据
Application data y bytes  程序数据
```

Websocket

- 环境依赖

  - Client

    实现websocket的协议，浏览器扮演着一个很重要的角色。所有最新的浏览器，除了Android的浏览器支持的最新规范（RFC 6455）的WebSocket协议

  - Server

    在服务器方面，网上都有不同对websocket支持的服务器：
    php - http://code.google.com/p/phpwebsocket/
    node.js - http://socket.io

Websocket

- 优势：
  - Header头，服务器与客户端之间交换的封包檔头很小，大概只有2字节
  - 服务器推送，服务器可以主动传送数据给客户端

- Live Demo
  - 一个完整的基于WebSocket的推送接收Client&Server端实现

Baidu 百度

Websocket

◆ 简易聊天室示例（基于Nodejs）：

手机扫码进入

http://labs.music.baidu.com/wave/room/1008

Websocket

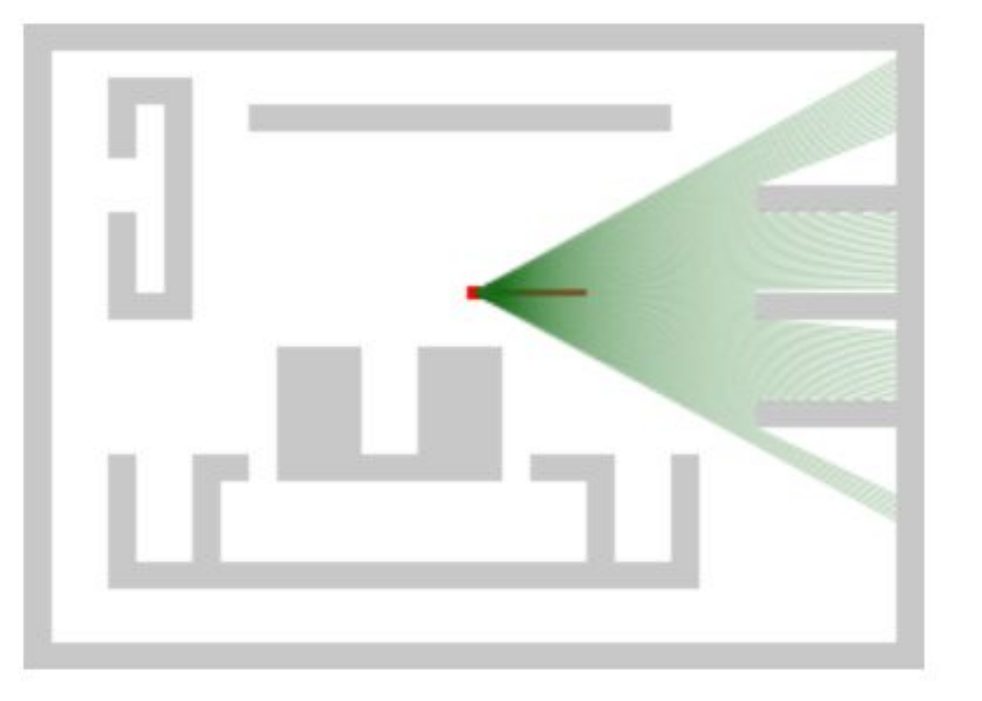
- Reference
  - Wiki: https://zh.wikipedia.org/zh/WebSocket
  - Chat socket io：
  https://github.com/Automattic/socket.io/tree/master/examples/chat
  - Wave：http://labs.music.baidu.com/wave/room/1008

Have fun

进阶应用

进阶应用

◆ 综合上述HTML5 Canvas+Websocket+Ajax能做什么？



http://labs.music.baidu.com/demo/raycast/

Thanks

骆勤

@enimo

Baidu 百度