

Practical HTML5



柴春燕

2015-08-15

关于我

2014-Now

金山云 前端架构师

2012-2014

百度 高级研发工程师

2010-2012

当当网 无线开发工程师

HTML





1994年
Netscape
Opera 1.x

HTML 1.0

1995年
Netscape
Opera 1.x
IE 1.x

CSS酝酿中...

1999年
Netscape
Opera 2.x
IE 5.x

HTML3.0, 4.0
CSS 2.1

2001年
IE 6.x
Opera 6.x
Mozilla 0.7

XHTML 1.0

2004年
IE 6.0 SP2
Opera 7.5
Mozilla Firefox
1.0
Safari 1.0

XHTML 1.1

20010年
IE 8, IE9
Opera 10.X
Mozilla Firefox 3.6X
Safari 4.x
Chrome 4x

HTML 5(2007)
CSS3

Hello HTML5



New version



No need for
plugins



Cross platform



Designed by W3C
and WHATWG

HTML5 特性



HTML5 更多特性

- Web SQL database
- Indexed Database API
- File API
- File Writer
- Background workers
- Geolocation
- Web sockets
- ...

HTML5 Video



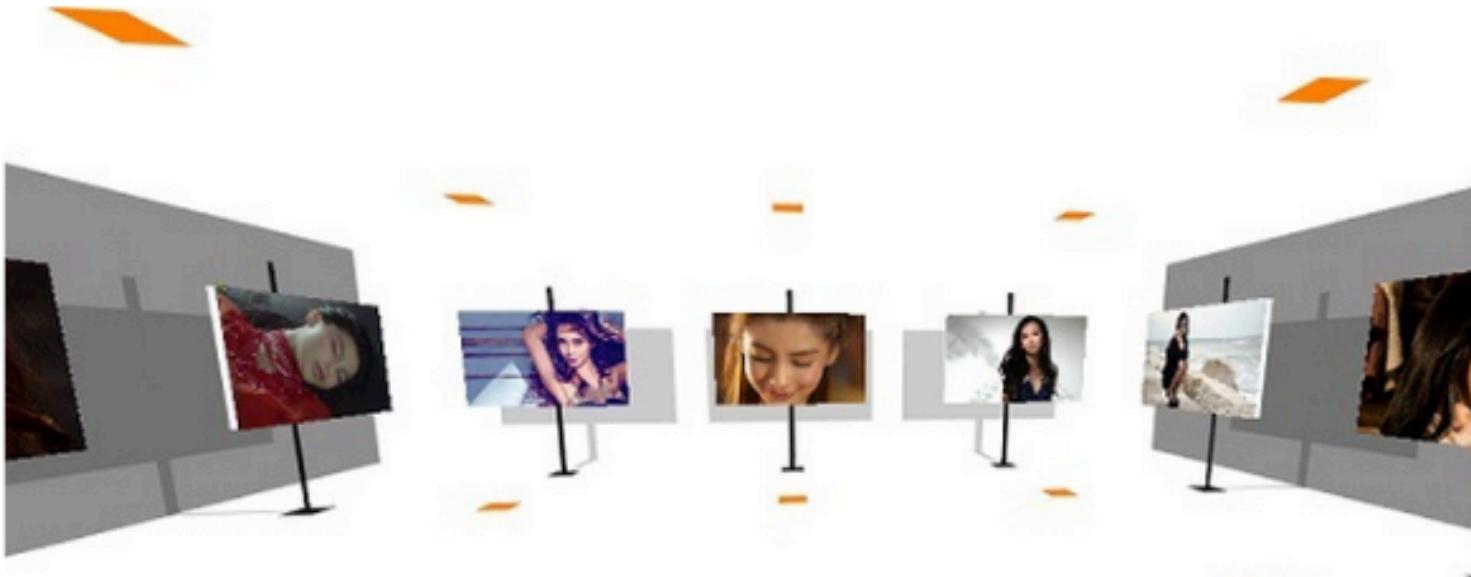
Fullscreen HTML5 Page Background Video

HTML5 Canvas



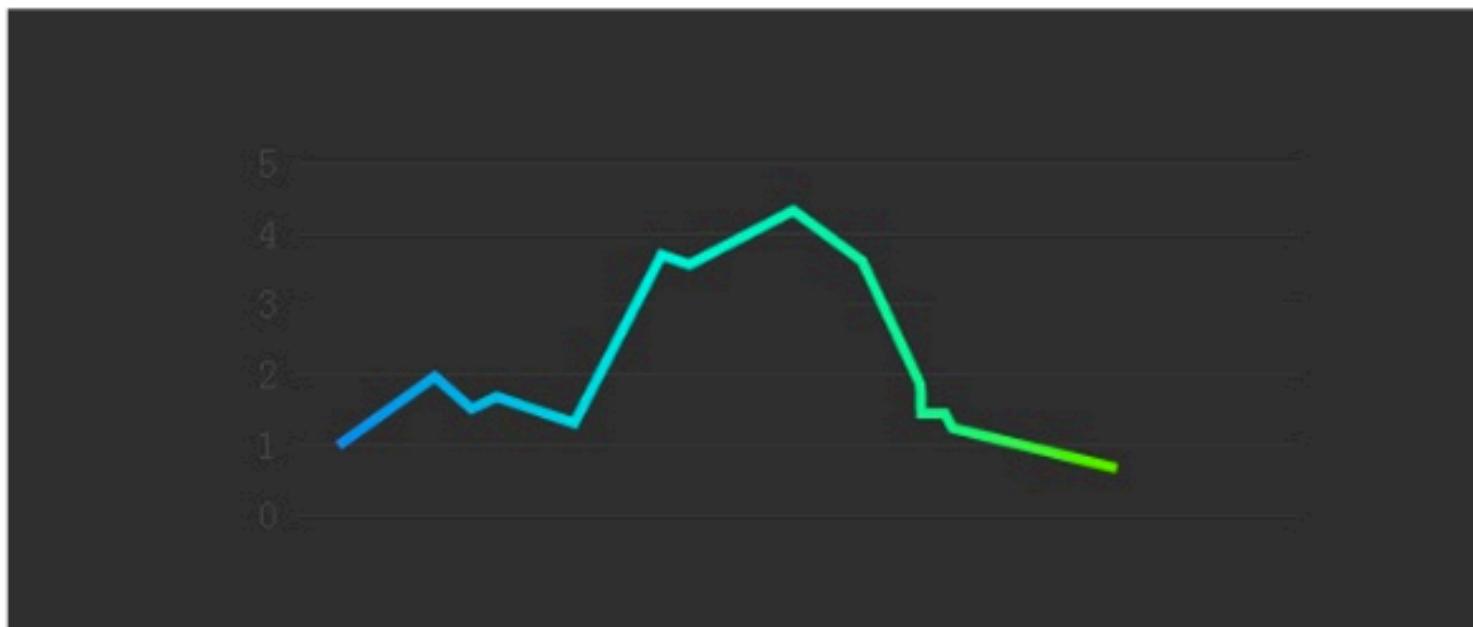
HTML5 3D 旋转地球

HTML5 CSS3



3D立体图片相册

HTML5 SVG



基于HTML5的SVG动画折线图表 线颜色渐变

HTML5 Game



HTML5版切水果游戏

And More ...

是不是有些心动
Let's Begin!

HTML5 TAGS FORM

Created by [ChaiChunyan](#) / [@chaichunyan](#)

HTML5 TAGS

UPDATING TO HTML5

IN HTML4:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
      "http://www.w3.org/TR/html4/strict.dtd">
```

IN HTML5: 更简洁的代码

```
<!DOCTYPE html>  
<html>  
  
  <meta charset="UTF-8">  
  <title>HTML5 Training</title>  
  
  ...  
  
</html>
```

CSS JS 引用

CSS

```
<link href="demo.css">
```

JAVASCRIPT

```
<script src="demo.js"></script>
```

HTML5 语义化标签

- header
- footer
- section
- nav
- article
- aside

Describing the structure of a web page in HTML 4

<div id="header"> *This div element contains branding like the logo*

<div id="nav"> *This div element contains the site navigation*

<div id="content">

This div element contains the web page's main content

<div id="sidebar">

This div element contains extra information and related content/links

<div id="footer"> *This div element contains copyright information*

<header>

<nav>

<article>

<section>

<aside>

<footer>

DON'T NEED DIV

```
<div id="header"></div>
<div class="section"></div>
<div class="article"></div>
<div id="footer"></div>
```

SECTION

```
<article>
  <hgroup>
    <h1>Apples</h1>
    <h2>Tasty, delicious fruit!</h2>
  </hgroup>
  <p>The apple is the pomaceous fruit of the apple tree.</p>
  <section class="future" aria-hidden="true" style="top: 330px; dis
    <h1>Red Delicious</h1>
    <p>These bright red apples are the most common found in many
  </section>
  <section class="future" aria-hidden="true" style="top: 330px; dis
    <h1>Granny Smith</h1>
    <p>These juicy, green apples make a great filling for apple p
  </section>
</article>
```

OTHER EXAMPLE

```
<output name="result"></output>

<progress id="p" max="100"><span>0</span>%</progress>

<meter value="6" max="8">6 blocks used (out of 8 total) </meter>

Voter turnout: <meter value="0.75"></meter>

Tickets sold: <meter min="0" max="100" value="75"></meter>
```

DETAILS AND SUMMARY

```
<details>
  <summary>
    Copying... <progress max="375505392" value="97543282"></progress>
  </summary>
  <dl>
    <dt>Transfer rate:</dt> <dd>452KB/s</dd>
    <dt>Local filename:</dt> <dd>/home/rpausch/raycd.m4v</dd>
  </dl>
</details>
```

ADDRESS

```
<address>
  <a href="/People/Raggett/">Dave Raggett</a>,
  <a href="/People/Arnaud/">Arnaud Le Hors</a>,
  contacts for the w3c HTML activity
</address>
```

Address 应用于最近的article或body标签

DATA 属性

```
<div class="spaceship" data-ship-id="92432" data--="" weapons="laser 2
<script>
div = document.getElementsByTagName('div')[0]; spaceships[div.dataset
</script>
```

FIGURE 和 FIGURE 标题

```
<figure>
  Bubbles at work</figcaption>
</figure>
```

TIME和发布日期属性

```
<time pubdate="" datetime="2009-10-09T14:28- 08:00"></time>
```

菜单(MENU)和命令(COMMAND)

```
<menu type="toolbar">
    <command type="radio" radiogroup="alignment" checked="checked" la=
    <command type="radio" radiogroup="alignment" label="Center" icon=
    <command type="radio" radiogroup="alignment" label="Right" icon=
</menu>
```

MARK用户高亮的东西

```
<p>
    <mark>Momentum is preserved across the wormhole. Electromagnetic
</p>
```

STRONG用来表示重要

```
<p>When a wormhole is created, a vortex normally forms. <strong>Warning!</strong> If you pass through a wormhole, you will be pulled into the vortex and never come out. This is because the wormhole connects two points in space-time, and the vortex is a disturbance in the fabric of space-time that pulls objects towards it.
```

需要注意的是

语义化标签是按需使用的

并不是所有的导航都要用

`<nav></nav>`只表示主导航，例如底部的nav

`Article`是页面的唯一内容

`<article></article>`

`Hgroup`最高级别的头部组

`<hgroup></hgroup>`

ITEMSCOPE 和 ITEMPROP

```
<div itemscope="">
  <p>My name is <span itemprop="name">Elizabeth</span>.</p>
</div>
<div itemscope="">
  <p>My name is <span itemprop="name">Daniel</span>.</p> </div>
<div itemscope="">
  <p>My name is <span itemprop="name">Neil</span>.</p> <p>My band is <span itemprop="band">The Red Elastic Band</span>.</p>
  <p>I am <span itemprop="nationality">British</span>.</p>
</div>
```

BUT HAVE ANY QUESTION ...

兼容性

```
<header><!--header--></header>
header{color: red;} IE6、7、8无法渲染。
```

HOW DO?

方法一:CODING JAVASCRIPT

```
<!--[if lt IE9]>
<script>
(function() {
    if (!
/*@cc_on!@*/
    0) return;
    var e = "abbr, article, aside, audio, canvas, datalist, details,
    var i= e.length;
    while (i--){
        document.createElement(e[i])
    }
})()
</script>
<![endif]-->
```

方法二:使用GOOGLE的HTML5SHIV包

```
<!--[if lt IE9]>
<script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></sc
<![endif]-->
```

HTML5 FORM

NEW HTML5 INPUT TYPES

								
		Firefox	Safari	Safari	Chrome	Opera	IE	Android
Email	?	4+	5+	3.1+	6+/10+	10.6+	10+	2.3-
Tel	?	4+	5+	3.1+	6+	10.6+	10+	2.3+
Url	?	4+	5+	3.1+	6+/10+	10.6+	10+	2.3-
Search	?	3.6/4+	5+	4+	6+	10.6+	9/10+	2.3-
Color	?	11-	5.2-	5-	20+	11+	10-	2.3-
Number	?	11-	4/5.2+	4+	9-/10+	11+	10-	2.3+
Range	?	11-	4+	5+	6+	9+	10+	2.3-
Date	?	11-	5+	5-	10/17/20+	10.6+	10-	2.3-

类型	类型名称	功能描述
color	颜色选择器	输入颜色值的文本框
date	日期选择器	输入日期的文本框
datetime	日期时间选择器	输入 UTC 日期和时间的文本框
datetime-local	日期时间选择器（本地）	输入本地日期和时间的文本框
month	月份选择器	输入月份的文本框
time	时间文本框	输入时间的文本框
week	周选择框	输入周的文本框
email	邮件输入框	输入 E-mail 地址的文本框
number	数字输入框	输入数字的文本框，可以设置输入值的范围
range	数字滑动条	通过拖动滑动条改变一定范围内的数字
search	搜索输入框	输入搜索关键字操作的文本框
tel	电话号码输入框	输入电话号码
url	Web 地址输入框	输入 URL 地址的文本框

Demo

NEW HTML5 INPUT ATTRIBUTES

							
Placeholder	4+	4/5+	4+	10+	11.10/11.50	10+	2.3+
Autofocus	4+	5+	5-	6+	11+	10+	2.3-
Maxlength	3.6/4+	5+	4+	6+	11+	9/10	2.3+
List (Datalist)	4+	5-	5-	20+	10.6+	10+	2.3-
Autocomplete	4+	5.2+	N/A	17+	10.6+	10-	2.3-
Required	6+	5-	5-	6+	10.6+	10+	2.3-
Pattern	4+	5-	5-	6/10+	10.6/11+	10+	2.3-
Spellcheck	3.6+	4+	5-	10+	11+	10+	N/A
Novalidate	4+	5-	5-	6+	10.6+	10+	2.3-
Formnovalidate	4+	5.2-	5-	6+	10.6+	10+	2.3-
Formaction	4+	5.2+	5+	10+	10.6+	10+	2.3-
Formmethod	4+	5.2+	5+	10+	10.6+	10+	2.3-
Formtarget	4+	5.2+	5+	10+	10.6+	10+	2.3-
Formenctype	?	?	?	?	?	?	?

PLACEHOLDER

当用户还没有输入值时,输入型控件可能通过placeholder向用户显示描述性说明文字或者提示信息

	Firefox 4+	Safari 4	Safari 5+	Safari Mobile iOS 4+	Chrome 10+	Opera 11.10	Opera 11.50+	IE 10+	Android 2.3
input	✓	✓	✓	✓	✓	✓	✓	✓	✓
textarea	✓	✗	✓	✓	✓	✗	✓	✓	?

PLACEHOLDER

```
<input type="search" name="user_search" placeholder="seach w3school">
```

[Demo](#)

AUTOCOMPLETE

浏览器通过autocomplete特性能够知晓是否应该保存输入值以备将来使用， autocomplete应该用于保护用户敏感数据，避免本地浏览器对它们进行不安全的存储。默认为on（开启）

这个所谓的自动完成指的是当用户曾经使用过的字符提交[submit]过，保存在浏览器cookie里面的用户使用痕迹。

和Ajax里的全文检索Autocomplete是另外一回事。

[Demo](#)

AUTOFOCUS

页面载入时，我们通过autofocus指定某个表单元素获得焦点

- 当页面中存在多个autofocus，自动聚焦最后一个
- 仅在input、textarea、button标签中使用



Demo

LIST(DATALIST)

通过使用list，能够为某个输入型控件构造一个选值列表，类似于js实现Autocomplete的效果。

datalist用法类似select标签



Demo

LIST(DATALIST)

```
列表: <input type="text" list="mydata" placeholder="热门电影排行">  
<datalist id="mydata">  
    <option label="让子弹飞" value="Top1">  
    </option><option label="非诚勿扰2" value="Top2">  
    </option><option label="大笑江湖" value="Top3">  
    </option><option label="赵氏孤儿" value="Top4">  
    </option><option label="初恋这件小事" value="Top5">  
</option></datalist>
```



REQUIRED

required属性规定必须在提交之前填写输入域（不能为空）。它是表单验证最简单的一种方式方法



REQUIRED

```
<input type="text" required="" name="username">
```

I'm Required

! 请填写此字段。

PATTERN

pattern属性规定用于验证input域的模式（pattern），模式（pattern）是正则表达式。

那些type为email或url的输入控件内置相关正则表达式，如果value不符合其正则表达式，那表单将通不过验证，无法提交。



PATTERN

在pattern使用属性时，下列限制条件适用：

- 输入必须与regex完全匹配。不允许部分匹配。
- 用于pattern的开头^和结尾\$的和锚（anchor）是自动暗含的。
- regex不应该包含于分隔符值（delimiter），例如斜线（forward slash）。
- 你不能使用限定语符（modifier），例如i来进行对大小写不敏感的匹配操作。

HTML5规范还建议title在属性中包含regex相应的描述。
浏览器将在输入验证失败的情形下使用它作为错误消息的一个组成部分。

```
<input type="text" name="arrival" placeholder="MM/DD/YYYY" pattern="^([0-9]{2})/([0-9]{2})/([0-9]{4})$">
```

Pattern

Zip Code:

Arrival Date:

Submit

Please match the format requested.

The date should be in the format MM/DD/YYYY with leading zeros where necessary

正则测试

! 请与所请求的格式保持一致。

扩展阅读

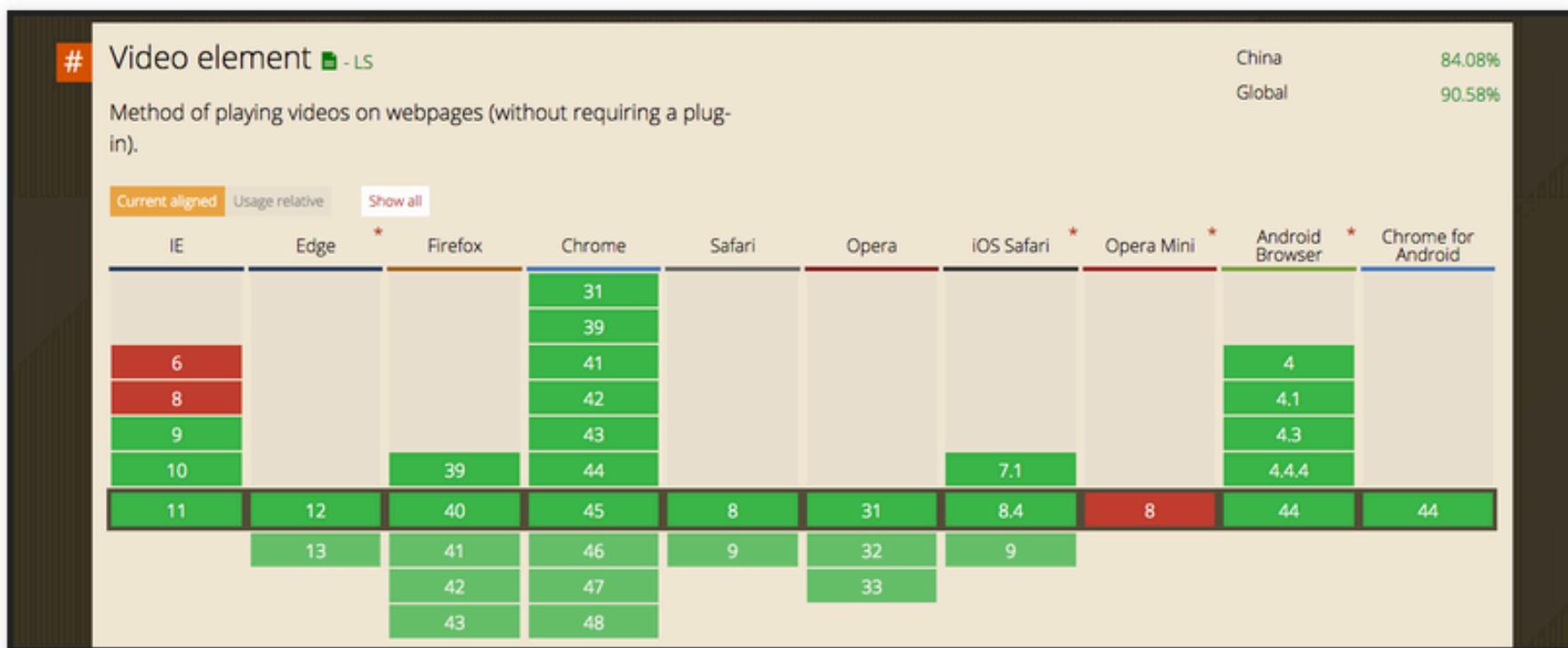
- 伪元素改变表单控件默认样式的完全展示
- 基于vertical-align的表单元素垂直对齐方式研究
- CSS实现input、checkbox垂直居中
- A form of madness
- Bring Your Forms Up to Date With CSS3 and HTML5 Validation

**VIDEO AUDIO
FILEREADER
DRAG DROP**

Created by [ChaiChunyan](#) / [@chaichunyan](#)

VIDEO

VIDEO



VIDEO

- No standard across browsers
- IE and Safari currently only support MP4
- MP4 is not free – there are patent/license issues
- WEBM – the current alternative

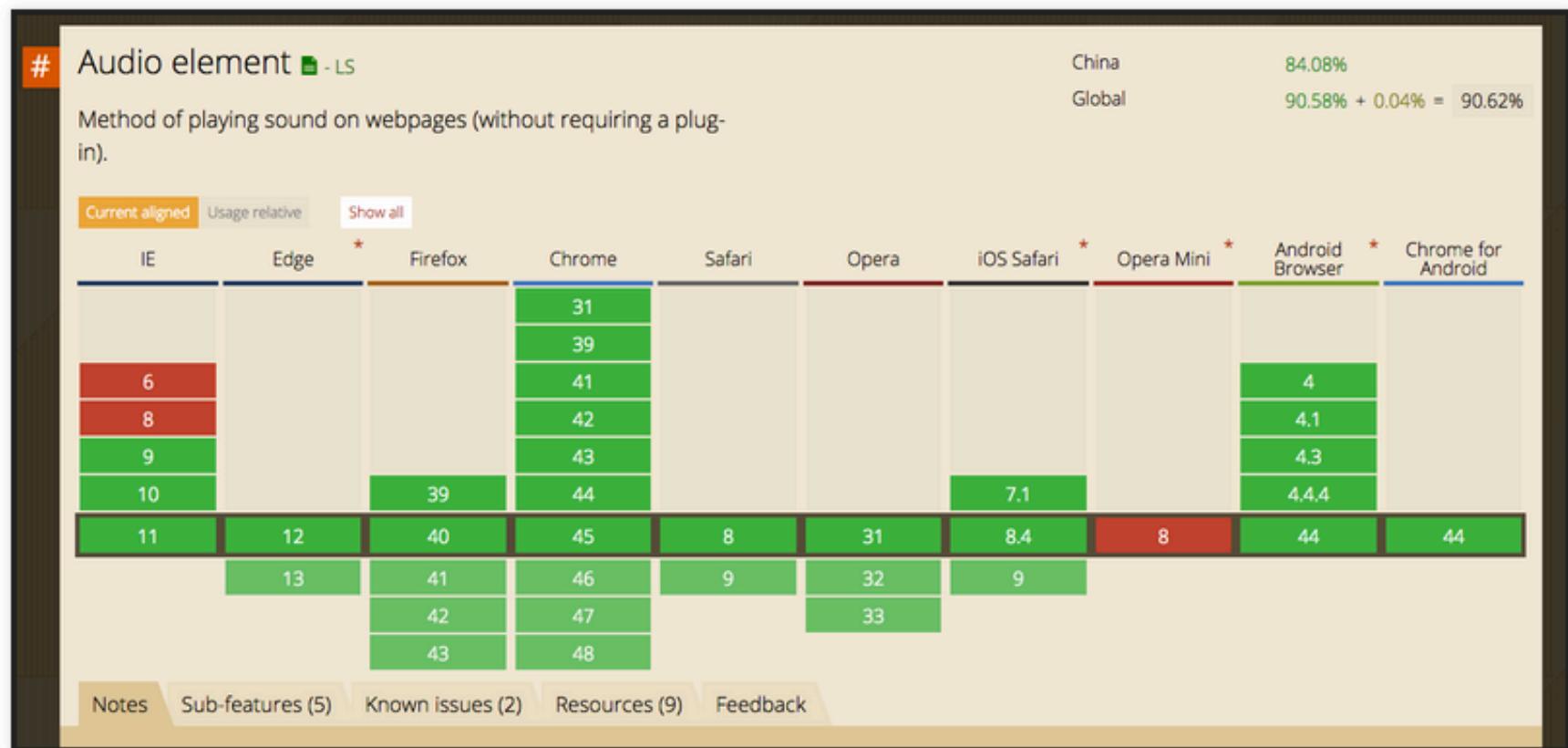
	MP4	WEBM
IE	Yes	No
Safari	Yes	No
Firefox	No	Yes
Chrome	No	Yes
Opera	No	Yes

VIDEO

- <video> element
- Attributes include:
 - src (takes a url)
 - controls
 - autoplay
 - loop
 - poster (takes a url)
 - preload
 - none
 - metadata
 - auto (default)

AUDIO

AUDIO



AUDIO

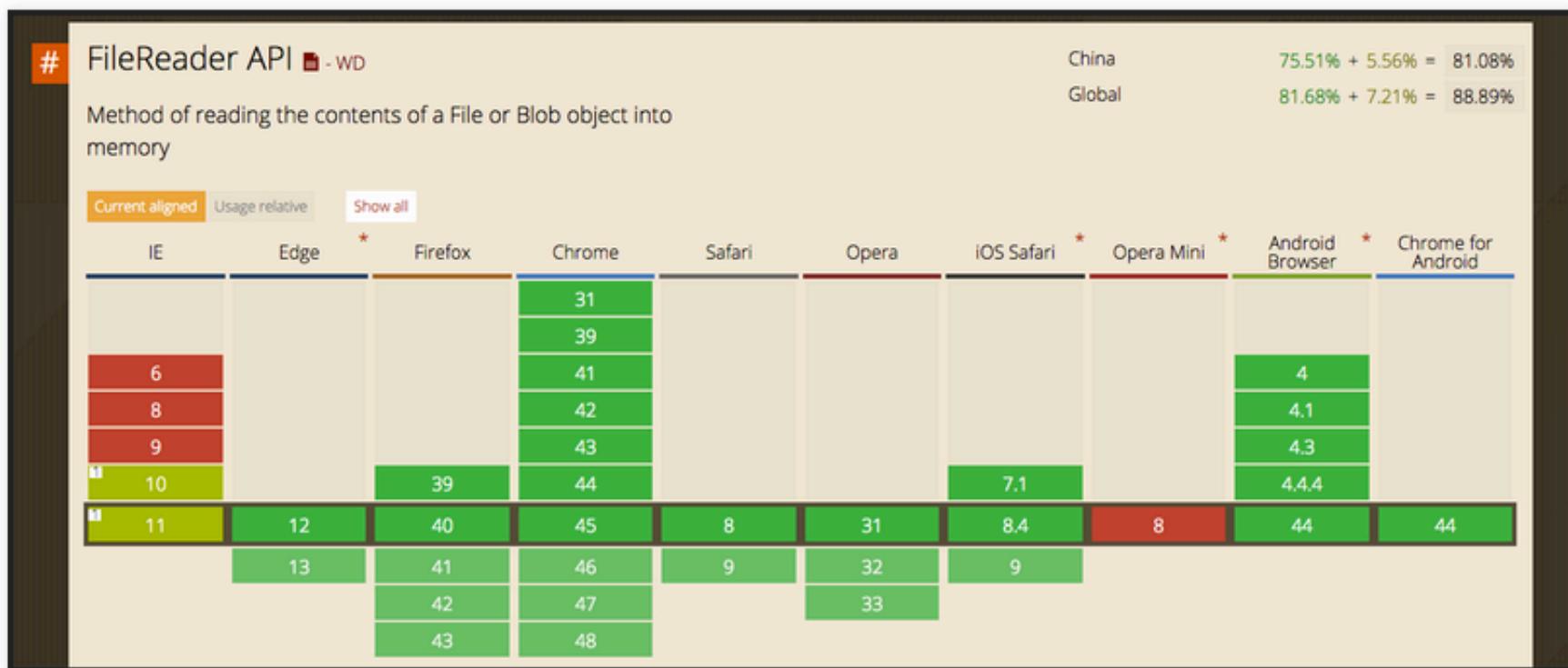
- No one standard across browsers
- IE and Safari currently only support MP3
- MP3 is not free – there are patent/license issues
- Other codecs
 - Vorbis (the audio codec in the OGG container)
 - Firefox and Opera
 - MP3
 - IE and Safari

AUDIO

- <audio> element
- Attributes include:
 - src (takes a url)
 - controls
 - autoplay
 - loop
 - preload
 - none
 - metadata
 - auto (default)

FILERADER

FILEREADER



FILEREADER

HTML5 提供的一种通过 File API 规范与本地文件交互的标准方式。

FILEREADER

- demo : 上传文本文件
- demo : 上传图片文件

DRAG, DROP

DRAG, DROP



demo : 多个拖放实例

CANVAS, WEBGL

Created by [ChaiChunyan](#) / [@chaichunyan](#)

CANVAS

CANVAS简介



CANVAS 能做什么？

3D 旋转模型

[17 Experimental Examples of Using HTML5 Canvas](#)

CANVAS

Canvas元素是HTML5的一部分，允许脚本语言动态渲染位图像。Canvas由一个可绘制地区HTML代码中的属性定义决定高度和宽度。JavaScript代码可以访问该地区，通过一套完整的绘图功能类似于其他通用二维的API，从而生成动态的图形。

CANVAS 基本使用

矩形, 渐变

- strokeRect, fillRect, clearRect
- strokeStyle, fillStyle

DEMO

- Canvas 绘制矩形
- Canvas 绘制渐变

PATHS

- `moveTo(x,y)`
- `lineTo(x,y)`
- `closePath`
- `clip`
- `rect(x, y, width, height)`
- `arc(x, y, radius, startAngle, endAngle, direction)`
- `quadraticCurveTo(cpx, cpy, x, y)`
- `bezierCurveTo(cp1x, cp1y, cp2x, cp2y, x, y)`

DEMO

- Drawing Lines
- Clipping Region
- DrawingWithArc
- Bezier Curve
- Quadratic Curve

TEXT

- font – sets the font properties using the same values as CSS
- textAlign
 - start
 - end
 - left
 - right
 - center

TEXT

- `textBaseline`
 - `top`
 - `hanging`
 - `middle`
 - `alphabetic`
 - `ideographic`
 - `bottom`
- `measureText(text)`

DEMO

- Text

WEBGL

WEBGL



WEBGL简介

WebGL是一种基于Web的Graphic库，由非盈利组织Khronos运营，目前结合HTML5元素广泛应用在3D图形开发中。

学习WebGL比较困难，因为它是底层开发——它运行在GPU上面，而且它实际上是一个OpenGL的JavaScript port，是一种游戏开发者使用的已经长期建立的API集。WebGL的主要受众是哪些已经拥有丰富OpenGL经验的游戏开发者，他们可以通过WebGL为web平台编写游戏。好在有很多资源可以帮助您学习WebGL，这些资源不仅仅是关于游戏开发的，还有很多奇幻的图形、视觉和音乐视频等方面。

WEBGL可以做什么

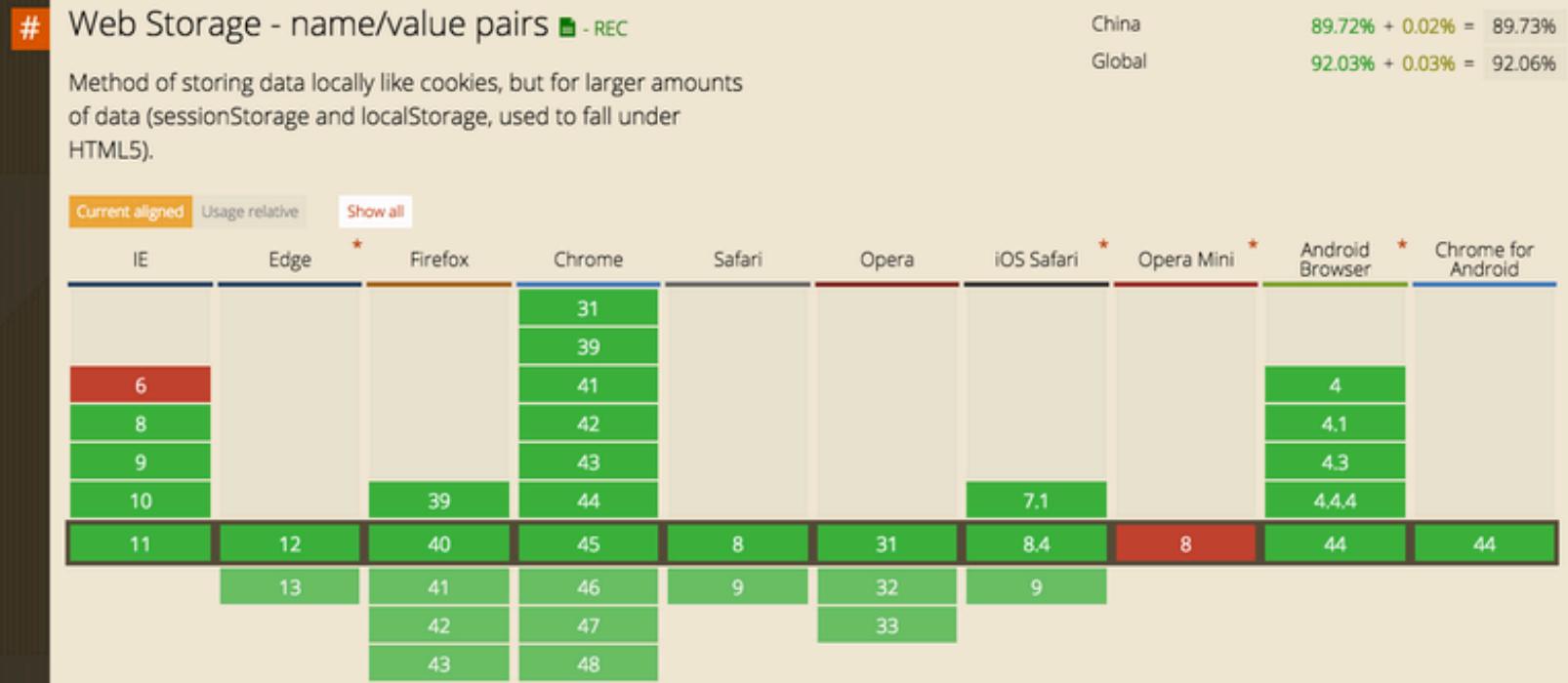
- 3D 汽车模型
- 太阳模型

WEB STORAGE

Created by [ChaiChunyan](#) / [@chaichunyan](#)

LOCALSTORAGE

浏览器支持



特点

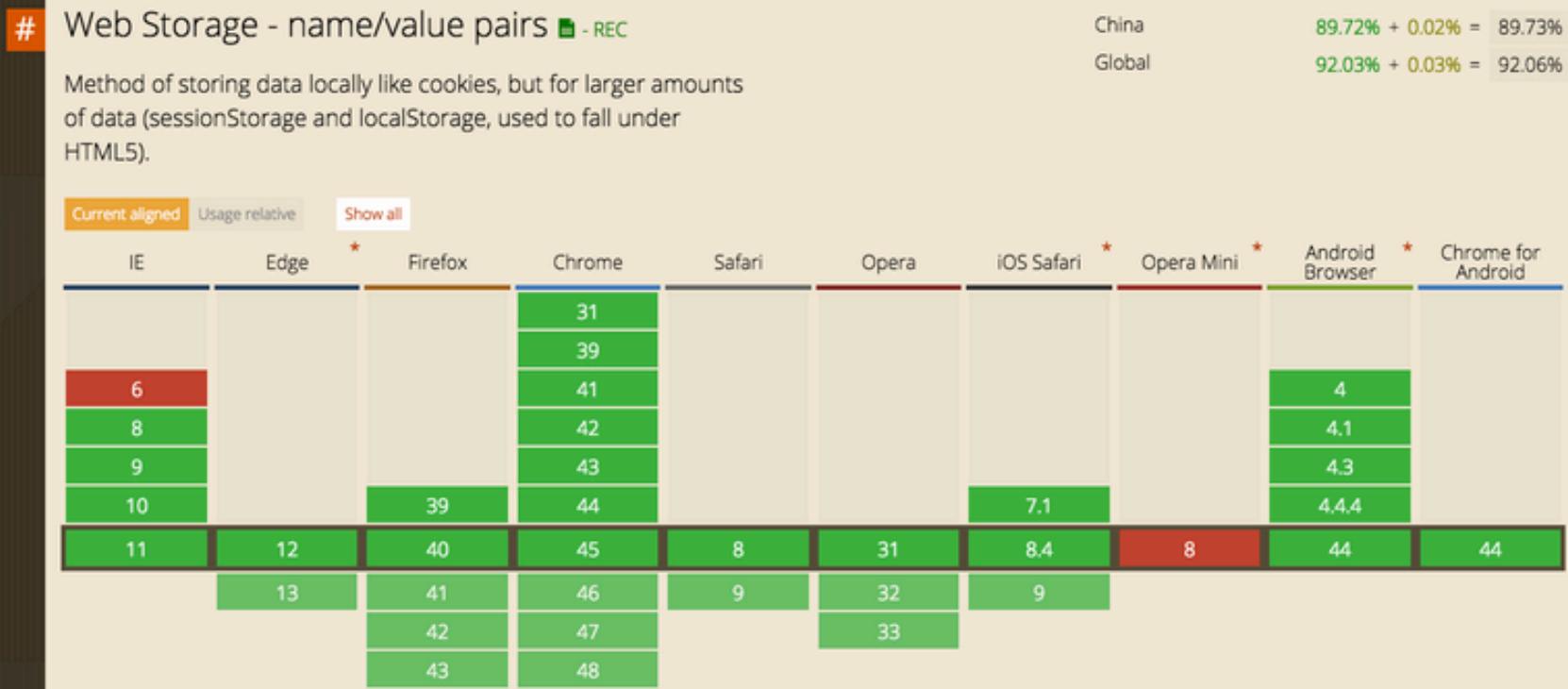
- localStorage只把数据存储在了客户端使用，不会发送到服务器上（除非你故意这样做）。
- 而且对于某一个域下来说，localStorage是共享的（多个窗口共享一个“数据库”）。
- localStorage用于持久化的本地存储，除非主动删除数据，否则数据是永远不会过期的。

DEMO

- `localStorage`

SESSIONSTORGE

浏览器支持



特点

- cookie的存储限制在了4k之内，相比来说，sessionstorage有了更大的存储空间，但至于具体多大，这要参照浏览器厂商的具体实现。
- cookie有一个机制，就是在每次客户端请求服务器的时候都会将cookie发送给服务器，这无疑会做很多不必要的操作，因为并不是每次请求服务器都需要cookie的所有信息，而sessionstorage很好的解决了这个问题，它不是采取自动发送的方式，这样就减少了不必要的工作。
- 通过sessionStorage所存储数据的生命周期，和Session类似，关闭浏览器（或标签页）后数据就不存在了。但刷新页面或使用“前进”、“后退按钮”后sessionStorage仍然存在。

特点

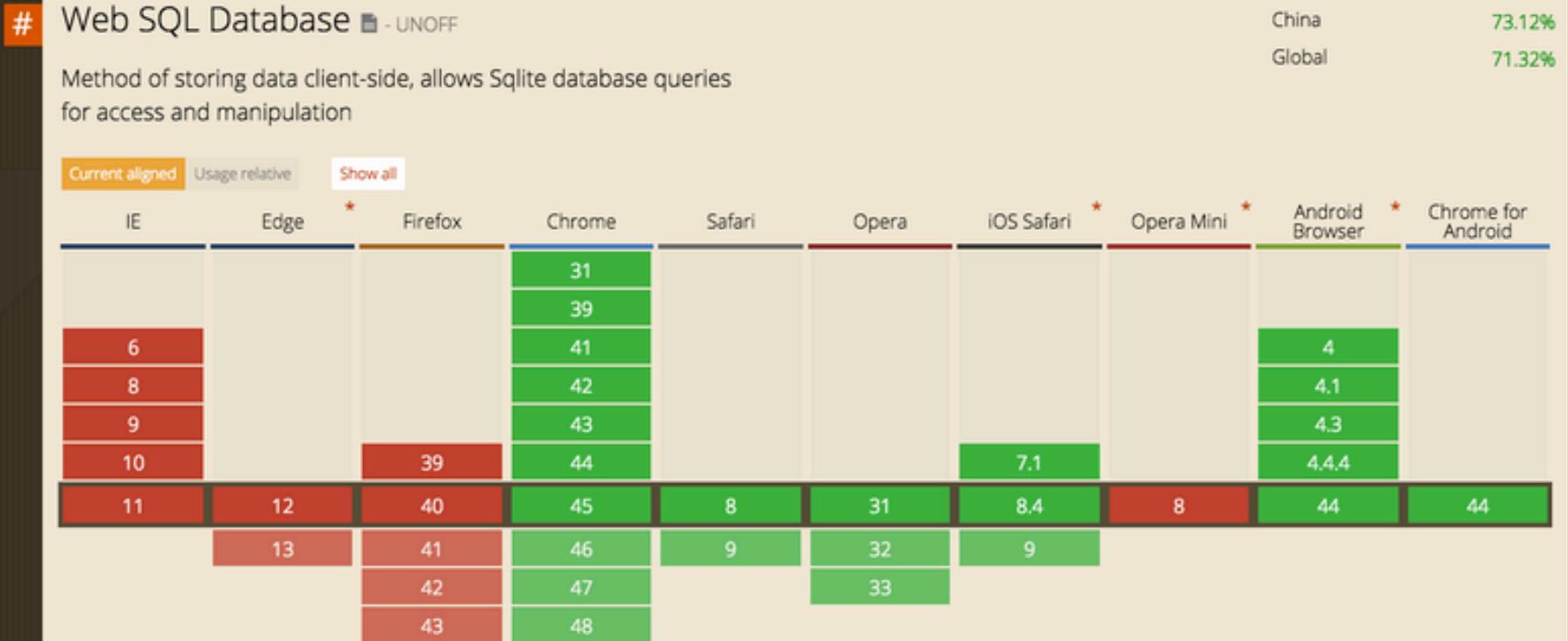
- session storage每个窗口的值都是独立的（每个窗口都有自己的数据），它的数据会随着窗口的关闭而消失，窗口间的sessionStorage也是不可以共享的。
- setItem中的key、value使用的是字符串的形式进行存储的。也就是说如果有如下的代码：setItem('count', 1);通过getItem('count') + 5 得到的将不是预期的6(整数)，而是'16'(字符串)。
- 再次使用setItem设置已经存在的key的value时，新的值将替代旧的值。
- 当存储中的数据发生改变时，会触发相应的事件（window.onstorage），但目前各浏览器对此事件的支持并不完善，暂时可以忽略。

DEMO

- sessionObject
- sessionStorage
- sessionStorage

WEBSQL

浏览器支持



简介

localStorage 和 sessionStorage 对于存储少量的数据非常有用，但是对大量的结构化数据进行处理时，它就力所不及了，而这正是 HTML5 的“Web SQL Database” API 接口的应用所在。

扩展阅读

基于 HTML5 中的 Web SQL Database 来构建应用程序

离线存储

浏览器支持



简介

HTML5的离线存储是基于一个新建的.appcache文件的，通过这个文件上的解析清单离线存储资源，这些资源就会像cookie一样被存储了下来。之后当网络在处于离线状态下时，浏览器会通过被离线存储的数据进行页面展示。

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。

扩展阅读

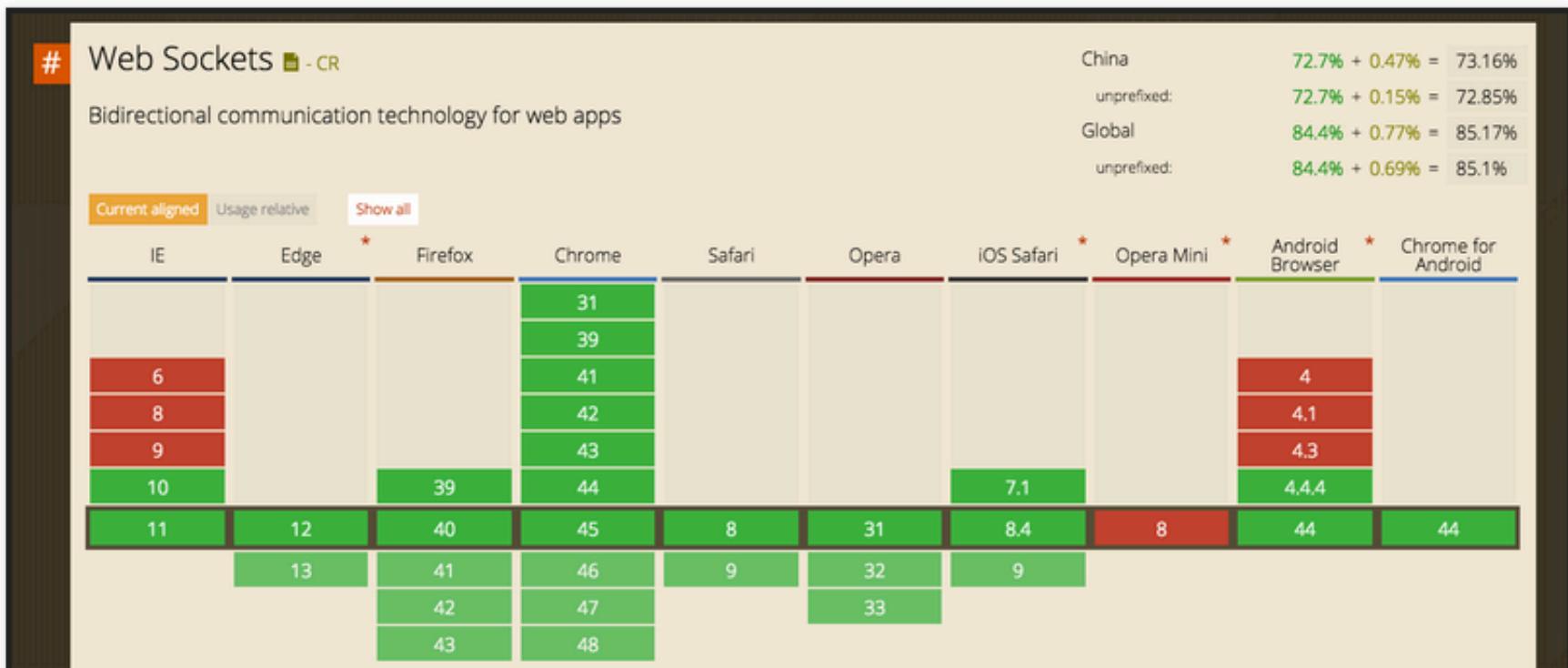
神奇的HTML5离线存储（应用程序缓存）

WEB SOCKET WEB WORKER

Created by [ChaiChunyan](#) / [@chaichunyan](#)

WEB SOCKET

WEB SOCKET



WEB SOCKET简介

WebSocket 是 HTML5 一种新的协议

它实现了浏览器与服务器全双工通信，能更好的节省服务器资源和带宽并达到实时通讯，它建立在 TCP 之上，同 HTTP 一样通过 TCP 来传输数据

举例

清单 1.WebSocket 客户端连接报文

```
GET /webfin/websocket/ HTTP/1.1
Host: localhost
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: xqBt3ImNzJbYqRINxEFlkg==
Origin: http://localhost:8080
Sec-WebSocket-Version: 13
```

举例

清单 2.WebSocket 服务端响应报文

```
HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: K7DJLdLooIwIG/MOpvWFB3y3FE8=
```

WEB WORKER

浏览器兼容性



WEB WORKER简介

Web Workers 出现之前，JavaScript 是现代 web 应用程序的核心。JavaScript 和 DOM 本质上都是单线程的：在任何时间都只能执行一个 JavaScript 方法。即使您的计算机有 4 个内核，在进行长期计算时，也只有一个内核比较繁忙。例如，您在计算到达月球的最佳轨道时，您的浏览器不能渲染一个显示轨迹的动画，以及 – 同时 – 对用户事件作出响应（比如鼠标单击或键盘输入）。

WEB WORKER简介

Web Workers 打破了传统 JavaScript 的单线程模式，引入了多线程编程模式。一个 worker 是一个独立的线程。有多个任务需要处理的 web 应用程序不再需要逐个处理任务。反之，应用程序可以将任务分配给不同的 workers。

CSS3

Created by [ChaiChunyan](#) / [@chaichunyan](#)

CSS3 与 CSS2

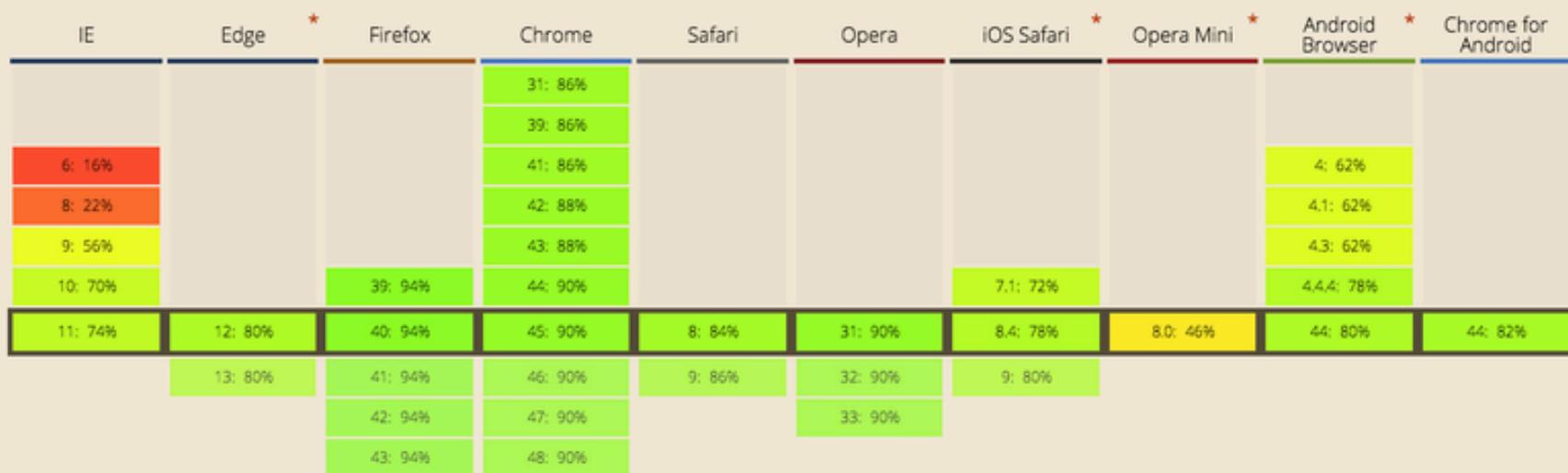
CSS3是CSS技术的升级版本，CSS3语言开发是朝着模块化发展的。以前的规范作为一个模块实在是太庞大而且比较复杂，所以，把它分解为一些小的模块，更多新的模块也被加入进来。这些模块包括：盒子模型、列表模块、超链接方式、语言模块、背景和边框、文字特效、多栏布局等。

CSS3 兼容性

兼容性列表

Summary

Calculation of support for currently selected criteria



■ = Supported ■ = Not supported ■ = Partial support ■ = Support unknown

各属性浏览器支持度查询

查询网站: [caniuse](#)

CSS3 常用属性

常用属性列表

- 选择器
- 自定义字体
- 多栏布局
- 文字,容器阴影
- 圆角
- 渐变效果
- 多背景图
- 翻转
- 动画
-

一个 DEMO 来说明

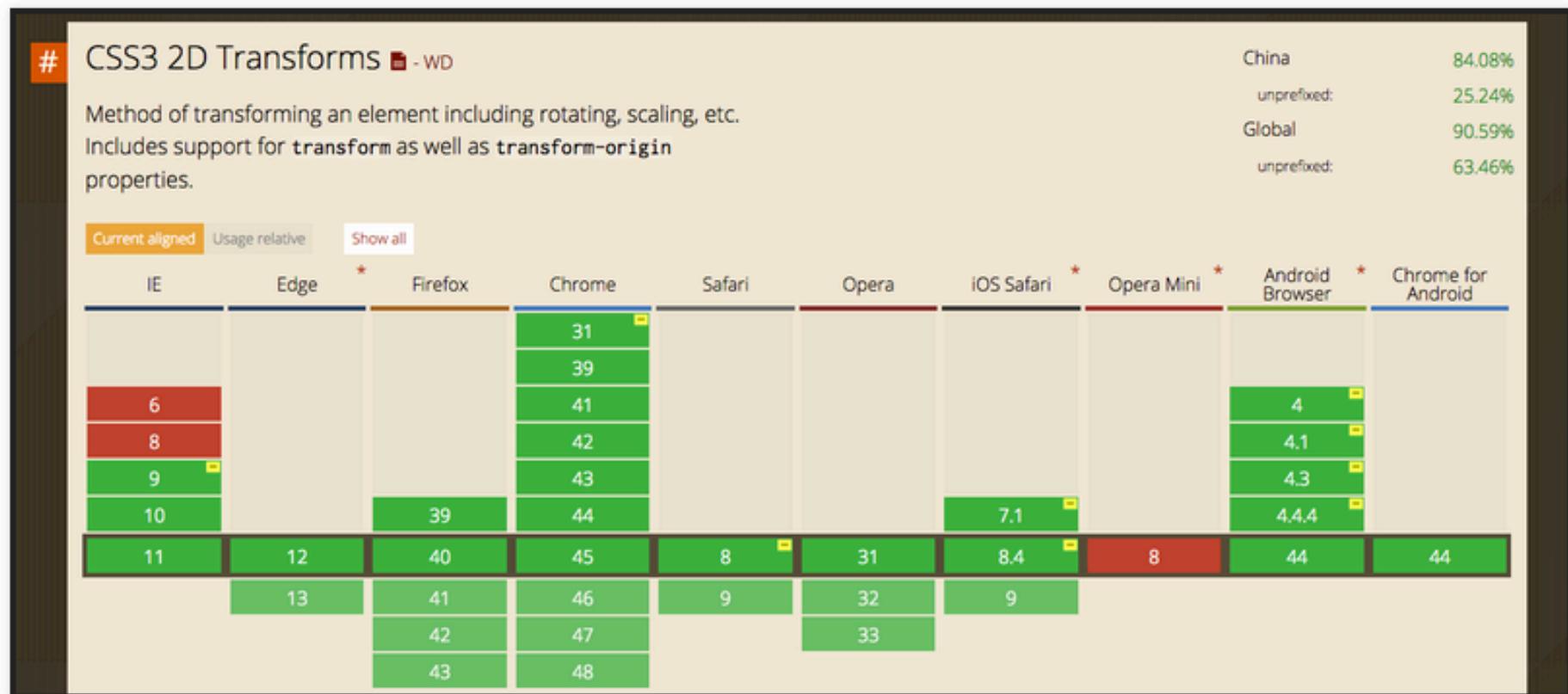
[Experimenting with CSS3](#)

ANIMATION TRANSFORM

Created by [ChaiChunyan](#) / [@chaichunyan](#)

TRANSITION

浏览器兼容性



CSS TRANSITION PROPERTIES

- Transition-property:
 - Specifies name or names of CSS Properties to animate
- Transition-duration:
 - Specifies duration of the animation
- Transition-delay
- Transition-timing-function

代码片段

The Transform Property

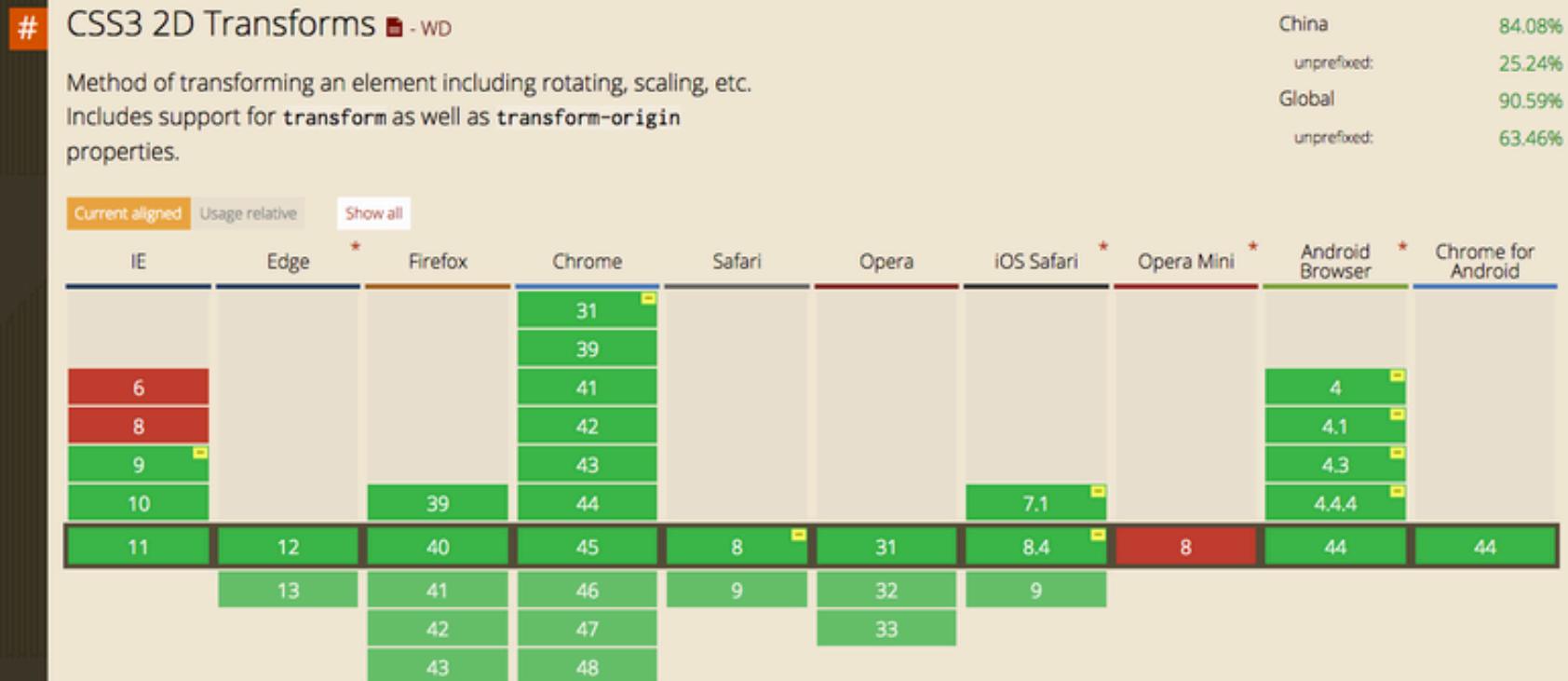
```
.box {  
height: 100px;  
width: 100px;  
background-color: red;  
transition: transform 2s;  
-webkit-transition: transform 2s;  
-moz-transition: transform 2s;  
-o-transition: transform 2s;  
-ms-transition: transform 2s;  
transition: transform 2s;  
}  
  
.box:hover {  
-webkit-transform: skewX(20deg);  
-moz-transform: skewX(20deg);  
-o-transform: skewX(20deg);  
-ms-transform: skewX(20deg);  
transform: skewX(20deg);  
}
```

DEMO

[Transition_base](#)

2D TRANSFORM METHODS

浏览器兼容



TRANSFORM METHODS

- Skew()
- Scale()
- Rotate()
- Translate()

SYNTAX

`transform: skew(parameters)`

代码片段

The Transform Property

```
.box {  
height: 100px;  
width: 100px;  
background-color: red;  
transition: transform 2s;  
-webkit-transition: transform 2s;  
-moz-transition: transform 2s;  
-o-transition: transform 2s;  
-ms-transition: transform 2s;  
transition: transform 2s;  
}  
  
.box:hover {  
-webkit-transform: skewX(20deg);  
-moz-transform: skewX(20deg);  
-o-transform: skewX(20deg);  
-ms-transform: skewX(20deg);  
transform: skewX(20deg);  
}
```

DEMO

Transform_base

综合实例

知识点

- Designing with 2D Transform
- Animating the Navigation Links
- Applying Multiple Transition Effects
- Pictures Gallery: scaling up with 2D Pictures
- Gallery: rotating in 3D
- Contact Us with an Original Button Link

综合实例

Whistler White Ski Resort

HTML5 WEBAPP

Created by [ChaiChunyan](#) / [@chaichunyan](#)



1999 -----> 2005 -----> 2008 ----->

移动设备的演化



纠结的去学习 OBJECTIVE-C?





- 平台选择
- Media Query
- 文字排版
- 流体布局
- 图片载入
- Dom操作性能优化
- 触屏事件
- ...

响应式设计独到之处

- 一套代码
- 一套开发环境
- 一个URL
- 内容所见即所得
- 不会因为终端升级而导致不可用
- ...

320



480



768



1024



偷懒的做法...

一种偷懒的做法是:等比缩放 但等比缩放后的页面文字像蚂蚁一样看不清楚

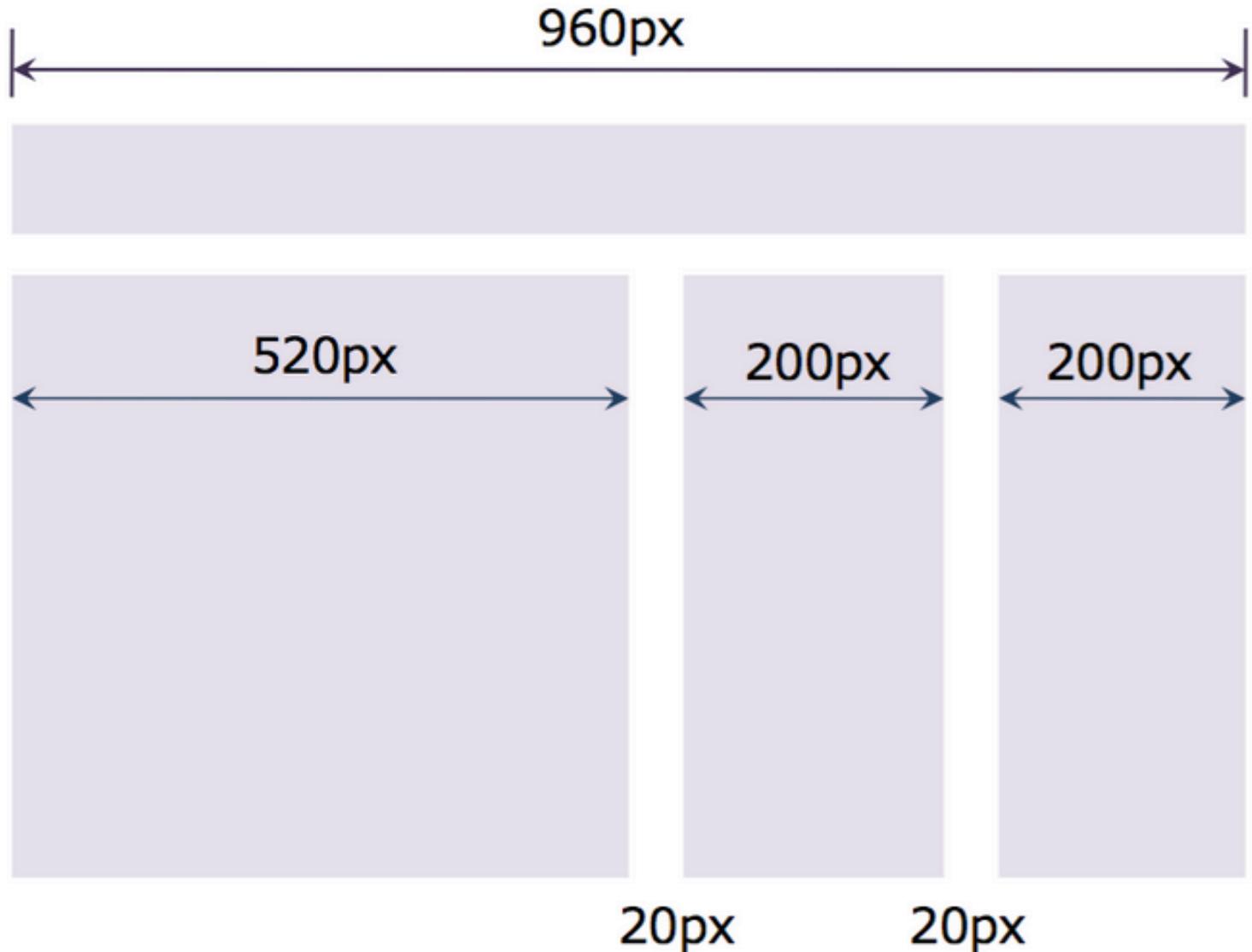


布局:百分比宽度+容器:浮动

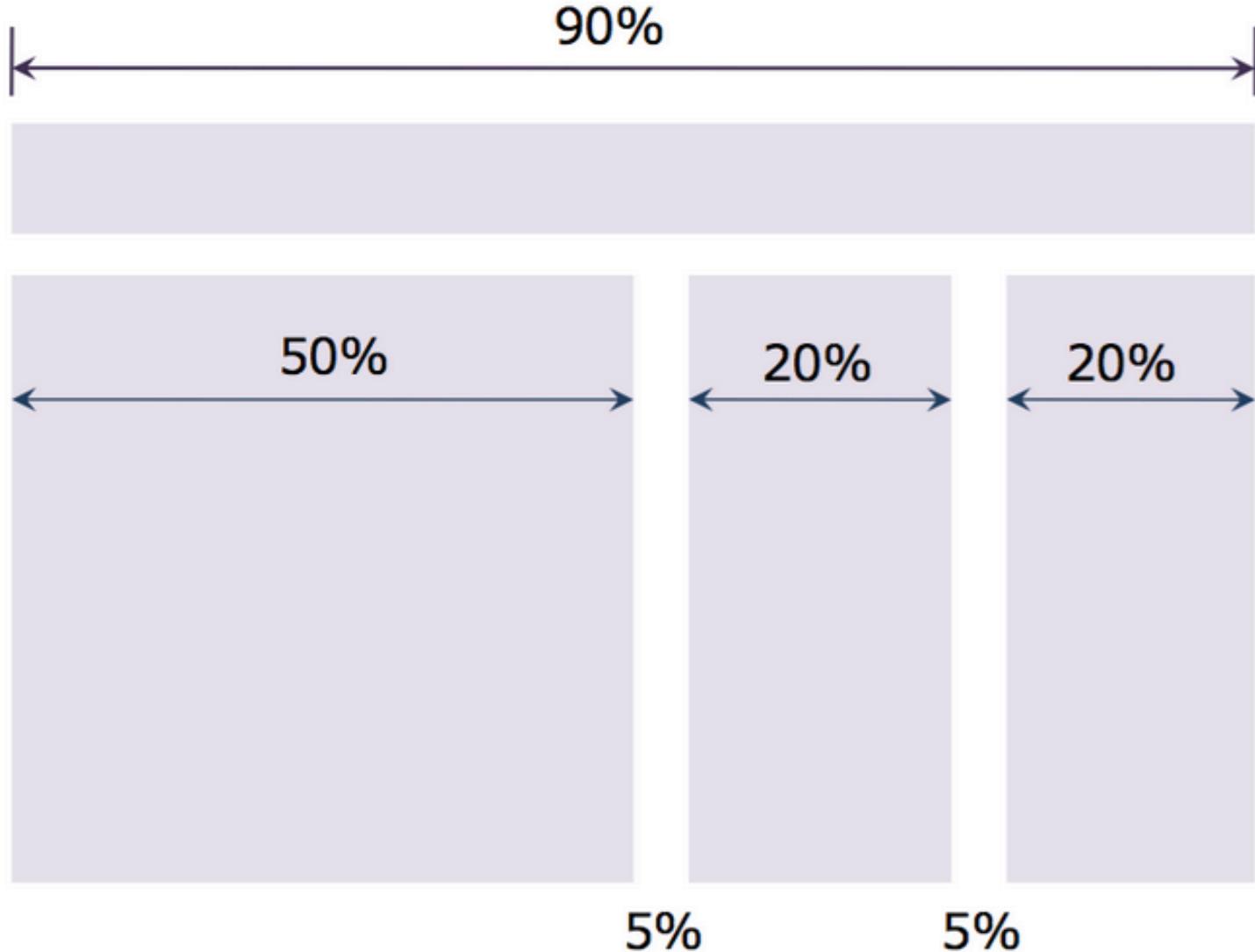
```
.wrapper {  
    width:100%;  
}
```

```
.wrapper .item {  
    float:left;  
}
```

定宽布局



流体布局



HOW DO

VIEWPORT META

```
<meta name="viewport" content="  
width=device-width, initial-scale=1">
```

VIEWPORT META 参数

- width=device-width
- user-scalable=1
- initial-scale=1
- maximum-scale=1
- minimum-scale=1



Visual viewport
= Layout viewport

VIEWPORT 参数最佳组合

```
<meta name="viewport" content="  
    width=device-width,  
    initial-scale=1,  
    maximum-scale=1">
```

MEDIA QUERY CSS

```
/* PC宽屏样式 */

/* iPad 及以下,所有小于(不等于)960宽度的平板电脑 */
@media only screen and (max-width: 959px) {}

/* 仅iPad 竖版,所有小于(不等于)960宽度的平板电脑的竖版 */
@media only screen and (min-width: 768px) and (max-width: 959px) {}

/* iPhone 及以下 */
@media only screen and (max-width: 767px) {}

/* 仅iPhone 横版,包括某些平板电脑的竖版 */
@media only screen and (min-width: 480px) and (max-width: 767px) {}

/* 仅iphone4 竖版 */
@media only screen and (max-width: 479px) {}
```

MEDIAQUERY书写思路

- 先写高分辨率样式？
- 还是，先写低分辨率样式？

先写高分辨率样式

原因

1,设计师设计原型往往首先基于宽屏
甚至不会给出适配设计稿

2,宽屏视觉元素更多,减法比加法容易

字体样式定义

reset.css

```
body {  
    font-family:tahoma,arial,\5b8b\4f53,sans-serif;  
}  
  
html {  
    -webkit-text-size-adjust: 100%;  
}
```

iOS : 华文细黑 + Helvetica



Android : 文泉驿微米黑 + Droid Sans



响应式图片

有两种基本的图片类型需要考虑

- 1, 需要缩放的图片
- 2, 不能缩放的图片



1.



2.

```
<style>
.selector-to-img{
    width:100px; height:100px; background:url("img-pc.png") no-repeat
}
@media only screen and [tablet] {
    background:url("img-tablet.png") no-repeat center;
}
@media only screen and [mobile] {
    background:url("img-mobile.png") no-repeat center;
}
</style>


```

针对单一设备做更极致的用户体验



```
<!doctype html>
<html manifest="http://www.../pad-sport-cache.php">
<head>
<!--ViewPortMeta设置， 禁止手动缩放-->
<meta name="viewport" content="
    width=device-width,
    initial-scale=1,
    maximum-scale=1">
<!--屏蔽拨号链接-->
<meta name="format-detection" content="telephone=no" />
<!--隐藏浏览器导航栏-->
<meta name="apple-mobile-web-app-capable" content="yes" />
<link rel="apple-touch-icon"
      sizes="72x72"
      href="http://cdn/img-72-72.png" />
</head>
...

```

几个新属性

```
<meta name="apple-mobile-web-app-capable"  
      content="yes">  
<meta name="apple-mobile-web-app-status-bar-style"  
      content="black" />  
<!--不自动将地址和email转为链接-->  
<meta name="format-detection"  
      content="address=no;email=no" />  
<!--添加到主屏时的图标-->  
<link rel="apple-touch-icon-precomposed"  
      href="http://cdn/img-114-114.png">  
<link rel="apple-touch-startup-image"  
      href="http://cdn/img-320-460.png">
```

终端事件的渐进增强

触屏事件不要用 'CLICK' 代替

触屏TOUCH事件

```
if ('ontouchstart' in document.documentElement) {  
    node.delegate('touchstart', function(e){  
        var x = e.changedTouches[0].clientX;  
        //...  
    });  
    node.delegate('touchend', function(e){  
        var x = e.changedTouches[0].clientX;  
        //...  
    });  
    node.delegate("touchmove",function(e){  
        var current_x = e.touches[0].pageX;  
    });  
    //...  
}
```

```
if ('ontouchstart' in document.documentElement) {  
    node.delegate('touchstart', function(e){  
        var x = e.changedTouches[0].clientX;  
        //...  
    });  
    node.delegate('touchend', function(e){  
        var x = e.changedTouches[0].clientX;  
        //...  
    });  
    node.delegate("touchmove",function(e){  
        var current_x = e.touches[0].pageX;  
    });  
    //...  
}
```

移动WEB中的性能问题



移动设备和浏览器性能概况

	型号	CPU	RAM
iOS	iPhone 4S	双核A5 800MHZ	512M
	iPhone 4	A4 800MHZ	512M
	iPhone 3GS	S5PC100 600MHZ	256M
Android	Galaxy Note	Exynos 双核 1.4GHZ	1G
	Nexus One	高通 1GHZ	512M
	MOTO XT615	高通 800MHZ	512M
	HTC Legend	高通 600MHZ	384M

常见移动设备硬件情况

	版本	渲染引擎	JS引擎
iOS	iOS 4.3+	Web Core	Nitro *
	Older iOS	Web Core	JavaScript Core
Android	Android 2.2+	Web Core	V8
	Older Android	Web Core	JavaScript Core

严峻的现实

- 1,低端设备众多
- 2,系统分配给浏览器的资源有限 ([参考](#))
- 3,简化的浏览器实现
- 4,不少老旧引擎
- 5,需要高性能支持的HTML5 OPOA

WEB APP 的性能优化

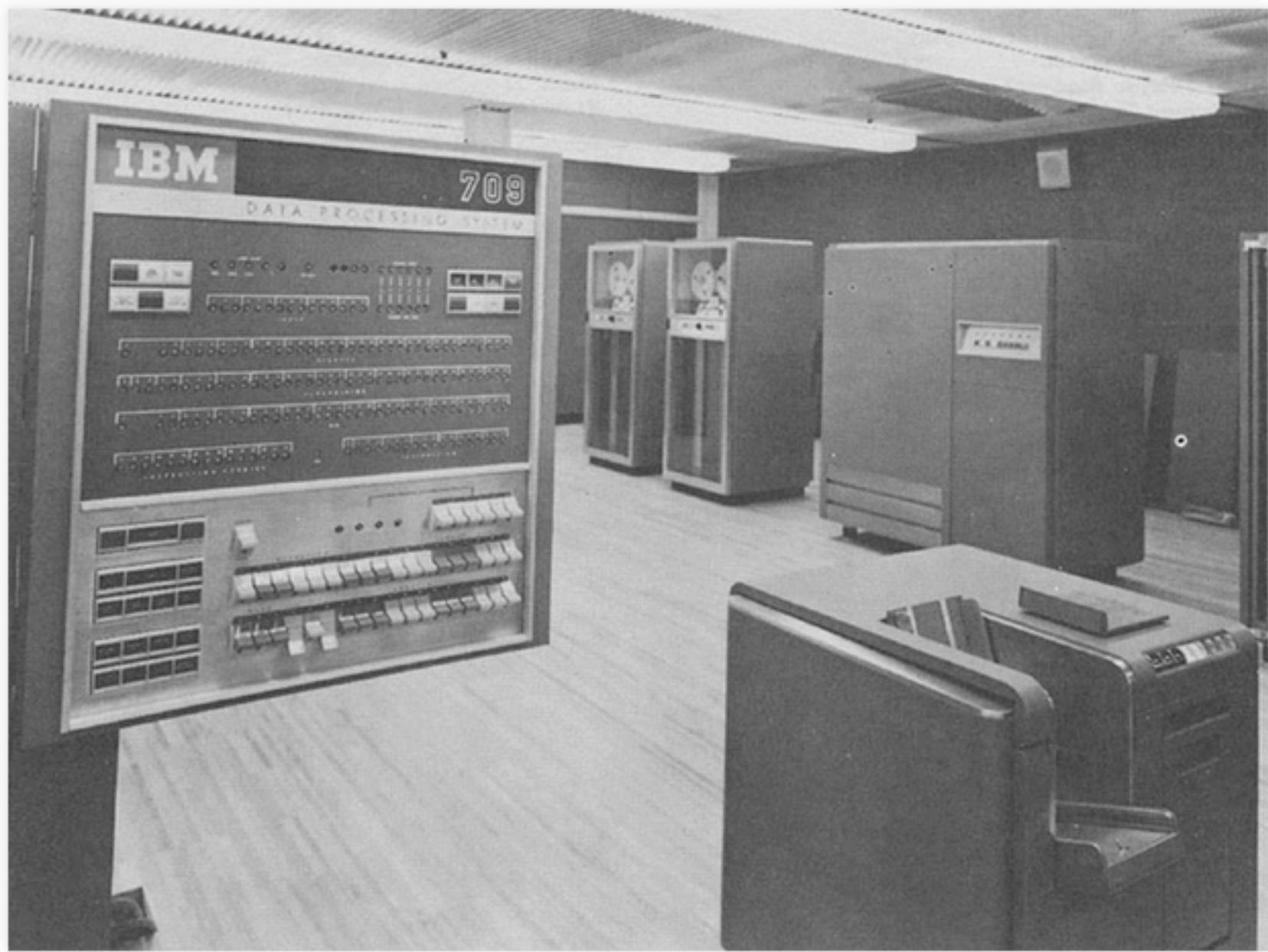
1. 处理性能(CPU&RAM)

- Reflow & Repaint
- CSS3的性能问题
- 动画
- JS中的内存控制
- 高效的JS技巧
- 关于电量
- HTML5带来的优化

2. 网络性能 Network

HTML5 HYBIRD DEVELOPMENT

Created by [ChaiChunyan](#) / [@chaichunyan](#)









MORE ...

- browsers
- smartphones
- tablets
- TVs

AND MORE ...



HOW DO ?

3种解决的方式

- Native
- webapp
- Hybird

NATIVE

优势

- 性能最佳
- 更加流程的动画
- 原生 API 调用

不足

- 不同的平台使用不同的开发语言
- 时间
- 成本
- ...
- iOS应用还要经过漫长的审核发布

WEBAPP

优势

- 单页应用
- 无须安装
- 广大web开发人员

不足

- 无法调用硬件API
- 有限的存储空间
- 性能不足
- 延迟、卡顿
- ...

HYBIRD

目前主流的两种混合方案

- 跨平台解决方案
- Native + HTML5

跨平台解决方案

- PhoneGap
- Inioc
- Titanium

NATIVE + HTML5

Native提供 JS 方法；
HTML5 通过 JS 同 Native 交互

NODEJS

Created by [ChaiChunyan](#) / [@chaichunyan](#)

什么是 NODEJS

Node 是一个服务器端 JavaScript 解释器，用于方便地搭建响应速度快、易于扩展的网络应用。

优势和劣势

优势

1、采用事件驱动、异步编程，为网络服务而设计。其实Javascript的匿名函数和闭包特性非常适合事件驱动、异步编程。而且JavaScript也简单易学，很多前端设计人员可以很快上手做后端设计。

优势

2、Node.js非阻塞模式的IO处理给Node.js带来在相对低系统资源耗用下的高性能与出众的负载能力，非常适合用作依赖其它IO资源的中间层服务。

优势

3、Node.js轻量高效，可以认为是数据密集型分布式部署环境下的实时应用系统的完美解决方案。Node非常适合如下情况：在响应客户端之前，您预计可能有很高的流量，但所需的服务器端逻辑和处理不一定很多。

劣势

1、可靠性低

劣势

2、单进程，单线程，只支持单核CPU，不能充分的利用多核CPU服务器。一旦这个进程崩掉，那么整个web服务就崩掉了。

使用场景

使用场景

- 聊天室
- 对象数据库接口
- 队列输入
- 数据流
-

聊天室

聊天应用程序是最能体现 Node.js 优点的例子：轻量级、高流量并且能良好的应对跨平台设备上运行密集型数据（虽然计算能力低）。

Demo：[在线聊天室](#)

对象数据库接口

尽管，Node.js 确实非常擅长实时交互的应用，同时它也十分适合通过对象数据库（object DB）来查询数据（如 MongoDB）。以 JSON 格式存储的数据允许 Node.js 直接处理，不需要纠结数据转换和匹配的问题。

不适合场景

- 使用关系型数据库的服务端 WEB 应用
- 繁重的服务端的计算和处理

使用关系型数据库的服务端 WEB 应用

对比 Node.js 上的 Express.js 和 Ruby on Rails，当你使用关系型数据库的时候请毫不犹豫的选择后者。

繁重的服务端的计算和处理

当涉及到大量的计算，Node.js 就不是最佳的解决方案。你肯定不希望使用 Node.js 建立一个斐波那契数的计算服务。一般情况下，任何 CPU密集型操作 会削弱掉 Node通过事件驱动，异步 I/O 模型等等带来的在吞吐量上的优势，因为当线程被非异步的高计算量占用时任何传入的请求将被阻塞。

前端工程化的一些心得

Created by ChaiChunyan / [@chaichunyan](#)

前后端分离

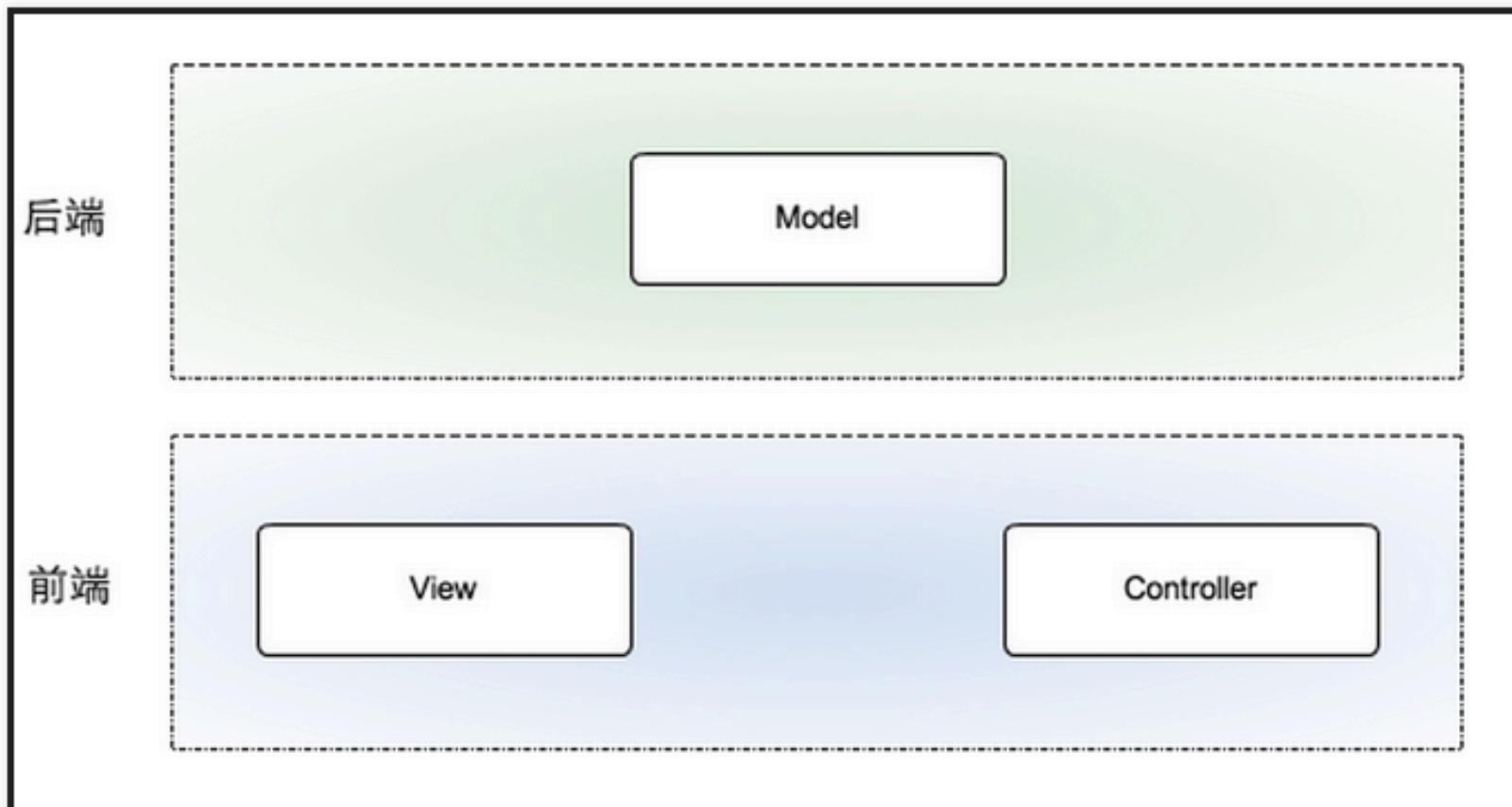
为什么要前后端分离

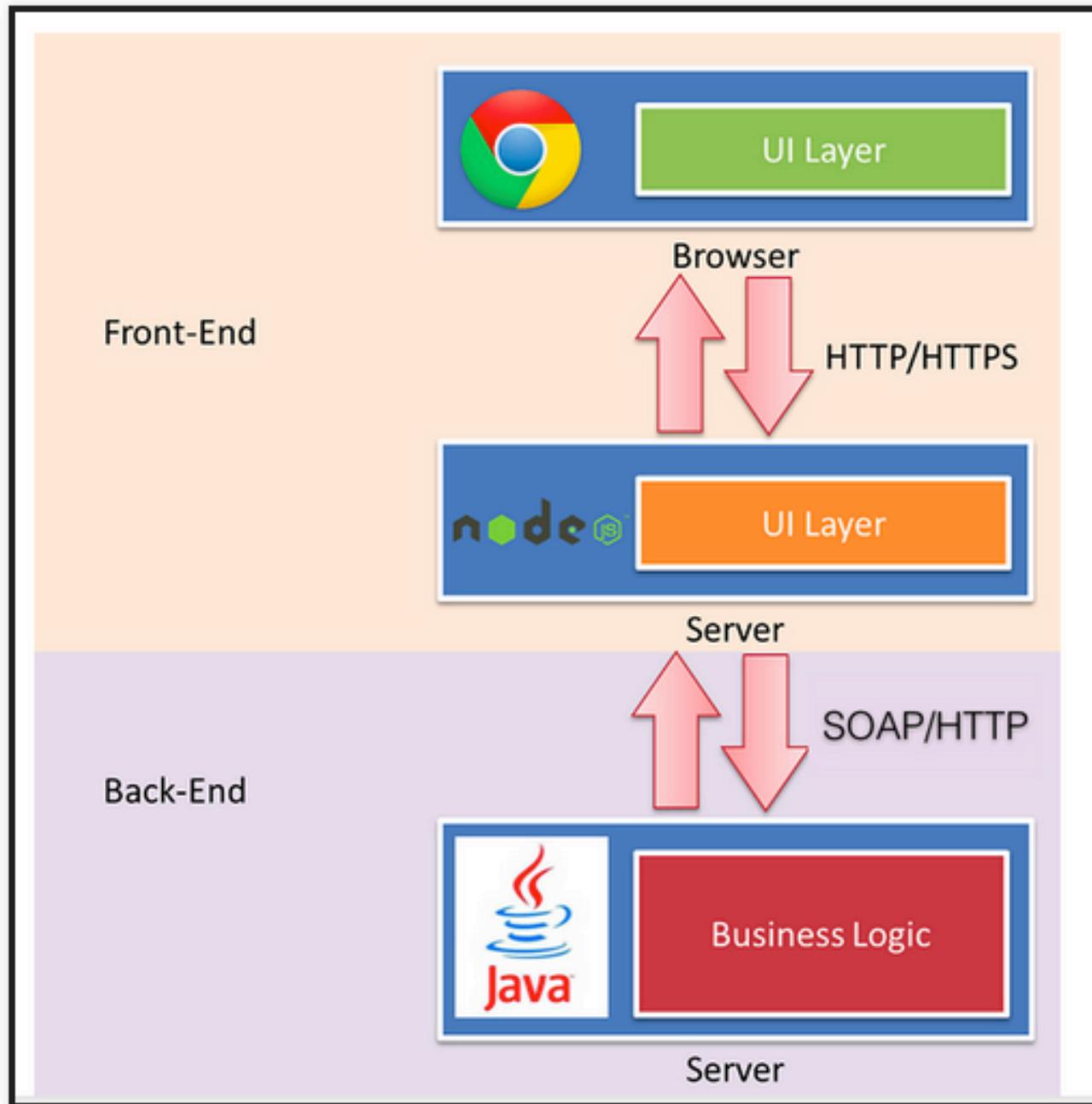
- 前后端职责不清
- 开发效率问题
- 对前端发挥的局限

前后端分离

前端：负责View和Controller层。

后端：只负责Model层，业务处理/数据等。





性能问题

分层就涉及每层之间的通讯，肯定会有一定的性能损耗。但是合理的分层能让职责清晰、也方便协作，会大大提高开发效率。分层带来的损失，一定能在其他方面的收益弥补回来。

另外，一旦决定分层，我们可以通过优化通讯方式、通讯协议，尽可能把损耗降到最低。

我们是怎么做前后端分离的

前端自动化 (GULP)

GULP 是什么

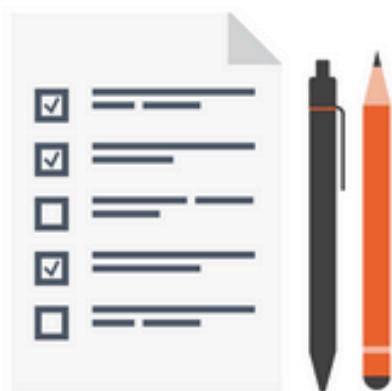
基于文件流的构建系统

[官网](#)

为什么要使用 GULP

现实中,我们有很多重复工作

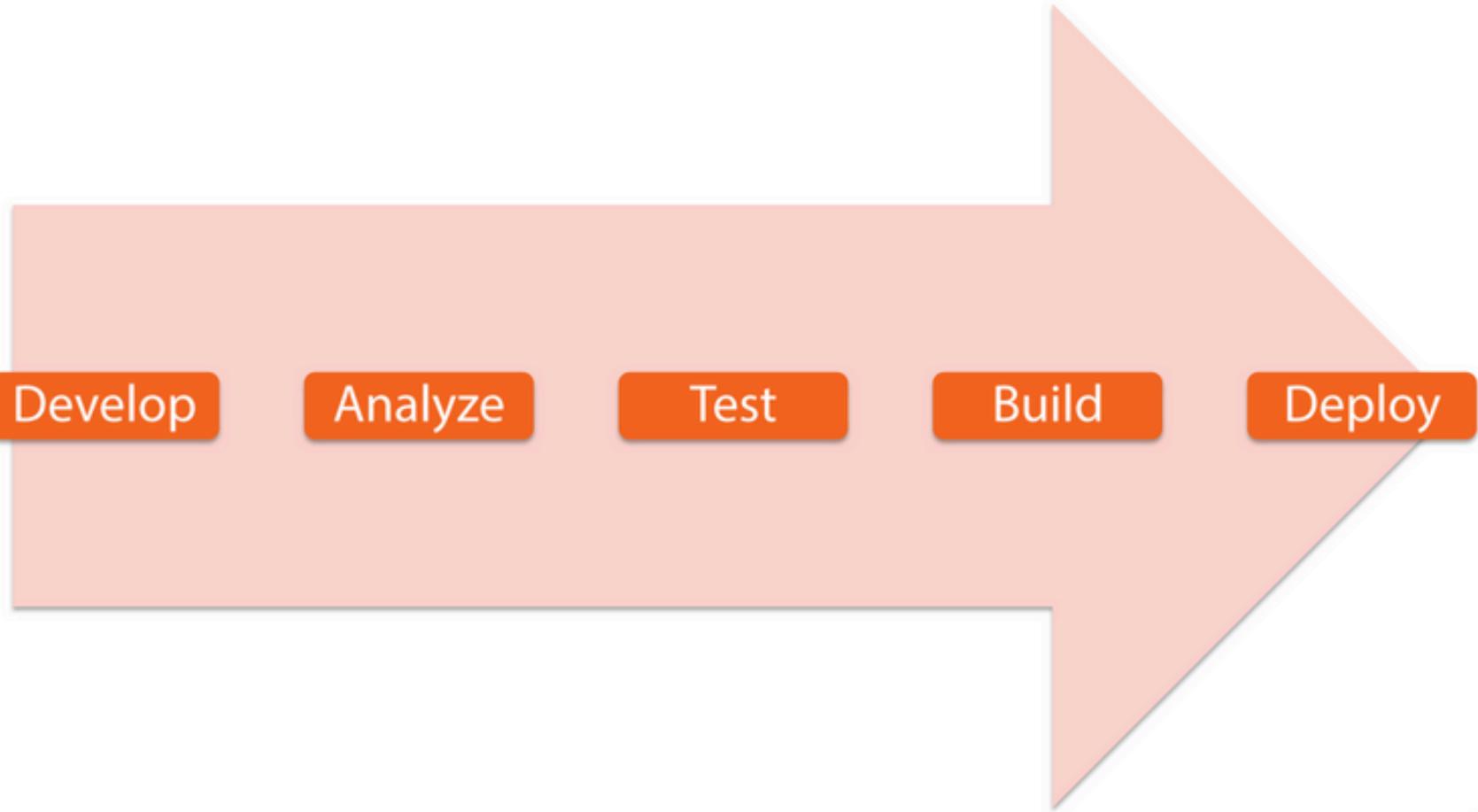
Preparing JavaScript/HTML/CSS for Production



- Minify
- Concat
- Vendor Prefixes
- Less to CSS Compilation
- Vendor and Custom Code
- Injecting Files into HTML
- Cache Busting
- Angular Template Cache
- Testing
- Code Analysis

SO...

Automated Build Pipeline



GULP VS GRUNT

How Grunt and Gulp Differ

Grunt



- Configuration over code
- File based
- 3900+ plugins

Gulp



- Code over configuration
- Stream based
- 1100+ plugins
- Use node more readily

More
Readable and
Easier to
Debug

ONLY 4 SIMPLE APIs



1. gulp.task
2. gulp.src
3. gulp.dest
4. gulp.watch



前端性能优化

一、内容部分

- 尽量减少 HTTP 请求次数
- 减少 DNS 查找次数
- 避免跳转
- 可缓存的 AJAX
- 推迟加载内容
- 预加载
- 减少 DOM 元素数量
- 根据域名划分页面内容
- 使 iframe 的数量最小
- 不要出现 404 错误

二、服务器端

- 使用内容分发网络
- 为文件头指定Expires或Cache-Control
- Gzip压缩文件内容
- 配置ETag
- 尽早刷新输出缓冲
- 使用GET来完成AJAX请求

三、JAVASCRIPT 和 CSS

- 把样式表置于顶部
- 避免使用CSS表达式(Expression)
- 使用外部JavaScript和CSS
- 削减JavaScript和CSS
- 用代替@import
- 避免使用滤镜
- 把脚本置于页面底部
- 剔除重复脚本
- 减少DOM访问
- 开发智能事件处理程序

四、COOKIE,图片及移动应用

- 减小Cookie体积
- 对于页面内容使用无coockie域名
- 优化图像
- 优化 CSS Sprite
- 不要在 HTML 中缩放图像
- favicon.ico 要小而且可缓存
- 保持单个内容小于 25K
- 打包组件成复合文本

更多优化建议

[Google : PageSpeed Insights规则](#)

[Yahoo : Best Practices for Speeding Up Your Web Site](#)