

# Lecture3

# HTML5基础





# 目录



什么是HTML5?



HTML5特征



HTML5工具



# 什么是 HTML5?

- ❄ HTML5 是下一代 HTML 标准。
- ❄ HTML , HTML 4.01的上一个版本诞生于 1999 年。自从那以后，Web 世界已经经历了巨变。
- ❄ HTML5 是 W3C 与 WHATWG 合作的结果,WHATWG 指 Web Hypertext Application Technology Working Group。
- ❄ HTML5 是专门为承载丰富的 web 内容而设计的，并且无需额外插件。
- ❄ HTML5 拥有新的语义、图形以及多媒体元素。
- ❄ HTML5 提供的新元素和新的 API 简化了 web 应用程序的搭建。
- ❄ HTML5 是跨平台的，被设计为在不同类型的硬件（PC、平板、手机、电视机等等）之上运行

# 为何HTML5?

❄ 旧路新铺

❄ 简单至上

❄ 国际化支持

\* 跨文化和语言

\* WAI – ARIA, web可访问性倡议–可访问的富internet 应用

❄ 原生替代插件

❄ 安全性

# 旧路新铺

HTML4

<form>

<input name="email" type="text">



crapload of  
validation code  
or extra js lib

(?:[a-zA-Z!#\$%&'\*+/=?^`{|}~-]+(?:\.[a-zA-Z!#\$%&'\*+/=?^`{|}~-]+)\*|"(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21\x23-\x5b\x5d-\x7f]|\\"[\x01-\x09\x0b\x0c\x0e-\x7f])\*")@(?:(:[a-zA-Z-](?:[a-zA-Z-]\*[a-zA-Z-])?\.|[a-zA-Z-]\*[a-zA-Z-]?|\\[(?:(:25[0-5]|2[0-4][0-9]|01)?[0-9][0-9]?){3}|(?:25[0-5]|2[0-4][0-9]|01)?[0-9][0-9]?|[a-zA-Z-]\*[a-zA-Z-]:(:[\x01-\x08\x0b\x0c\x0e-\x1f\x21-\x5a\x53-\x7f]|\\"[\x01-\x09\x0b\x0c\x0e-\x7f])+)\])



HTML5

<input type=email required>

# 简单至上

## HTML4/XHTML

- HTML 4.01 Strict
- HTML 4.01 Transitional
- HTML 4.01 Frameset
- XHTML 1.0 Strict
- XHTML 1.0 Transitional
- XHTML 1.0 Frameset
- XHTML 1.1



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01  
Transitional//EN"
```

"<http://www.w3.org/TR/html4/loose.dtd>">



HTML5

```
<!DOCTYPE html>
```

# 简化的字符集

HTML4

```
<meta http-equiv="Content-Type"  
content="text/html; charset=utf-8">
```



HTML5

```
<meta charset=utf-8>
```

# 简化的Markup

HTML5

```
<!DOCTYPE html>  
<meta charset=utf-8>  
<title>HTML5</title>  
<h1>HTML5!</h1>
```

70 characters!

DOM

```
<!DOCTYPE html>  
└<html>  
  └<head>  
    <meta charset="utf-8">  
    <title>HTML5</title>  
  </head>  
  └<body>  
    <h1>HTML5!</h1>  
  </body>  
</html>
```

# Universal Access

## \* Support for all world languages

- \* For example, the new <ruby> element

## \* Accessibility

- \* Semantic markup
- \* WebVTT



```
<video controls  
      preload="metadata">  
  <source src="rocpoc.mp4">  
  <track label="English"  
        kind="subtitles"  
        src="subtitles_en.vtt">  
</video>
```

WEBVTT

1  
00:00:01,000 --> 00:00:03,050  
What do I think about HTML 5?...

2  
00:00:04,000 --> 00:00:07,100  
It's pretty sweet. I just hope that  
people use the new features  
thoughtfully.

[http://html5videoguide.net/presentations/  
WebVTT](http://html5videoguide.net/presentations/WebVTT)

# 无插件范式

❄ 之前很多功能通过插件或者复杂的hack（本地绘图api, 本地socket等）来实现，现在，HTML5提供原生支持。



# Plugins are on the way out

- ❄ Built-in is better than bolt-on —@brucel
- ❄ Plugins may not be installed (for long)
- ❄ Plugins can be an attack vector (Proxy poisoning)
- ❄ “Whenever a Mac crashes, more often than not is because of Flash” —Steve Jobs

[Home](#) > [Industry News](#) > Adobe kills mobile Flash, blames iPhone and iPad, “Occupy Flash” rises

## Adobe kills mobile Flash, blames iPhone and iPad, “Occupy Flash” rises

November 14, 2011 by Bill Palmer

by Bill Palmer

### Data Storage Solutions

Best-in-class data storage products for AV, Enterprise and Home Users.  
[www.promarktech.com](http://www.promarktech.com)

AdChoices ▾



Adobe has discontinued its much maligned Mobile Flash Player software for smartphones and tablets, after years of bizarrely pushing forward with the the 1990s-era web development tool still had a place in the twenty-first century, long after the internet had passed it by. But just when it was starting to look like Adobe finally figured out what everyone else had known about Flash for centuries, the company had to go and crush that notion by stating that it only killed off Flash because Apple's refusal to allow the buggy and resource-hogging plugin onto its iPhone and iPad. In other words, Adobe never did figure out that

### News

[My Bookmarks](#)

[Contribute News](#)

[print](#)

## Microsoft has Abandoned Silverlight and All Other Plugins in Metro IE

Posted by Jonathan Allen on Sep 15, 2011

Sections Development Topics [Silverlight](#) , [.NET](#) , [Javascript](#) , [HTML5](#) , [Web 2.0](#) Tags [Windows 8](#) , [Flash](#)

Share [+1](#) | [Facebook](#) [Twitter](#) [Digg](#) [StumbleUpon](#) [Email](#)

[Bookmark this!](#)

Though it has been hard, we have been trying to avoid reporting on rumors about the death of Silverlight for quite some time. As in all things, rumors tend to be exaggerated or out-right false. A good example of this is the idea that Web Forms has entered maintenance mode and would never be updated to HTML5. Unfortunately the end of Silverlight is no rumor; if Microsoft doesn't change course it, as well as Flash and other plugin technologies, will be effectively unusable when Windows 8 is released.

<http://goo.gl/mzFjM>

<http://www.infoq.com/news/2011/09/Metro-Plug-ins>

# HTML5 is Secure by Default

- ❄ HTML5 uses origin-based security
- ❄ Defines secure cross-origin sharing
- ❄ Issues in HTML5 are fixed quickly



enable cross-origin resource sharing

**What is this about?**

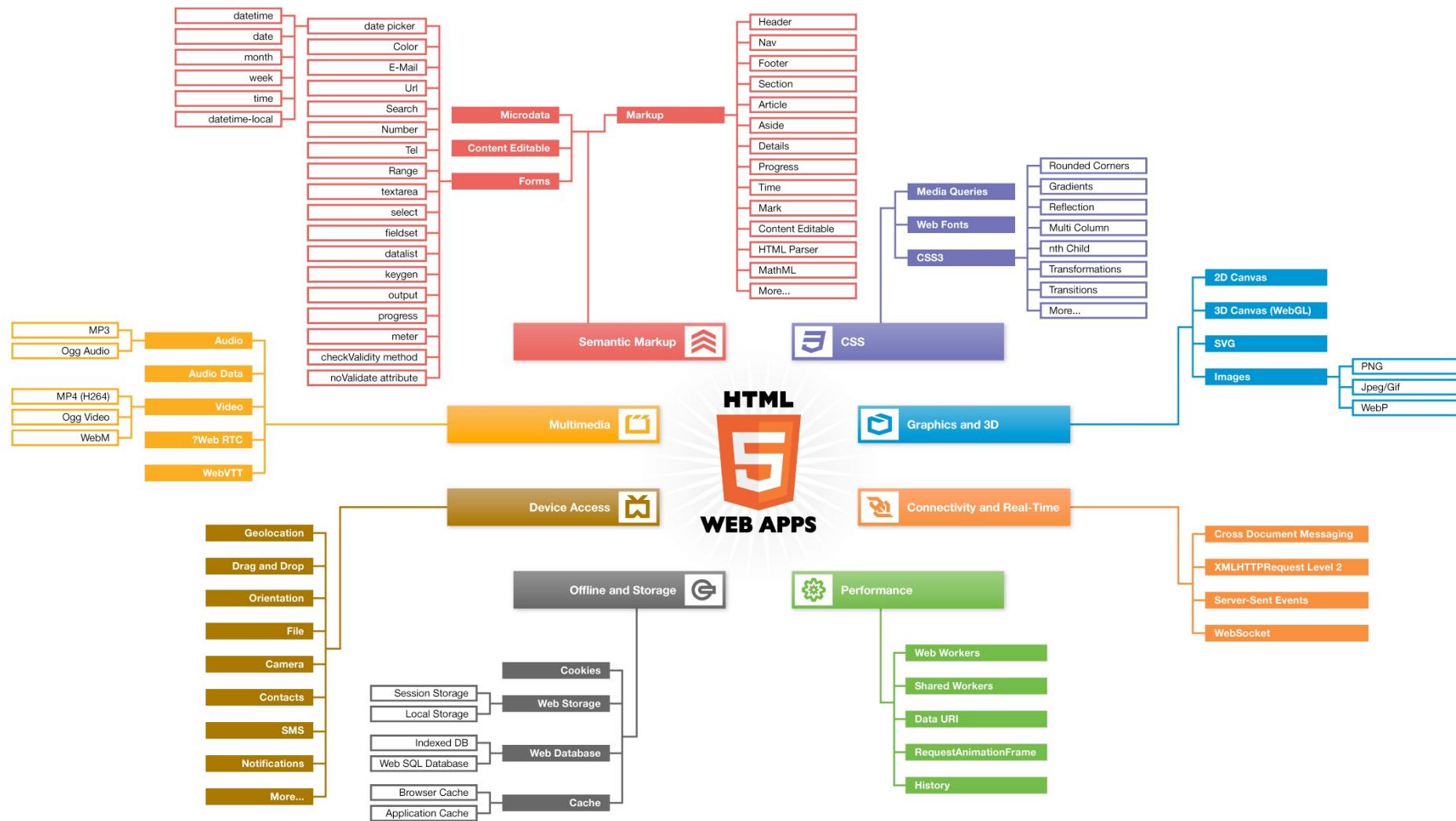
Cross-Origin Resource Sharing (CORS) is a [specification](#) that enables a truly open access across domain-boundaries. With this site we want to support the adoption of CORS. [\[more...\]](#)

If you have **public content** that doesn't use require cookie or session based authentication to see, then please consider opening it up for universal JavaScript/browser access. [\[more...\]](#)

**Why is CORS important?**

<http://enable-cors.org/>

# HTML5 At a Glance



# HTML5 的改进

❄ 新元素

❄ 新属性

❄ 完全支持 CSS3

❄ Video 和 Audio

❄ 2D/3D 制图

❄ 本地存储

❄ 本地 SQL 数据

❄ Web 应用

# New HTML5 Elements

Layout (semantics)		Multimedia	Other
<b>article</b>	<b>section</b>	<b>audio, video</b>	<b>command, menu*</b>
<b>aside</b>	<b>nav</b>	<b>source</b>	<b>embed</b>
<b>figure</b>	<b>figcaption</b>		
<b>header</b>	<b>footer</b>	<b>Graphics</b>	<b>Forms</b>
<b>details</b>	<b>summary</b>	<b>canvas</b>	<b>datalist</b>
<b>wbr</b>	<b>hgroup</b>	<b>svg</b>	<b>progress</b>
<b>time</b>	<b>mark</b>		<b>output</b>
<b>meter</b>	<b>ruby, rp, rt</b>		<b>keygen</b>

\* menu redefined from HTML 4

[dev.w3.org/html5/markup/elements.html](http://dev.w3.org/html5/markup/elements.html)

# Obsolete Elements

能使用CSS替代的元素

- \* basefont, center, font, tt, u

不再使用frame框架

只有部分浏览器支持的元素

- \* applet, bgsound, marquee

其他被废除的元素

- \* isindex, dir, rb等

# 语义化

✿ 根据结构化的内容选择合适的标签

✿ why

- \* 有利于SEO
- \* 开发维护体验好
- \* 用户体验更好
- \* 更好的可访问性，方便任何设备对代码进行解析

# 新的语义/结构元素

<header>

<nav>

<article>

<section>

<aside>

<footer>

# Microdata

- ❄ 微数据是WHATWG HTML标准的一部分，用于在网页上的现有内容中嵌套元数据。
- ❄ 搜索引擎，网络抓取工具和浏览器可以从网页中提取和处理微数据，并使用它为用户提供更丰富的浏览体验。
- ❄ 高级的微数据由一组name-value对组成



# 词汇表

- ❖ 谷歌和其他主要搜索引擎支持结构化数据的 Schema.org词汇表。谷歌，微软和雅虎等主要搜索引擎运营商依靠schema.org 词汇表来改进搜索结果。
- ❖ 微数据词汇表提供了项目的语义或含义
- ❖ 此词汇表定义了一组标准类型名称和属性名称
  - \* 例如，Schema.org Music Event表示音乐会表演，startDate和location 属性用于指定音乐会的关键细节。在这种情况下，Schema.org Music Event将是itemtype和startDate使用的URL，而location将是Schema.org Music Event 定义的itemprop。

# HTML5 Microdata Example

HTML

```
<section itemscope>
  <article id="html5-fast-track"
            itemtype="http://data-vocabulary.org/Product">
    <header>
      <h1 itemprop="name">HTML5 Fast Track</h1>
    </header>
    <p itemprop="description">The HTML5 Fast Track
       training course is a two day experience...
    </p>
  </article>
</section>
```

# Microdata Testing

The screenshot shows the Google Developers website for search documentation. The main title is "Microdata Testing". Below it, there's a section titled "测试结构化数据" (Testing Structured Data). A note from Google suggests performing a "富媒体搜索结果测试" (Rich Result Test) to see what Google can extract from the page. There are two main tools demonstrated: "Rich Results Test" and "Schema.org". The "Rich Results Test" tool shows the original HTML code and the detected structured data items. The "Schema.org" tool shows the JSON-LD code and its validation results.

Google 建议您先进行富媒体搜索结果测试，了解系统可以为您的网页生成哪些 Google 富媒体搜索结果。对于常规架构验证，请使用架构标记验证器 测试所有类型的 schema.org 标记，而无需进行 Google 特有的验证。

This screenshot shows the "Rich Results Test" tool interface. On the left, the original HTML code is displayed. In the center, the "Test results" section shows a green checkmark indicating "4 valid items detected". It also includes links to "VIEW TESTED PAGE" and "PREVIEW RESULTS". Below this, a "Detected structured data" section lists four categories: "Guided recipes", "Recipes", "Review snippets", and "Videos", each with a green checkmark and a "1 valid item detected" message. The entire interface has a clean, modern design with a dark header and light body.

```
<html>
<head>
<title>Party Coffee Cake</title>
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "Recipe",
  "name": "Party Coffee Cake",
  "image": [
    "https://example.com/photos/1x1/photo.jpg",
    "https://example.com/photos/4x3/photo.jpg",
    "https://example.com/photos/16x9/photo.jpg"
  ],
  "author": {
    "@type": "Person",
    "name": "Mary Stone"
  },
  "datePublished": "2018-03-10",
  "description": "This coffee cake is awesome and perfect for parties."
  "prepTime": "PT20M",
  "cookTime": "PT30M",
  "totalTime": "PT50M",
  "keywords": "cake for a party, coffee",
  "recipeYield": "10",
  "recipeCategory": "Dessert",
  "recipeCuisine": "American",
  "nutritionValue"
}
```

## 富媒体搜索结果测试

这是一款测试结构化数据的官方 Google 工具，可用于查看系统能根据您网页上的[结构化数据](#)生成哪些 Google 富媒体搜索结果。您还可以预览富媒体搜索结果在 Google 搜索结果中的显示效果。

[转到富媒体搜索结果测试](#)

This screenshot shows the "Schema.org" validation tool. At the top, there's a navigation bar with "Documentation", "Schemas", and "About". The main area displays a JSON-LD code snippet for a "Recipe". Below the code, a "Validation" section shows a green checkmark and the text "0 ERRORS 0 WARNINGS". The right side of the interface is a detailed view of the "Recipe" schema with various properties listed: @type, name, image, author, datePublished, description, prepTime, cookTime, totalTime, keywords, and nutrition. Each property is shown with its value and a small green checkmark. The overall layout is professional and user-friendly, designed for developers to quickly assess their schema implementation.

```
<html>
<head>
<title>Party Coffee Cake</title>
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "Recipe",
  "name": "Party Coffee Cake",
  "image": [
    "https://example.com/photos/1x1/photo.jpg",
    "https://example.com/photos/4x3/photo.jpg",
    "https://example.com/photos/16x9/photo.jpg"
  ],
  "author": {
    "@type": "Person",
    "name": "Mary Stone"
  },
  "datePublished": "2018-03-10",
  "description": "This coffee cake is awesome and perfect for parties."
  "prepTime": "PT20M",
  "cookTime": "PT30M",
  "totalTime": "PT50M",
  "keywords": "cake for a party, coffee",
  "recipeYield": "10",
  "recipeCategory": "Dessert",
  "recipeCuisine": "American",
  "nutrition": {
    "@type": "NutritionInformation",
    "calories": "270 calories"
  }
}
```

## 架构标记验证器

验证网页中嵌入的所有基于 Schema.org 的结构化数据，不过系统不会显示特定于 Google 功能的警告。

[转到架构标记验证器](#)

# HTML5 表单

## 新的表单功能:

- \* 无需JavaScript
- \* 原生的输入类型: color, date, email, number, tel 等等
- \* 新的输入属性: autocomplete, autofocus, min, max等

Features degrade gracefully 旧浏览器中新的表单控件会平滑降级(unknown input types are treated as text)

# 新的输入类型实例

Runner:@peterlubbers

Tel #: 1-877-KAAZING

E-mail: peter

DOB: 1944-06-06  
Please enter a valid email address

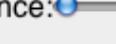
T-shirt Size:

- Small
- Medium
- Large

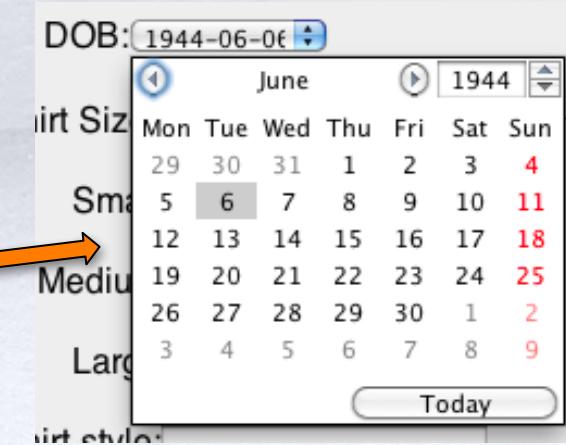
Shirt style: \_\_\_\_\_

Shirt Color: 

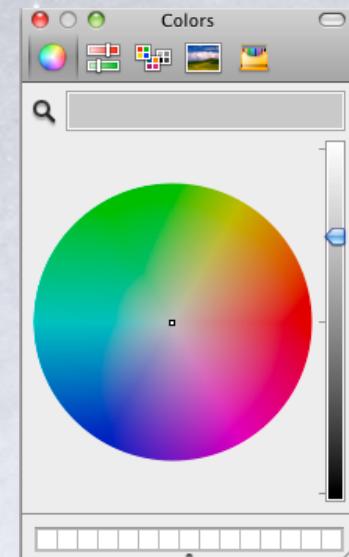
Expectations:

Confidence:  0%

Notes: \_\_\_\_\_



Date picker



Color picker

# HTML5 媒体

❄ 新的 HTML5媒体元素

- \* <audio> , <video>

❄ 原生支持

- \* JavaScript可编程
- \* 借助CSS实现样式

❄ 支持多种格式



# HTML5 视频示例

## HTML

```
//Basic Video
<video controls src="goldrush.mp4">
    A movie about HTML5
</video>

// Video with additional attributes
<video id="movies"
    controls preload="metadata"
    width="400px" height="300px"
    poster="html5.png" >
    <source src="goldrush.webm">
    <source src="goldrush.mp4">
    <track label="English" kind="subtitles"
        srclang="en" src="subtitles_en.vtt">
    A movie by Rocky Lubbers
</video>
```

# HTML5图形

❄ Canvas 和 SVG

❄ 原生绘图功能

- \* 区别于插件plugins (Flash, Silverlight)

❄ 完全整合于HTML5 documents (DOM的一部分)

- \* 由CSS提供样式

- \* 由 JavaScript来绘制

❄ 用于动画、图表、图像、像素操作等

# SVG

- ❖ SVG 指可伸缩矢量图形 (Scalable Vector Graphics)
- ❖ SVG 用于定义用于网络的基于矢量的图形
- ❖ SVG 使用 XML 格式定义图形
- ❖ SVG 图像在放大或改变尺寸的情况下其图形质量不会有损失
- ❖ SVG 是万维网联盟的标准

# SVG 的优势

与其他图像格式相比（比如 JPEG 和 GIF），使用 SVG 的优势在于：

- \* SVG 图像可通过文本编辑器来创建和修改
- \* SVG 图像可被搜索、索引、脚本化或压缩
- \* SVG 是可伸缩的
- \* SVG 图像可在任何的分辨率下被高质量地打印
- \* SVG 可在图像质量不下降的情况下被放大

# HTML5 Canvas

- HTML5 的 canvas 元素使用 JavaScript 在网页上绘制图像。
- 画布是一个矩形区域，您可以控制其每一像素。
- canvas 拥有多种绘制路径、矩形、圆形、字符以及添加图像的方法。

# HTML 5 Canvas vs. SVG

## SVG

- \* SVG 是一种使用 XML 描述 2D 图形的语言。
- \* SVG 基于 XML，这意味着 SVG DOM 中的每个元素都是可用的。您可以为某个元素附加 JavaScript 事件处理器。
- \* 在 SVG 中，每个被绘制的图形均被视为对象。如果 SVG 对象的属性发生变化，那么浏览器能够自动重现图形。

## Canvas

- \* Canvas 通过 JavaScript 来绘制 2D 图形。
- \* Canvas 是逐像素进行渲染的。
- \* 在 canvas 中，一旦图形被绘制完成，它就不会继续得到浏览器的关注。如果其位置发生变化，那么整个场景也需要重新绘制，包括任何或许已被图形覆盖的对象。

# Canvas 与 SVG 的比较

Canvas	SVG
依赖分辨率	不依赖分辨率
不支持事件处理器	支持事件处理器
弱的文本渲染能力	最适合带有大型渲染区域的应用程序 (比如谷歌地图)
能够以 .png 或 .jpg 格式保存结果图像	复杂度高会减慢渲染速度 (任何过度使用 DOM 的应用都不快)
最适合图像密集型的游戏，其中的许多对象会被频繁重绘	不适合游戏应用

# HTML5 Web 存储

❄ HTML5 web存储,一个比cookie更好的本地存储方式。

❄ 客户端存储数据的两个对象为：

- \* **localStorage** – 用于长久保存整个网站的数据，保存的数据没有过期时间，直到手动去除。
- \* **sessionStorage** – 用于临时保存同一窗口(或标签页)的数据，在关闭窗口或标签页之后将会删除这些数据。

# HTML5 应用程序缓存

- ✿ 使用 HTML5，通过创建 cache manifest 文件，可以轻松地创建 web 应用的离线版本。
- ✿ HTML5 引入了应用程序缓存，这意味着 web 应用可进行缓存，并可在没有因特网连接时进行访问。
- ✿ 应用程序缓存为应用带来三个优势：
  - \* 离线浏览 – 用户可在应用离线时使用它们
  - \* 速度 – 已缓存资源加载得更快
  - \* 减少服务器负载 – 浏览器将只从服务器下载更新过或更改过的资源。
- ✿ manifest 文件是简单的文本文件，它告知浏览器被缓存的内容（以及不缓存的内容）。manifest 文件可分为三个部分：
  - \* CACHE MANIFEST – 在此标题下列出的文件将在首次下载后进行缓存
  - \* NETWORK – 在此标题下列出的文件需要与服务器的连接，且不会被缓存
  - \* Fallback – 在此标题下列出的文件规定当页面无法访问时的回退页面（比如 404 页面）

# 更新缓存

一旦应用被缓存， 它就会保持缓存直到发生下列情况：

- \* 用户清空浏览器缓存
- \* manifest 文件被修改
- \* 由程序来更新应用缓存

```
CACHE MANIFEST  
# 2012-02-21 v1.0.0  
/theme.css  
/logo.gif  
/main.js
```

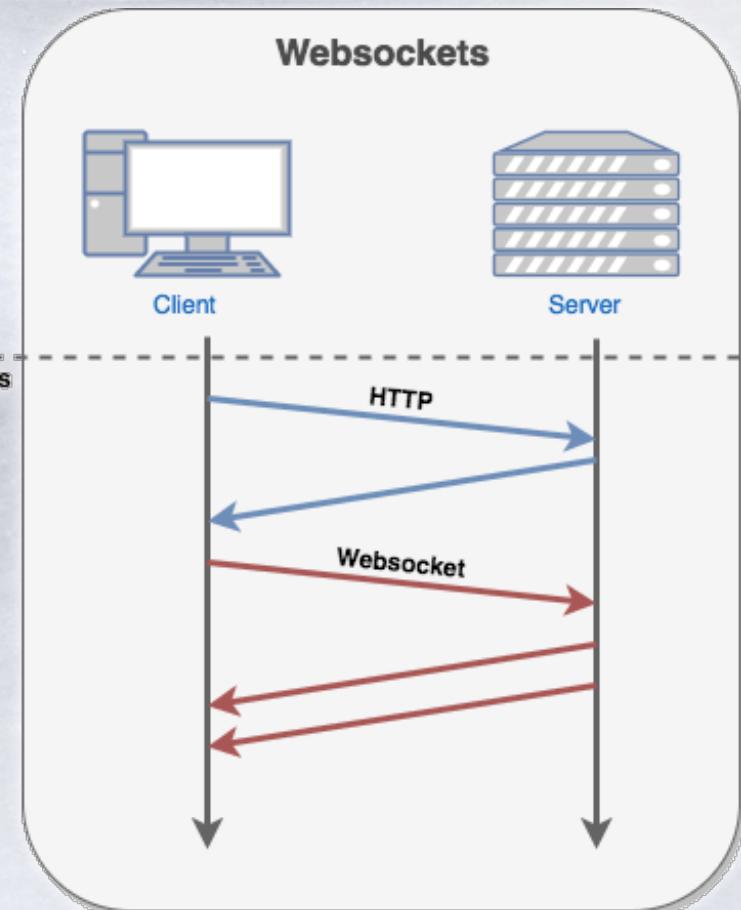
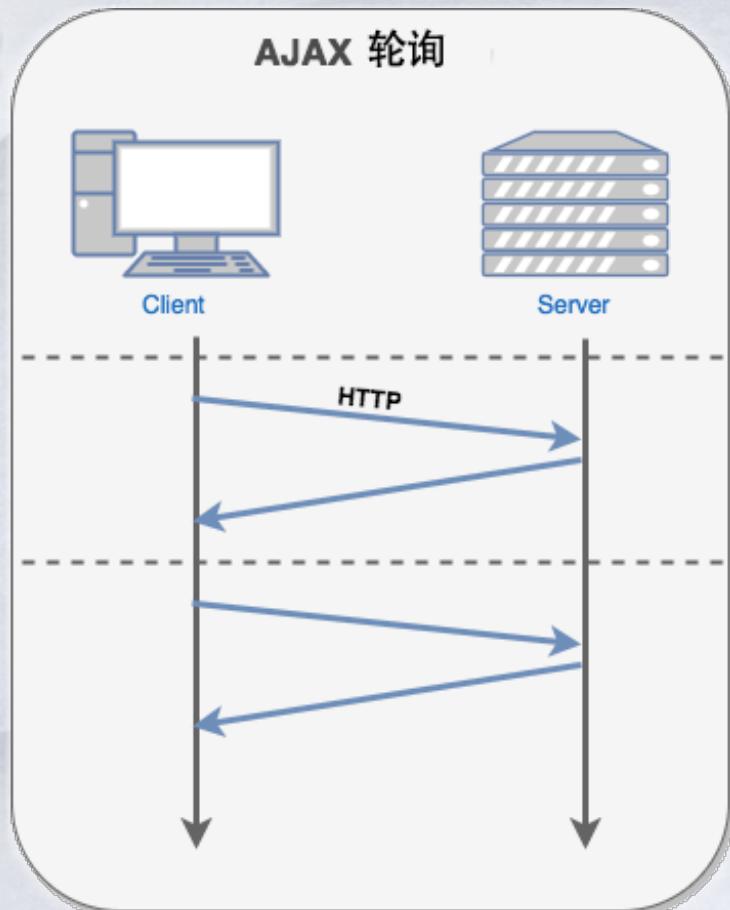
NETWORK:  
login.php

FALLBACK:  
/html/ /offline.html

# HTML5 Web Workers

- ❄ 当在 HTML 页面中执行脚本时，页面的状态是不可响应的，直到脚本已完成。
- ❄ web worker 是运行在后台的 JavaScript，独立于其他脚本，不会影响页面的性能。可以继续做任何愿意做的事情：点击、选取内容等等，而此时 web worker 在后台运行。

# HTML5 WebSocket



# HTML5 地理定位

- ❖ 定位用户的位置
- ❖ HTML5 Geolocation API 用于获得用户的地理位置。
- ❖ 鉴于该特性可能侵犯用户的隐私，除非用户同意，否则用户位置信息是不可用的。

# HTML5 拖放

- ✿ 拖放（Drag 和 Drop）是很常见的特性。它指的是您抓取某物并拖入不同的位置。
- ✿ 拖放是 HTML5 标准的组成部分：任何元素都是可拖放的。

# Feature Detection

❄ Feature detect with Modernizr (part of H5BP):  
<http://www.modernizr.com>



# HTML5 Enabling JavaScript



使用 Sjoerd Visscher 创建的 "HTML5 Enabling JavaScript", " shiv" 来解决该问题:

```
<!--[if lt IE 9]>
<script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>
<![endif]-->
```

```
<!--[if lt IE 9]>
<script src="http://cdn.static.runoob.com/libs/html5shiv/3.7/
html5shiv.min.js"></script>
<![endif]
```

# Who is Developing HTML5?

\* Web Hypertext Application Technology Working Group (WHATWG)

- \* <http://www.whatwg.org/specs/web-apps/current-work/>

\* World Wide Web Consortium (W3C)

- \* <http://dev.w3.org/html5/spec/Overview.html>

\* Internet Engineering Task Force (IETF)

- \* <http://tools.ietf.org/html/draft-hixie-thewebsocketprotocol>



# References

- ❖ Pro HTML5 Programming (Apress, 2011)
- ❖ <https://www.w3schools.com/>



# Thanks!!!

