

关于 Front End Dev 你可能需要知道的事情

About me

朱琳

From: 前端组

1. 前端定义
2. Web标准

Q1: HTML5到底指的是什么？

3. 浏览器

Q2: 为什么会有兼容性问题？

4. 浏览器之外

Q3: 现在的html+css+js可以做什么？

5. 工作流

1. 前端定义

2. Web标准

Q1: HTML5到底指的是什么？

3. 浏览器

Q2: 为什么会有兼容性问题？

4. 浏览器之外

Q3: 现在的html+css+js可以做什么？

5. 工作流

Front End

In [software engineering](#), the terms "**front end**" and "**back end**" are distinctions which refer to the separation of concerns between a presentation layer and a data access layer respectively.

—[wikipedia](#)

(Web) Front End (Dev)

HTML + CSS + javascript

开发APP有三种方式

- 原生应用 (Native App)
 - React Native和phoneGap也算作这一类？
- 网页应用 (Web App)
- 混合应用 (Hybrid App)

Native

- Android: 纯Java ; IOS: 纯Object-C...
 - PC, Mac, linux...
-
- Pros : 速度快性能佳 可用接口多
 - Cons: 不够灵活 发布慢

Native



Web

- html + css + javascript
 - 浏览器访问
-
- Pros : 灵活 更新发布快
 - Cons: 性能瓶颈, 接口少, 寄托在浏览器内

Web App

A screenshot of a dark-themed web application interface. At the top left is a user profile icon with the name 'zhulin'. Below it is a navigation bar with icons for '+' (New), '上传' (Upload), and '导入笔记' (Import Notes). On the left side, there's a sidebar with a list of groups: 'LUNA看天下', '有道早早餐群智慧集锦', '家居讨论群', '技能交换', '测试dreamfox测试dream...', '直播课项目', 'test', and '助手'. At the bottom, there are links for 'Mac' and 'Android'.

LUNA看天下			
文件名	修改时间	最近修改人	大小
各种文档	2014-09-17 14:31	陈虎	
3、高保真	2014-08-25 18:36	吴宇峰	
2、原型图	2014-08-20 11:10	吴宇峰	
1、backlog	2014-07-24 15:36	李斌	
4、会议记录	2014-06-24 14:17	吴宇峰	
cassandra统计项目.note	2014-08-15 15:54	杨哲	6.88 KB

A screenshot of a Youdao search result page on a mobile browser. The URL in the address bar is m.youdao.com/dict?le=... . The search term 'youdao' is entered in the search bar. The results are categorized under '基本释义' (Basic Definition), which includes the definition 'n. 有道 (网易公司的搜索服务)'. Other categories shown are '英英释义' (English-English Definition), '网络释义' (Network Definition), '双语例句' (Bilingual Examples), and '百科' (Encyclopedia). At the bottom, there's an advertisement for '有道词典-市场占有率第一的词典，立即下载' (Youdao Dictionary - China's No. 1 dictionary, download now).

slide=id.g476ff858c...

有道 youdao 词典 翻译 笔记 惠惠 网页

中英 youdao

基本释义 ^

youdao

n. 有道 (网易公司的搜索服务)

英英释义 ▼

网络释义 ▼

双语例句 ▼

百科 ▼

有道词典-市场占有率第一的词典，立即下载

有道首页 | 反馈意见 | 切换到PC版
©2014 公司 京ICP证080268号

◀ 前词 | 后词 ▶



参考价: ¥4450.00

惠 下降

其他8家报价

收藏

商品评分: ★★

配送至: 北京

服务: 由京东

易迅商城 ¥3219

京东商城 ¥3262

唯品会 ¥3500

国美在线 ¥2999

苏宁易购 ¥3204

1号店 ¥3204

当当网 ¥3337

淘宝网 ¥1692

联系客服

5:00前完

夜间配

男表B

表B

网带 男表E

男表E

选择颜色:



更多报价>>

手机也能比价啦! 点我下载>>

个护化妆

鞋靴、箱包、钟表、奢

运动户外

汽车用品

母婴、玩具乐器

食品饮料、酒类、生鲜

网易出品 保健

惠惠
购物助手

超值购(60)

我的收藏

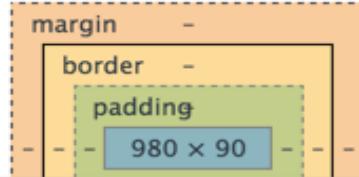
```
<!DOCTYPE html>
▼ <html>
  ▶ <head>...</head>
  ▼ <body>
    ▼ <div id="w">
      ▶ <div id="n">...</div>
      <div class="sp1"></div>
      ▶ <div id="x" class="w">...</div>
      <div class="sp2"></div>
      ▶ <div id="sb" class="w">...</div>
      <div class="sp3"></div>
      ▶ <div id="f" class="w">...</div>
    </div>
    ▶ <div id="snv" class="clear">...</div>
    <script type="text/javascript" src="http://shared.ydstatic.com/images/myth/js/index_min.js?201311221"></script>
    ▶ <script type="text/javascript">...</script>
    ▶ <div id="youdaoDictChromeExtension" style="display: none;">...</div>
  </body>
```

html body div#w div.sp1

Styles Event Listeners DOM Breakpoints Properties

element.style {
}

@media screen and (max-height:
10000px) and (min-height: 768px)
.sp1 {
 height: 90px;
}



Find in Styles

Filter

Console Search Emulation Rendering

Device

Model: Apple iPhone 5

Media

Resolution:

320

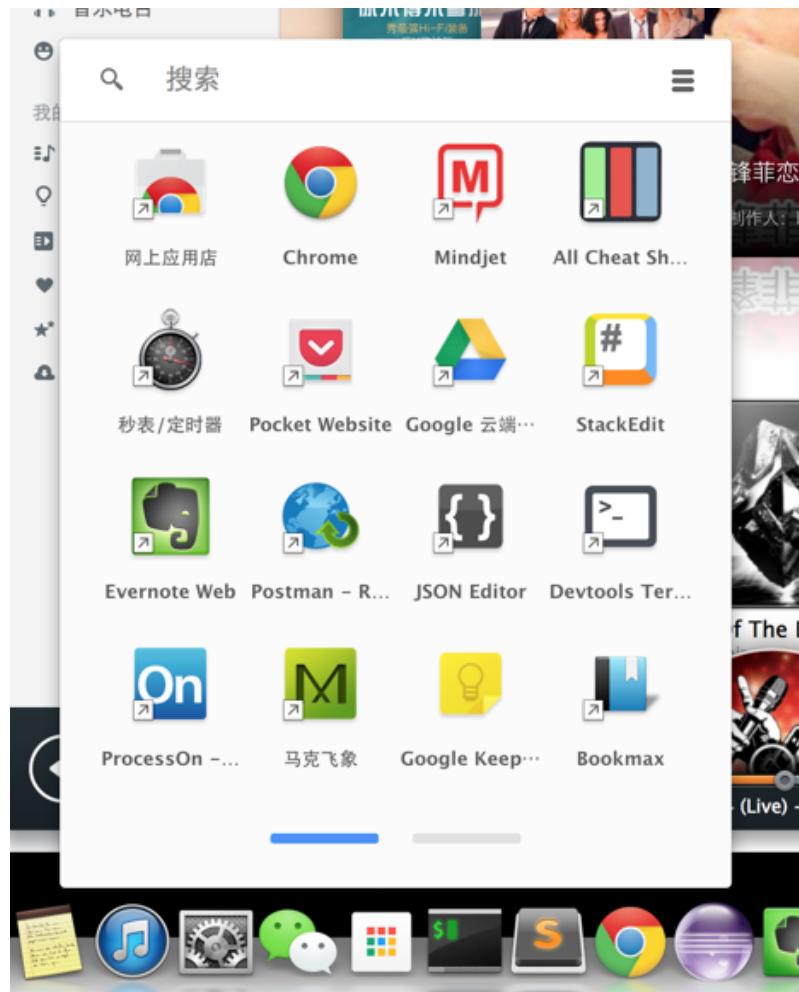


568

Network

Device pixel ratio:

?



hybrid

web + native

- Pros: 兼顾双方优点
- Cons: debug比较难

Hybrid

有道词典 词典 例句 百科 翻译 mini - □ ×

En / 输入要查询的单词或词组



It is not easy to meet each other in such a big world.
世界这么大，能遇见，不容易。 [更多 ›](#)

满意度



双语阅读
越南人为什么喜欢吃猫?
Vietnam's taste



学好英语，速度海淘
海淘大集合
All about Online Shopping



以假乱真的3D铅笔画(多图)
眼睛又被骗！



科学有态度
为了以后K歌时，能对得起大家的耳朵

✓ 取词 | ✓ 划词

中国大学MOOC, 37所985高校 秋学期开课啦!

有道云笔记·协作

LUNA看天下

有道云协作产品近况

新建 | 上传 | 导入笔记

全部文件

文件名 | 修改时间

各种文档 2014-09-17 14:30

3、高保真 2014-08-25 18:10

2、原型图 2014-08-20 11:10

1、backlog 2014-07-24 15:10

4、会议记录 2014-06-24 14:10

词典安卓客户端sprint流程的建议 by yangzhe.note 2014-09-22 23:10

搜索文件和讨论

24

杨哲 修改了笔记 22:03

词典安卓客户端sprint流程的建议 by yangzhe.note
第 8 版

查看 打开文件夹

杨哲 修改了笔记 22:19

词典安卓客户端sprint流程的建议 by yangzhe.note
第 9 版

查看 打开文件夹

@群成员或直接输入...

发送

GRE词汇



GRE词汇

词汇量: 7318



乱序背诵全部词汇，遵循记忆曲线，科学记单词，从背过到熟记，层层深入至掌握

背单词

按顺序浏览单词，提供单词释义、发音、例句、用法、真题，全方位助你逐个击破

词汇表

*在书内仍可通过右上角的按钮，在两种功能之间切换



上班刷微信？NO！学英语才是正经事

每天坚持5分钟，
30天后在老板面前自信表达

GRE词汇

背单词

list 1

50

list 2

50

list 3

50

list 4

50

list 5

50

list 6

50

list 7

50

list 8

50

list 9

50

list 10

50

list 11

50

list 12

50



上班刷微信？NO！学英语才是正经事

每天坚持5分钟，
30天后在老板面前自信表达

已背 0 / 未背 7318

explicable

英 [ɪk'splɪkəb(ə)l; ek-; 'eksplɪ,kəb(ə)l]

美 [ɪk'splɪkəbl]

显示释义



没记住



记住了



上班刷微信？NO！学英语才是正经事

每天坚持5分钟，
30天后在老板面前自信表达

1. 前端定义

2. Web标准

Q1: HTML5到底指的是什么？

3. 浏览器

Q2: 为什么会有兼容性问题？

4. 浏览器之外

Q3: 现在的html+css+js可以做什么？

5. 工作流

1. 推进WEB开发的三方
2. 万维网联盟[World Wide Web Consortium]
 - 成员 / 使命 / 标准推进的过程
3. HTML相关标准
 - HTML4 / HTML5 / DOM
4. CSS相关标准
 - CSS2.1 / CSS3系列
5. JS相关标准
 - ECMAScript / DOM / Commonjs / AMD

开发者们

制定标准的



浏览器厂商



TECHNICAL REPORTS

By date

By group

WEB AND INDUSTRY

Web of Data

Digital Publishing

Mobile Web

TV and Broadcasting

WEB FOR ALL

W3C Turns 20

1 October 2014 | [Archive](#)



This month W3C celebrates its 20th anniversary. In a [1994 press release about the launch of W3C](#), Director [Tim Berners-Lee](#) explained, "The decision to form the consortium came at the urging of many firms investing increasing

resources into the Web, whether in creation of software products, selling information, or for sharing information within their own companies, with business partners and the public at large." Twenty years on, the W3C community is actively building an Open Web Platform for application development that has the unprecedented potential to enable developers to build sophisticated interactive experiences that are available on any device. Pursuing a vision of one Web available to all, [400 Members](#), thousands of individuals in [Working and Interest Groups](#), nearly 4500 people in [Community and Business Groups](#), and many more subscribed to public mailing lists power today's W3C community. W3C thanks all of you for 20 years of creating, implementing, and supporting [open standards](#) for the Web.



Microsoft



Google



YAHOO!

Baidu 百度



UC浏览器

Tencent 腾讯 | 一切以用户价值为依归



*“the W3C community is actively building an Open Web Platform for application development that has the unprecedented potential to enable developers to build **sophisticated interactive experiences that are available on any device**. Pursuing a vision of **one Web available to all**, 400 Members, thousands of individuals in **Working and Interest Groups**, nearly 4500 people in **Community and Business Groups**, and many more subscribed to public mailing lists power today’s W3C community. W3C thanks all of you for 20 years of creating, implementing, and supporting **open standards for the Web**.”*

The World Wide Web Consortium (W3C) is an international community that develops open standards to ensure the long-term growth of the Web.

Web Standard

1. W3C publishes documents that define Web technologies.
2. These documents follow a **process** designed to promote consensus, fairness, public accountability, and quality.
3. At the end of this process, W3C publishes **Recommendations**, which are considered Web standards.

标准发布流程

PD – Public Draft, 公共草案

初步想法。不乏 PD 之后就废弃的。

WD – Working Draft, 工作草案。

发布给社区/团体 review 的文档，这些社区/团体包括：W3C 成员，公众和其它一些技术组织。通常情况下，工作草案意味着将被发展为标准，当然，也有中途夭折的。

LC – Last Call, 对外宣布

CR – Candidate Recommendation, 候选推荐。

W3C 相信此阶段的文档已经经过广泛的 review，并能满足 WG 工程技术需求。发布候选版本用于收集技术可行性建议

PR – Proposed Recommendation, 提议

经过技术安全性和可行性的广泛 review 后，提交到 W3C 咨询/顾问委员会做最后的批准

REC – Recommendation, 正式推荐。也就是正式版 标准。

推荐：在达成广泛的共识，收到 W3C 成员和理事的批复后，开始大规模的推荐/部署

详细参考：<http://www.w3.org/2005/10/Process-20051014/tr#maturity-levels>

ATTENTION !

有标准不代表立即有实现。

标准制订跟不上发展，浏览器厂商在竞争中冒进，苦逼的是开发者。

标准处于草案阶段也可能已经有浏览器支持

浏览器厂商从自身利益出发，选择性的支持了一些仍处于WD状态的标准。于是出现各种厂商前缀。

```
.box {  
    -webkit-border-radius: 10px;  
    -moz-border-radius: 10px;  
    -ms-border-radius: 10px;  
    border-radius: 10px;  
}
```

4 Syntax and basic data types

Contents

4.1 Syntax

4.1.1 Tokenization

4.1.2 Keywords

4.1.2.1 Vendor-specific extensions

4.1.2.2 Informative Historical Notes

4.1.3 Characters and case

4.1.4 Statements

4.1.5 At-rules

4.1.2.2 Informative Historical Notes

This section is informative.

At the time of writing, the following prefixes are known to exist:

prefix	organization
-ms-, mso-	Microsoft
-moz-	Mozilla
-o-, -xv-	Opera Software
-atsc-	Advanced Television Standards Committee
-wap-	The WAP Forum
-khtml-	KDE
-webkit-	Apple
prince-	YesLogic
-ah-	Antenna House
-hp-	Hewlett Packard
-ro-	Real Objects
-rim-	Research In Motion
-tc-	TallComponents

Web Design and Applications involve the standards for building and Rendering Web pages, including HTML, CSS, SVG, device APIs, and other technologies for Web Applications ("Web"). It also includes information on how to make pages accessible to people with disabilities (WCAG), to internationalize them, and make them work on mobile devices.

HTML & CSS

HTML and CSS are the fundamental technologies for building Web pages: HTML (html and xhtml) for structure, CSS for style and layout, including WebFonts. Find resources for good Web page design as well as helpful tools.

Audio and Video

Some of the W3C formats that enable authoring audio and video presentations include HTML, SVG, and SMIL (for synchronization). W3C is also working on a timed text format for captioning and other applications.

Mobile Web

W3C promotes "One Web" that is available on any device. W3C's Mobile Web Best Practices help authors understand how to create content that provides a reasonable experience on a wide variety of devices, contexts, and locations.

JavaScript Web APIs

Standard APIs for client-side Web Application development include those for Geolocation, XMLHttpRequest, and mobile widgets. W3C standards for document models (the "DOM") and technologies such as XBL allow content providers to create interactive documents through scripting.

Accessibility

W3C's Web Accessibility Initiative (WAI) has published Web Content Accessibility Guidelines (WCAG) to help authors create content that is accessible to people with disabilities. WAI-ARIA gives authors more tools to create accessible Web Applications by providing additional semantics about widgets and behaviors.

Privacy

The Web is a powerful tool for communications and transactions of all sorts. It is important to consider privacy and security implications of the Web as part of technology design. Learn more about tracking and Web App security.

Graphics

W3C is the home of the widely deployed SVG vector format, and the Canvas API, a more specialized format used, for example, in automotive engineering, aeronautics.

Internationalization

W3C has a mission to design technologies that work across cultures and languages. W3C standards such as XML are built on Unicode, for instance. W3C has published guidance for authors relating to bi-directional (bidi) text, and more.

Math on the Web

Mathematics and formulae are used on the Web in reports, education materials and scientific publications. MathML enables mathematics to be semantically processed on the World Wide Web, just as HTML has enabled this functionality for other types of content.

HTML

超文本标记语言(英文:HyperText Markup Language, HTML)是为“网页创建和其它可在网页浏览器中看到的信息”设计的一种标记语言。

HTML被用来结构化信息——例如标题、段落和列表等等，也可用来在一定程度上描述文档的外观和语义。



HTML 4.01 Specification

W3C Recommendation 24 December 1999

This version:

<http://www.w3.org/TR/1999/REC-html401-19991224>

(plain text [794Kb], gzip'ed tar archive of HTML files [371Kb], a .zip archive of HTML files [405Kb], gzip'ed Postscript file [963Kb])

Latest version of HTML 4.01:

<http://www.w3.org/TR/html401>

Latest version of HTML 4:

<http://www.w3.org/TR/html4>

Latest version of HTML:

<http://www.w3.org/TR/html>

Previous version of HTML 4.01:

<http://www.w3.org/TR/1999/PR-html40-19990824>

Previous HTML 4 Recommendation:

<http://www.w3.org/TR/1998/REC-html40-19980424>

Editors:

[Dave Raggett <dsr@w3.org>](mailto:dsr@w3.org)

Arnaud Le Hors, W3C

Ian Jacobs, W3C

History

1. 超文本标记语言(第一版)——在1993年6月发为互联网工程工作小组(IETF)工作草案发布(并非标准).
2. HTML 2.0——1995年11月作为【RFC1866】发布, 在【RFC2854】于2000年6月发布之后被宣布已经过时。
3. HTML 3.2——1997年1月14日, W3C推荐标准
4. HTML 4.0——1997年12月18日, W3C推荐标准 [<http://www.w3.org/TR/html4/>]
5. **HTML 4.01(微小改进)——1999年12月24日, W3C推荐标准**
6. HTML 5草案——在2008年1月22日, 第一份正式草案发布
7. HTML 5规范——在2012年12月17日, 第一份正式规范发布 w3c小组发布了正式的html5和canvans 2D版本的规范。
8. **HTML 5推荐标准(PR)**——2014年九月, HTML5进入推荐标准阶段

插个八卦

The Web Hypertext Application
Technology Working Group



Abbreviation WHATWG

Motto Maintaining and evolving
HTML since 2004

Formation June 4, 2004; 10 years ago

Spokesman Ian "Hixie" Hickson

Main organ Membership

Website whatwg.org ↗

HTML

Living Standard – Last Updated 14 October 2014



HTML5

A vocabulary and associated APIs for HTML and XHTML

W3C Proposed Recommendation 16 September 2014

This Version:

<http://www.w3.org/TR/2014/PR-html5-20140916/>

Latest Published Version:

<http://www.w3.org/TR/html5/>

Latest Version of HTML:

<http://www.w3.org/TR/html/>

Latest Editor's Draft:

<http://www.w3.org/html/wg/drafts/html/CR/>

Previous Version:

<http://www.w3.org/TR/2014/CR-html5-20140731/>

Editors:

WHATWG:

[Ian Hickson](#), Google, Inc.

W3C:

[Robin Berjon](#), W3C

[Steve Faulkner](#), The Paciello Group

[Travis Leithead](#), Microsoft Corporation

HTML5

- 旨在提高 HTML 的交互行为, 支持当前多样的, 复杂的 Web 内容
- 不仅描述了如何使用 HTML 标记文档, 还列出了对 DOM 的变更并添加了特定的 Javascript 扩展
- 目标: 取代 1999 年所制定的 HTML 4.01 和 XHTML 1.0 标准, 以期能在互联网应用迅速发展的时候, 使 网络标准达到符合当代的网络需求。
- 添加了许多新的语法特征, 其中包括 `<video>`, `<audio>`, 和 `<canvas>` 元素, 同时集成了 SVG 内容。这些元素是为了更容易的在网页中添加和处理多媒体和图片内容而添加的。其它新的元素包括 `section`, `article`, `header`, 和 `nav`, 是为了丰富文档的数据内容

CSS = Cascading Style Sheets

样式来源

- 用户自定义样式



- 浏览器用户自己设置的css文件；

- 网页样式

- 内联样式 (eg : <p style="color:red"></p>)

- 外部样式表 (`<link rel="stylesheet" href="http://shared.ydstatic.com/f...`

- 内部样式表

- 浏览器默认样式

- 各浏览器会有个默认的样式



```
<style id="style-1-cropbar-clipper">
    /* Copyright 2014 Evernote Corporation. All rights reserved. */
    .en-markup-crop-options {
        top: 18px !important;
        left: 50% !important;
        margin-left: -100px !important;
        width: 200px !important;
        border: 2px rgba(255,255,255,.38) solid !important;
        border-radius: 4px !important;
    }
    .en-markup-crop-options div div:first-of-type {
        margin-left: 0px !important;
    }

```

TABLE OF SPECIFICATIONS

Ordered from most to least stable:

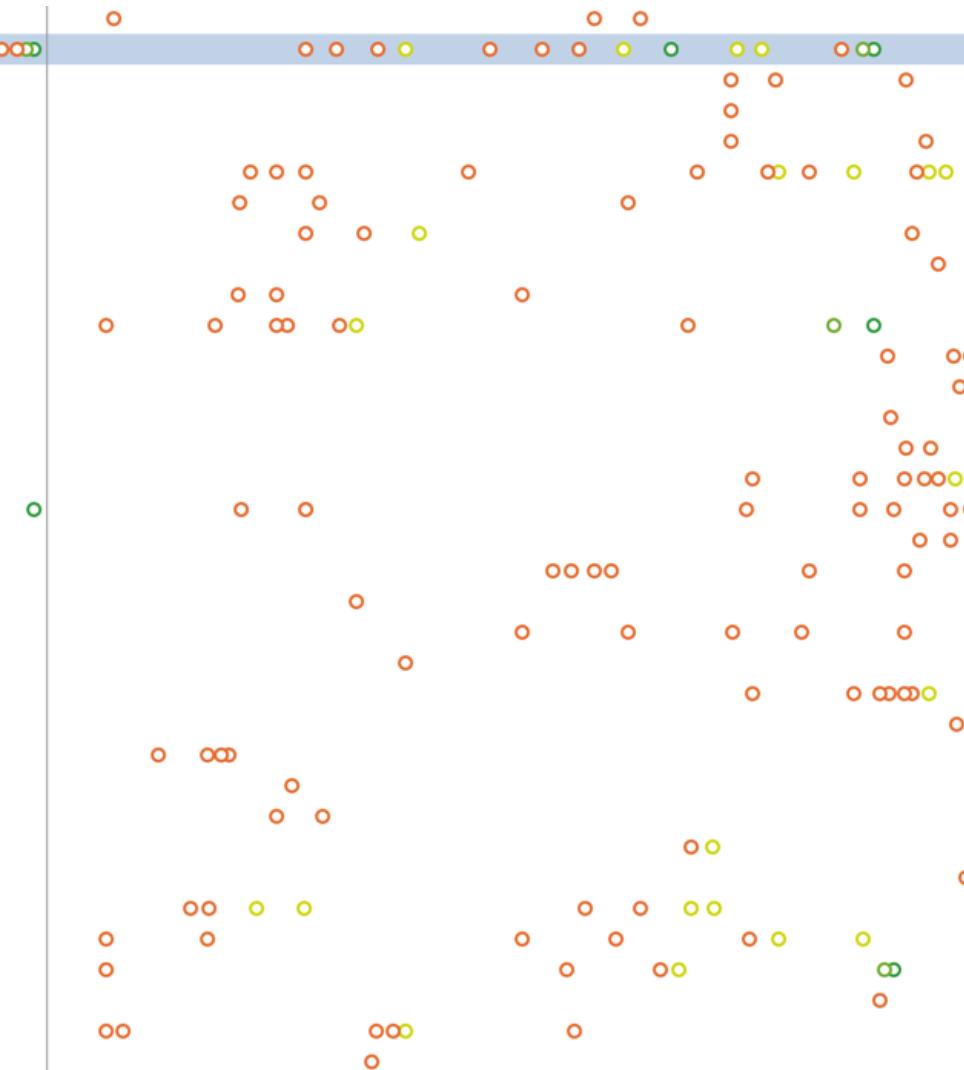
Completed	Current	Upcoming	Notes	
CSS Snapshot 2010	NOTE		Latest stable CSS	i
CSS Snapshot 2007	NOTE			i
CSS Color Level 3	REC	REC	See Errata	i
CSS Namespaces	REC	REC		i
Selectors Level 3	REC	REC		i
CSS Level 2 Revision 1	REC	REC	See Errata	i
CSS Level 1	REC		Unmaintained, see Snapshot	i
CSS Print Profile	NOTE			i
Media Queries	REC	REC		i
CSS Style Attributes	REC	REC		i

Testing	Current	Upcoming	Notes	
CSS Backgrounds and Borders Level 3	CR	PR		i
CSS Conditional Rules Level 3	CR	CR		i
CSS Image Values and Replaced Content Level 3	CR	PR		i
CSS Multi-column Layout	CR	CR		i
CSS Speech	CR	PR		i
CSS Values and Units Level 3	CR	PR		i
CSS Flexible Box Layout	LG	CR		i

CSS3

f2e CSS3概述#模块

1998-8-20



1. Behavioral Extensions to CSS

2. CSS 2.1

3. CSS 2D Transformations Module

4. CSS 3D Transformations Module

5. CSS Animations Module

6. CSS Backgrounds and Borders level 3

7. CSS Basic Box Model

8. CSS Basic User Interface

9. CSS Box Alignment Module Level 3

10. CSS Cascading and Inheritance

11. CSS Color

12. CSS Conditional Rules

13. CSS Counter Styles Level 3

14. CSS Device Adaptation

15. CSS Exclusions and Shapes

16. CSS Flexible Box Layout Level 3

17. CSS Fonts

18. CSS Fragmentation Module Level 3

19. CSS Generated Content for Paged Media

20. CSS Generated and Replaced Content

21. CSS Grid Template Layout

22. CSS Hyperlink Presentation

23. CSS Image Values and Replaced Content Module Level 3

24. CSS Intrinsic & Extrinsic Sizing Module Level 3

25. CSS Introduction

26. CSS Line Layout

27. CSS Lists

28. CSS Marquee

29. CSS Masking

30. CSS Mobile Profile

31. CSS Multi-column Layout

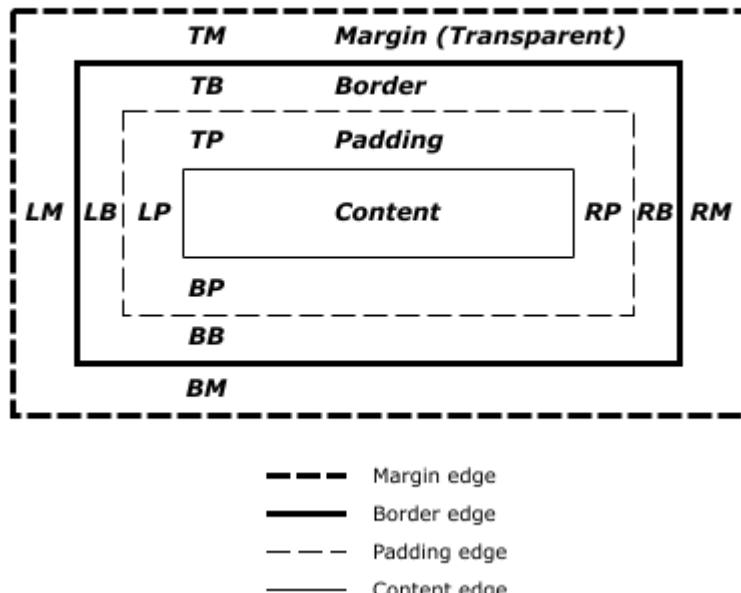
32. CSS Namespaces

33. CSS Object Model

34. CSS Paged Media

35. CSS Presentation Levels

Example: Box Model



Quick Table of Contents

- [1 About the CSS 2.1 Specification](#)
- [2 Introduction to CSS 2.1](#)
- [3 Conformance: Requirements and Recommendations](#)
- [4 Syntax and basic data types](#)
- [5 Selectors](#)
- [6 Assigning property values, Cascading, and Inheritance](#)
- [7 Media types](#)
- [8 Box model](#)
- [9 Visual formatting model](#)
- [10 Visual formatting model details](#)
- [11 Visual effects](#)
- [12 Generated content, automatic numbering, and lists](#)
- [13 Paged media](#)
- [14 Colors and Backgrounds](#)
- [15 Fonts](#)
- [16 Text](#)
- [17 Tables](#)
- [18 User interface](#)
- [Appendix A Aural style sheets](#)

火狐的3D view Demo

Javascript 相关



INTERNATIONAL

Standard ECMA-262

5.1 Edition / June 2011

ECMAScript Language Specification

- 1996年11月 网景通讯公司将JavaScript提交给欧洲计算机制造商协会进行标准化。
- 1997年06月 ECMA-262的第一个版本被欧洲计算机制造商协会采纳。并将ECMA-262 标准取名为 ECMAScript。
- 1998年06月 第二个版本：格式修正，以使得其形式与ISO/IEC16262国际标准一致
- **1999年12月 第三版** 强大的正则表达式，更好的文字链处理，新的控制指令，异常处理，错误定义更加明确，数输出的格式化及其它改变
- 第四版废弃；
- **2009年12月 第五版**发布：实际上是v3.1
- 2008年8月 原先的第四版重命名为**EcmaScript Harmony, 也就是第六版**

DOM(Document Object Model)

Completed Work

2008-12-22	Element Traversal Specification	Recommendation
2004-04-07	Document Object Model (DOM) Level 3 Load and Save Specification	Recommendation
2004-04-07	Document Object Model (DOM) Level 3 Core Specification	Recommendation
2004-01-27	Document Object Model (DOM) Level 3 Validation Specification	Recommendation
2003-01-09	Document Object Model (DOM) Level 2 HTML Specification	Recommendation
2000-11-13	Document Object Model (DOM) Level 2 Core Specification	Recommendation
2000-11-13	Document Object Model (DOM) Level 2 Events Specification	Recommendation
2000-11-13	Document Object Model (DOM) Level 2 Style Specification	Recommendation
2000-11-13	Document Object Model (DOM) Level 2 Traversal and Range Specification	Recommendation
2000-11-13	Document Object Model (DOM) Level 2 Views Specification	Recommendation
1998-10-01	Document Object Model (DOM) Level 1	Recommendation
2004-02-26	Document Object Model (DOM) Requirements	Group Note
2004-02-26	Document Object Model (DOM) Level 3 Views and Formatting Specification	Group Note
2004-02-26	Document Object Model (DOM) Level 3 XPath Specification	Group Note
2002-07-25	Document Object Model (DOM) Level 3 Abstract Schemas Specification	Group Note

浏览器提供的其他内置对象

- **Browser Object Model (BOM)**
- **Device APIs :**
 - Ambient Light Sensor API, Battery Status API, Geolocation API, Pointer Lock API, Proximity API, Device Orientation API, Screen Orientation API, Vibration API.....
- **Communication APIs :**
 - Network Information API, Web Notifications, Simple Push API.
- **Data management APIs :**
 - FileHandle API, IndexedDB.

Current Status of Specifications

Learn more about the current status of specifications related to:

- [Accessible Rich Internet Applications \(WAI-ARIA\)](#)
- [DOM](#)
- [DOM events](#)
- [Declarative Web Applications](#)
- [Javascript APIs](#)
- [Mobile Web Applications](#)
- [WICD](#)
- [Widgets](#)
- [XBL](#)
- [HTML](#)
- [Web Components](#)
- [Web Performance](#)
- [Web Applications Manifest Formats](#)

These [W3C Groups](#) are working on the related specifications:

- [HTML Working Group](#)
- [Web Applications Working Group](#)
- [Audio Working Group](#)
- [Browser Testing and Tools Working Group](#)
- [Cascading Style Sheets \(CSS\) Working Group](#)
- [Device APIs Working Group](#)

JAVASCRIPT WEB APIS

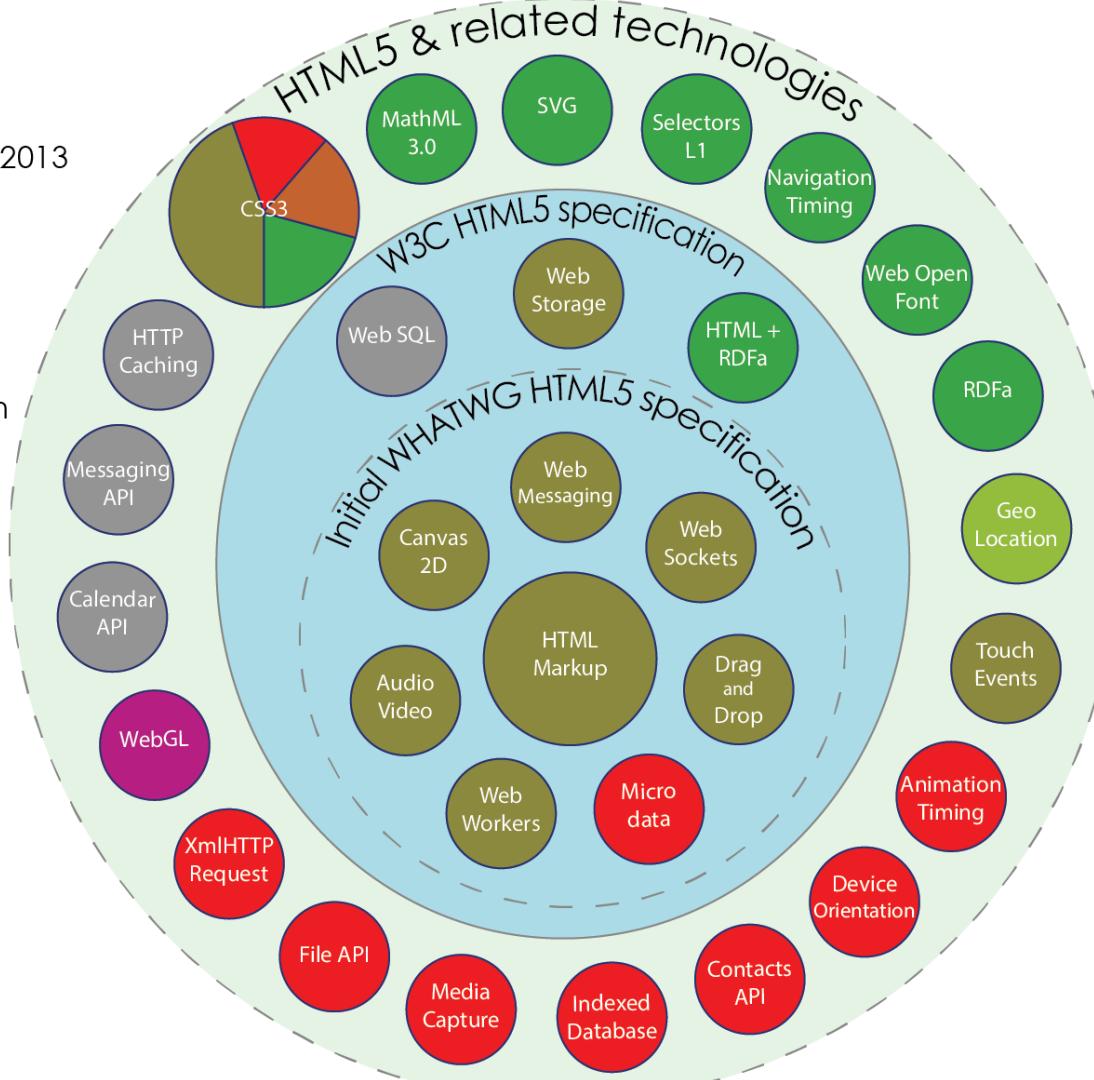
On this page → [what is scripting](#) • [what scripti](#)

Q1: HTML5到底指的是什么？

HTML5

Taxonomy & Status on January 20, 2013

- W3C Recommendation
- Proposed Recommendation
- Candidate Recommendation
- Last Call
- Working Draft
- Non-W3C Specifications
- Deprecated



CSS3

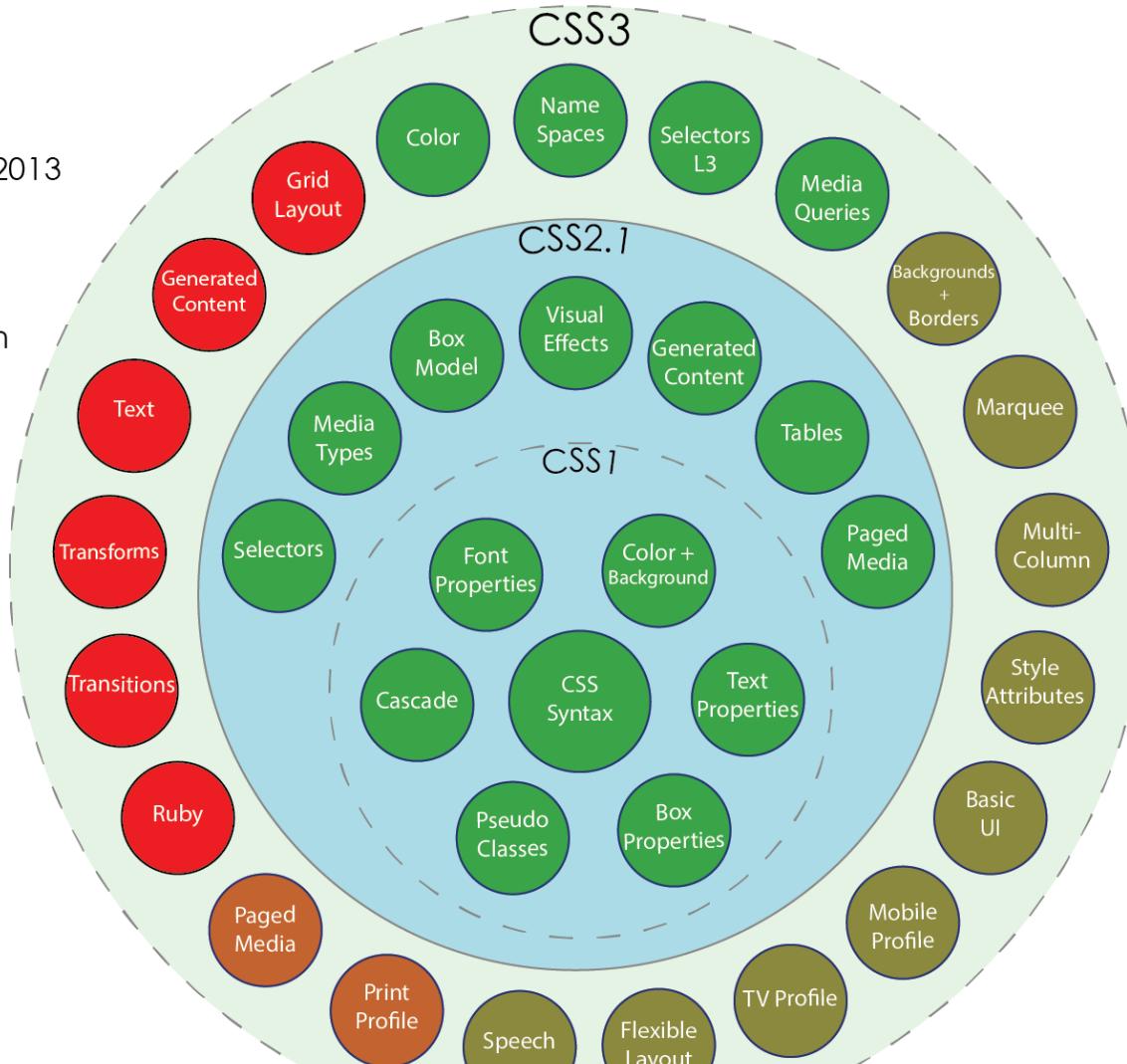
Taxonomy & Status on January 20 2013

 W3C Recommendation

 Candidate Recommendation

 Last Call

 Working Draft



1. 前端定义
2. Web标准

Q1: HTML5到底指的是什么？

3. 浏览器

Q2: 为什么会有兼容性问题？

4. 浏览器之外

Q3: 现在的html+css+js可以做什么？

5. 工作流

1. 浏览器内核

- Webkit / Gecko / Trident

2. 开源项目

- Webkit (blink) / V8 / Chromium

3. 工作原理梗概

- Q: 为什么css要放在头部, js要放在底部?

4. 浏览器提供的其他API概览

浏览器内核

Internet Explorer	Trident
Chrome, Safari, Android, Chrome OS, Opera.... (国内各山寨浏览器)	Webkit (或者说Blink) + V8 / nitro 等等
Firefox, Firefox OS	Gecko + SpiderMonkey , and Rhino

Application	Dialect and latest version	ECMAScript
Google Chrome, the V8 engine	JavaScript	ECMA-262/5
Mozilla Firefox, the Gecko layout engine, SpiderMonkey, and Rhino	JavaScript 1.8.5	ECMA-262/5
Safari, the Nitro engine	JavaScript	ECMA-262/5.1
Opera	ECMAScript with some JavaScript 1.5 and JScript extensions	ECMA-262/3
KHTML layout engine,KDE's Konqueror	JavaScript 1.5	ECMA-262-3
Adobe Acrobat	JavaScript 1.5	ECMA-262-3
OpenLaszlo	JavaScript 1.4	ECMA-262/3
Max/MSP	JavaScript 1.5	ECMA-262/3
ANT Galio 3	JavaScript 1.5 with RMAI extensions	ECMA-262/3

03年 Apple发布Safari

05年 Apple发起开源项目 [Webkit](#)

08年 V8引擎与Chrome 1.0版本同时发布

09年 Google将Webkit纳入Chromium项目

14年 Google和Opera开发Blink, 计划将其作为chromium的一部分

Webkit

WebCore WebCore是一个由WebKit专案所开发的布局(Layout)、渲染(Rendering)及HTML和SVG的DOM函式库, 完整的程式码皆由[GNU通用公共许可证](#)所授权, WebKit框架包装了WebCore及JavaScriptCore, 并提供一个Objective-C[应用程序接口](#)来接介由C++所开发的WebCore渲染引擎及JavaScriptCore脚本引擎, 透过[Cocoa API](#)就可以在应用程式中很简单的使用这些元件。之后的版本同时包含了一个[跨平台](#)的C++抽象平台, 并且提供各种API使用。

JavaScriptCore (google替換成了V8)

Web Inspector

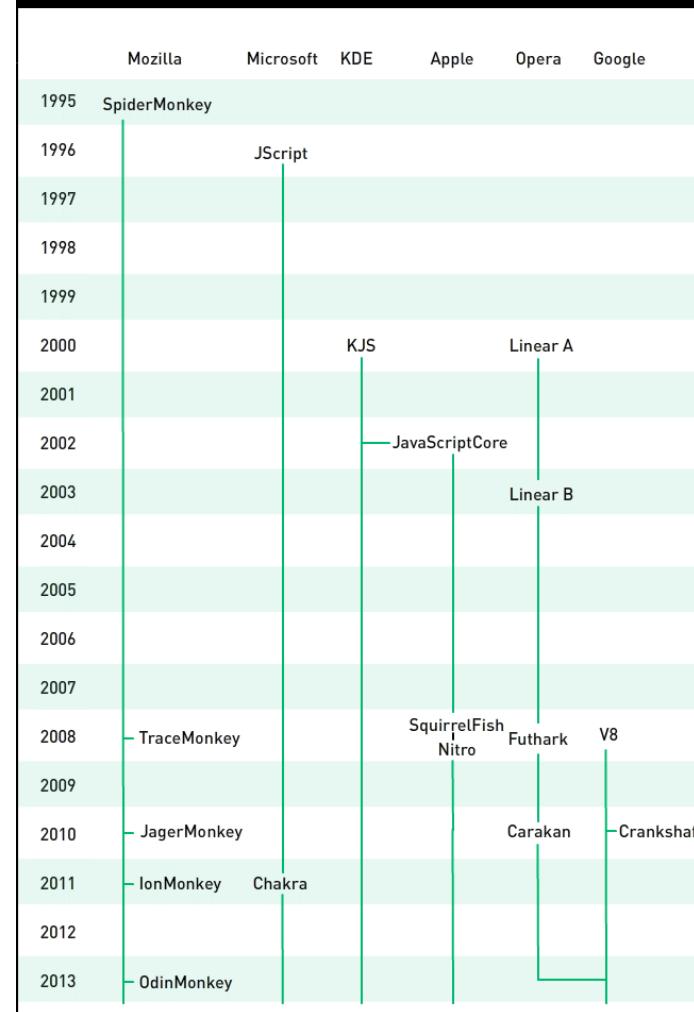
V8 JavaScript Engine

- Google家的Js引擎
- 实现ECMA-262第五版; 可运行于 Windows (XP or newer), Mac OS X (10.5 or newer), and Linux systems
- 运行之前将JavaScript编译成了机器码, 而非字节码或是解释执行它, 以此提升性能。
- 更进一步, 使用了如内联缓存(inline caching)等方法来提高性能。有了这些功能, JavaScript程序与V8引擎的速度媲美二进制编译。
- Nodejs的基础



JavaScript Engine Family Tree

插花:JS引擎历史



Chromium

- Google主导开发的网页浏览器，以BSD许可证等多重自由版权发布并开放源代码。
- 在架构上使用了苹果发展出来的WebKit排版引擎(自28版起改为由WebKit所分支的Blink排版引擎)、Safari的部份源代码与Firefox的成果，并采用Google独家开发出的V8引擎以提升解译JavaScript的效率，
- 设计了“沙盒”、“黑名单”、“无痕浏览”等功能来实现稳定与安全的网页浏览环境。
- 所以Chromium相当于Google Chrome的工程版或称实验版
- Chromium的更新速度很快，每隔数小时即有新的开发版本发布
- Google Chrome是基于Chromium制造，但包含非开放源代码套件，主要是多媒体相关。



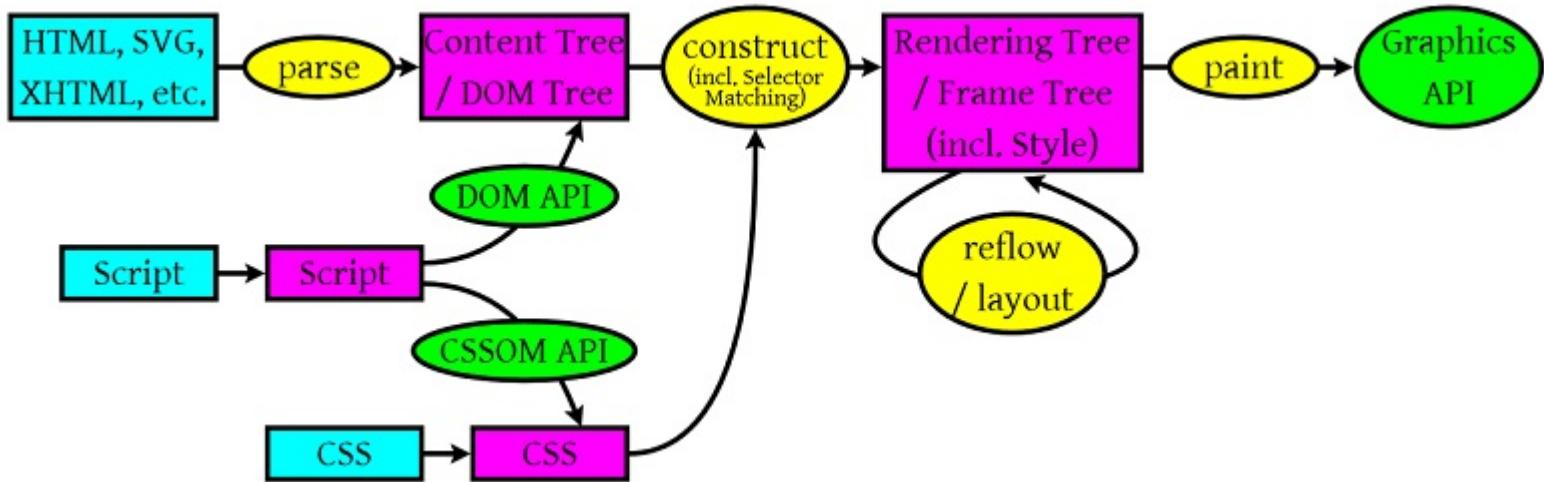
浏览器工作原理

Browsers' job description

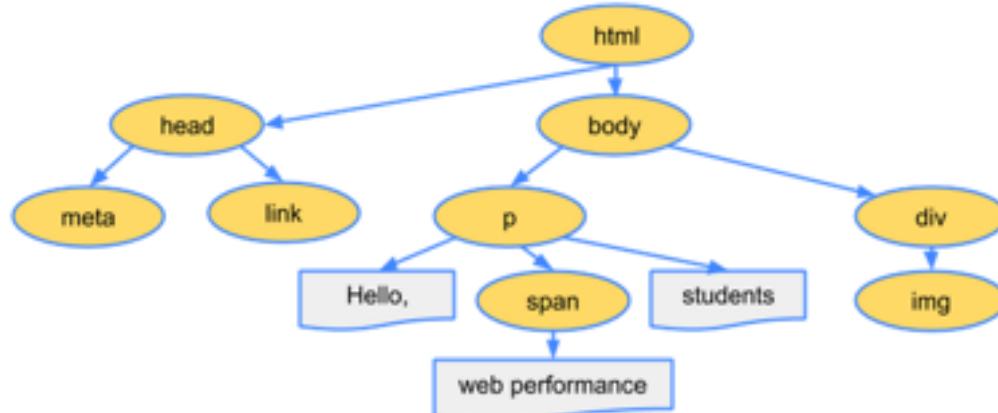
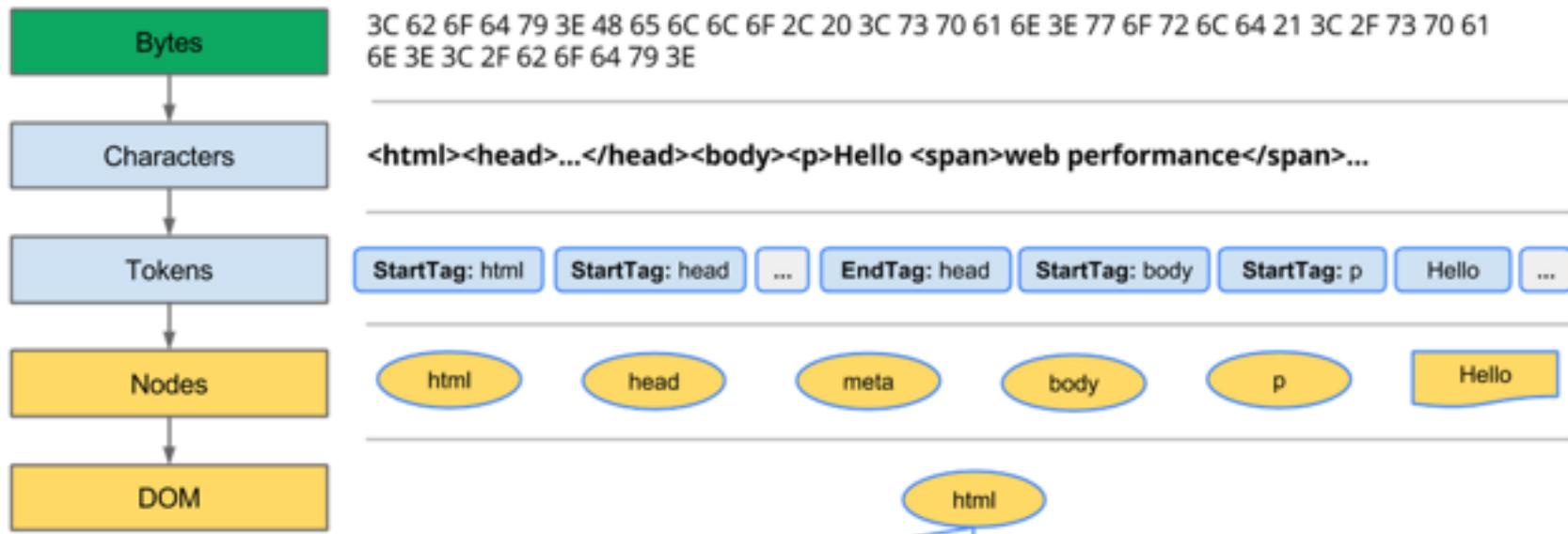
输入URL, 获取资源, 转化为图片推送到屏幕

webkit组成

- ❖ HTML parser: HTML转成DOM Tree
- ❖ CSS parser: CSS转成 CSSOM tree
- ❖ layout : 计算布局
- ❖ javascript engine : 运行js



```
<html>
  <head>
    <meta name="viewport" content="width=device-width,initial-scale=1.0">
    <link href="style.css" rel="stylesheet">
    <title>Critical Path</title>
  </head>
  <body>
    <p>Hello <span>web performance</span> students!</p>
    <div></div>
  </body>
</html>
```



1. Bytes → characters → tokens → nodes → object model.
2. HTML标记语言生成**Document Object Model (DOM)**,

```
body { font-size: 16px }
p { font-weight: bold }
span { color: red }
p span { display: none }
img { float: right }
```

插花1：css标准

5 Selectors

Contents

- [5.1 Pattern matching](#)
- [5.2 Selector syntax
 - \[5.2.1 Grouping\]\(#\)](#)
- [5.3 Universal selector](#)
- [5.4 Type selectors](#)
- [5.5 Descendant selectors](#)
- [5.6 Child selectors](#)
- [5.7 Adjacent sibling selectors](#)
- [5.8 Attribute selectors
 - \[5.8.1 Matching attributes and attribute values\]\(#\)
 - \[5.8.2 Default attribute values in DTDs\]\(#\)
 - \[5.8.3 Class selectors\]\(#\)](#)
- [5.9 ID selectors](#)
- [5.10 Pseudo-elements and pseudo-classes](#)
- [5.11 Pseudo-classes
 - \[5.11.1 :first-child pseudo-class\]\(#\)
 - \[5.11.2 The link pseudo-classes: :link and :visited\]\(#\)
 - \[5.11.3 The dynamic pseudo-classes: :hover, :active, and :focus\]\(#\)
 - \[5.11.4 The language pseudo-class: :lang\]\(#\)](#)
- [5.12 Pseudo-elements
 - \[5.12.1 The :first-line pseudo-element\]\(#\)
 - \[5.12.2 The :first-letter pseudo-element\]\(#\)
 - \[5.12.3 The :before and :after pseudo-elements\]\(#\)](#)



Cascading Style Sheets Level 1

W3C Recommendation 07 Jun 1996

5.4 Type selectors

A *type selector* matches the name of a document language element type. A type selector matches every instance of the element type in the document tree.

The following rule matches all H1 elements in the document tree:

```
h1 { font-family: sans-serif }
```

5.5 Descendant selectors

At times, authors may want selectors to match an element that is the descendant of another element in the document tree (e.g., "Match those EM elements that are contained by an H1 element"). Descendant selectors express such a relationship in a pattern. A descendant selector is made up of two or more selectors separated by white space. A descendant selector of the form "A B" matches when an element B is an arbitrary descendant of some ancestor element A.

For example, consider the following rules:

```
h1 { color: red }
em { color: red }
```

Although the intention of these rules is to add emphasis to text by changing its color, the effect will be lost in a case such as:

```
<H1>This headline is <EM>very</EM> important</H1>
```

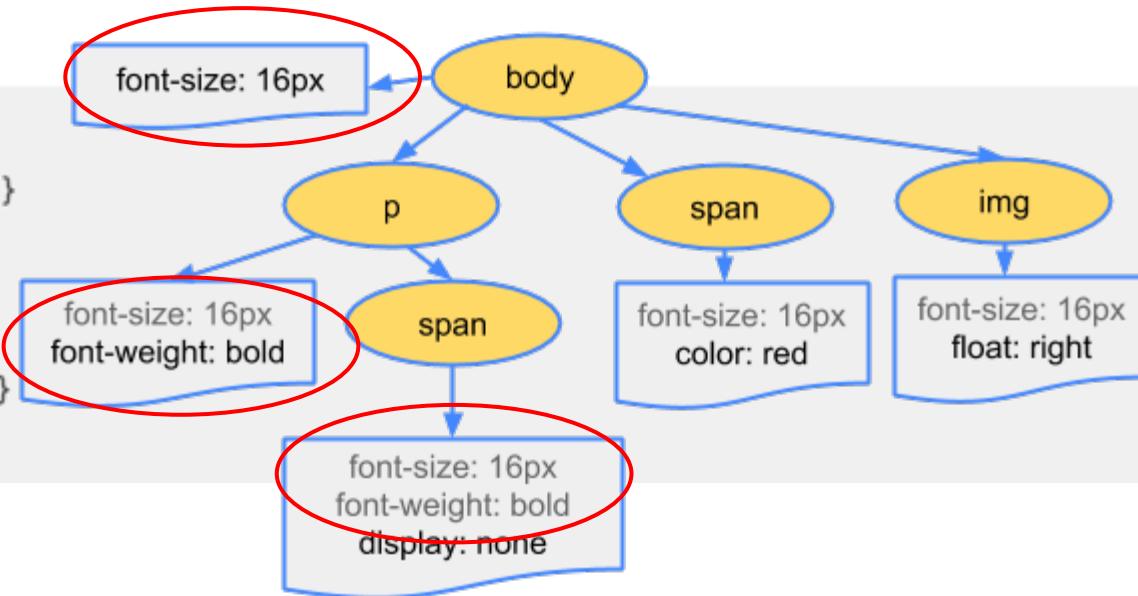
We address this case by supplementing the previous rules with a rule that sets the text color to blue whenever an EM occurs anywhere within an H1:

```
h1 { color: red }
em { color: red }
h1 em { color: blue }
```

The third rule will match the EM in the following fragment:

```
<H1>This <SPAN class="myclass">headline
is <EM>very</EM> important</SPAN></H1>
```

```
body { font-size: 16px }
p { font-weight: bold }
span { color: red }
p span { display: none }
img { float: right }
```



插花2：css标准-继承&层叠

6 Assigning property values, Cascading, and Inheritance

Contents

6.1 Specified, computed, and actual values

6.1.1 Specified values

6.1.2 Computed values

6.1.3 Used values

6.1.4 Actual values

6.2 Inheritance

6.2.1 The 'inherit' value

6.3 The @import rule

6.4 The cascade

6.4.1 Cascading order

6.4.2 !important rules

6.4.3 Calculating a selector's specificity

6.4.4 Precedence of non-CSS presentational hints

15.7 Font size: the 'font-size' property

'font-size'

<i>Value:</i>	<absolute-size> <relative-size> <length> <percentage> inherit
<i>Initial:</i>	medium
<i>Applies to:</i>	all elements
<i>Inherited:</i>	yes
<i>Percentages:</i>	refer to inherited font size
<i>Media:</i>	visual
<i>Computed value:</i>	absolute length

The font size corresponds to the em square, a concept used in typography. Note that certain glyphs may bleed outside their element's boundaries. The following meanings:

<absolute-size>

An <absolute-size> keyword is an index to a table of font sizes computed and kept by the UA. Possible values are:

[xx-small | x-small | small | medium | large | x-large | xx-large]

The following table provides user agent guidelines for the absolute-size mapping to HTML heading and absolute font-sizes. The user's preferred font size and is used as the reference middle value.

CSS absolute-size values	xx-small	x-small	small	medium	large	x-large	xx-large	
HTML font sizes	1		2	3	4	5	6	7

Implementors should build a table of scaling factors for absolute-size keywords relative to the 'medium' font size and the platform characteristics (e.g., the resolution of the device).

Different media may need different scaling factors. Also, the UA should take the quality and availability of fonts into account. The table may be different from one font family to another.

6.4.3 Calculating a selector's specificity

A selector's specificity is calculated as follows:

- count 1 if the declaration is from a 'style' attribute rather than a rule with a selector, 0 otherwise (= a) (In HTML, values are style sheet rules. These rules have no selectors, so a=1, b=0, c=0, and d=0.)
- count the number of ID attributes in the selector (= b)
- count the number of other attributes and pseudo-classes in the selector (= c)
- count the number of element names and pseudo-elements in the selector (= d)

The specificity is based only on the form of the selector. In particular, a selector of the form "[id=p33]" is counted as an attribute (d=0), even if the id attribute is defined as an "ID" in the source document's DTD.

Concatenating the four numbers a-b-c-d (in a number system with a large base) gives the specificity.

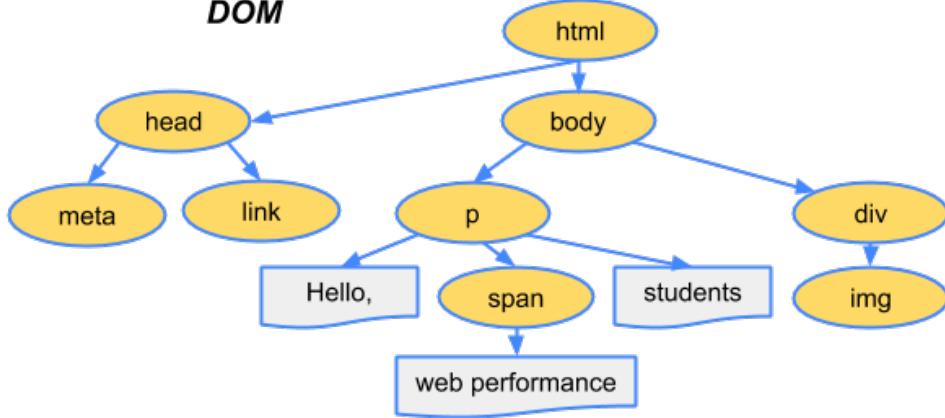
Some examples:

```
*          {} /* a=0 b=0 c=0 d=0 -> specificity = 0,0,0,0 */
li         {} /* a=0 b=0 c=0 d=1 -> specificity = 0,0,0,1 */
li:first-line {} /* a=0 b=0 c=0 d=2 -> specificity = 0,0,0,2 */
ul li      {} /* a=0 b=0 c=0 d=2 -> specificity = 0,0,0,2 */
ul ol+li   {} /* a=0 b=0 c=0 d=3 -> specificity = 0,0,0,3 */
h1 + *[rel=up]{} /* a=0 b=0 c=1 d=1 -> specificity = 0,0,1,1 */
ul ol li.red {} /* a=0 b=0 c=1 d=3 -> specificity = 0,0,1,3 */
li.red.level {} /* a=0 b=0 c=2 d=1 -> specificity = 0,0,2,1 */
#x34y     {} /* a=0 b=1 c=0 d=0 -> specificity = 0,1,0,0 */
style=""    /* a=1 b=0 c=0 d=0 -> specificity = 1,0,0,0 */
```

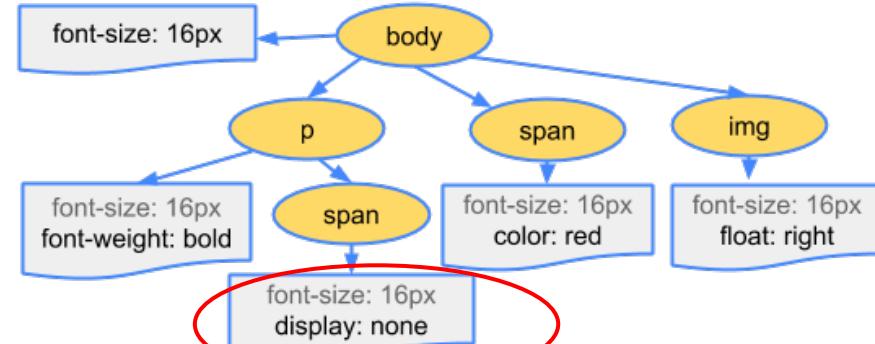
```
<HEAD>
<STYLE type="text/css">
  #x97z { color: red }
</STYLE>
</HEAD>
<BODY>
```

1. Bytes → characters → tokens → nodes → object model.
2. HTML标记语言生成**Document Object Model (DOM)**,
3. CSS被渲染生成**CSS Object Model (CSSOM)**.

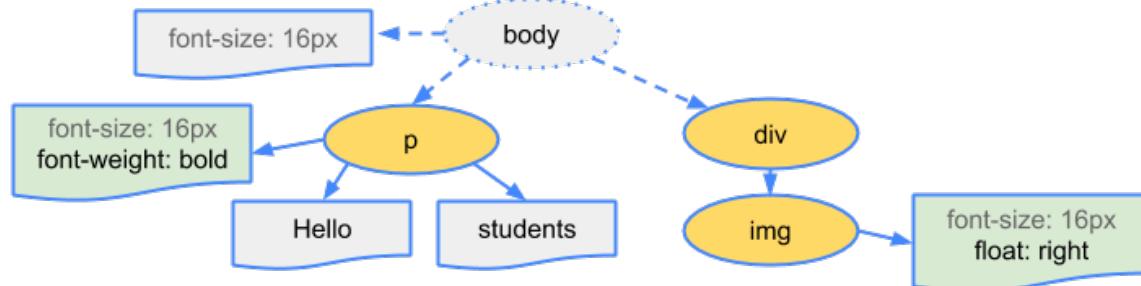
DOM



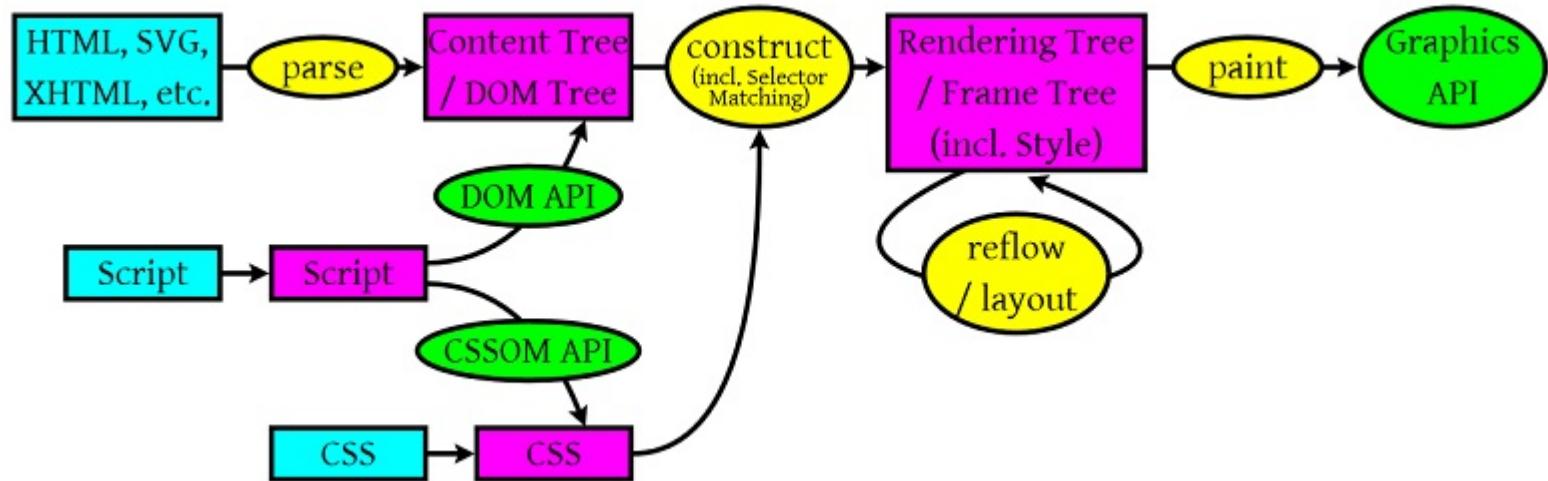
CSSOM



Render Tree



1. Bytes → characters → tokens → nodes → object model.
2. HTML标记语言生成**Document Object Model (DOM)**
3. CSS被渲染生成**CSS Object Model (CSSOM)**.
4. DOM树和CSSOM树合成**render tree**.
5. **布局引擎(Layout)** 计算每个对象的位置和大小
6. **绘制(Paint)**将render tree转为图像 推送到屏幕
 - GPU加速发生在该阶段



五

- 只有CSSOM树和DOM树完全就绪才会合成render tree
- 渲染HTML时遇到script标签，则暂停渲染，先执行完毕该js脚本
- js可以改变HTML和CSS，进而触发以上步骤【Q?】

- [HTML5ROCKS:How Browser works](#)
- [HTML5ROCKS:浏览器工作原理汉化版](#)
- [Rendering: repaint, reflow/relayout, restyle](#)
- [GPU Accelerated Compositing in Chrome](#)
- [Video: Faster HTML and CSS: Layout Engine Internals for Web Developers](#)
- [Slide: How webkit works](#)
- [Video: Google I/O 2012 - Breaking the JavaScript Speed Limit with V8 by Daniel Clifford](#)
- [Know Your Engines at O'Reilly Velocity 2011](#)
- [Video: GDC 2012: From Console to Chrome](#)
- [Video: What Browsers Really Think of Your App](#)
- [WebKit blog five-part series on rendering](#)
- [Video: Alex Russell - Life Of A Button Element](#)
- [How a web page loads](#)
- [Video: The Fundamentals, Primitives and History of HTML5](#)
- [The JavaScript engine family tree](#)
- [How JavaScript compilers work](#)

Chrome Dev Tools Timeline

Q2:为什么会有兼容性问题？

- 标准是标准(w3c, ecma..)实现是实现(浏览器厂商)
- 实现版本不一致 eg: IE低版本是ecma v3, 不支持类似于bind方法等的ES5内容
- 具体实现方式不一致 eg: 盒模型 ; web fonts
- 有时候他们会改东西: 比如**小米手机**原生浏览器
- 基于开源项目的浏览器碎片化非常严重

参考项目 : W3help [[前端组内镜像](#)]

根本原因内容列表

[本栏目文章的名词及格式介绍](#)

[更新记录](#)

[关闭全部](#) [展开全部](#)

渲染-HTML相关

渲染-CSS相关

渲染-混合类型

[脚本](#)

服务端通信

浏览器特性

- 记得查询兼容性情况
 - <http://caniuse.com/> <http://mobilehtml5.org/>
- 能力检测
- 使用polyfill
 - modernize.js等等



Web Platform Features (216)



No active development

Proposed

In development

40 canary/dev

39 beta

38 stable

37

36

35

34

33

32

31

30

NO ACTIVE DEVELOPMENT

Box Alignment

CSS



CSS properties for aligning boxes within their container. Allows for true vertical centering among other features.

CSS Custom Properties (Variables)

CSS



Introduces cascading variables as a new primitive value type that is accepted by all CSS properties, and custom...

CSS filter() image function

CSS



The function allows filtering an CSS input image with a set of filter functions. The used filter functions are the s...

Comprehensions (ES6)

JavaScript



Concise syntax for transforming collections.

DOM3 KeyboardEvent |code| attribute

DOM



1. 前端定义

2. Web标准

Q1: HTML5到底指的是什么？

3. 浏览器

Q2: 为什么会有兼容性问题？

4. 浏览器之外

Q3: 现在的html+css+js可以做什么？

5. 工作流

1. webview等类似的容器

- 移动端: android webview / IOS UIWebView [WKWebView](#) ..
- 桌面端: CEF / hex / node-webkit / chrome app

2. commonjs规范简介

- module / package / promise

3. AMD规范简介

4. node.js npm概述

移动端

- 客户端开发的浏览器控件
- 一般类似于自带浏览器 or safari
- 客户端开发可以进行配置
- 原生方式提供浏览器内无法获得的接口
- js和原生方法的互相调用

Android Webview

IOS UIWebView / **WKWebview**

CEF (chromium embedded framework)

- 在其他应用内嵌入以chromium为基础的浏览器
- BSD授权方式的开源项目 成立于2008年 基于Google Chromium项目
- 目前有CEF 1 和 CEF 3
- 笔记客户端使用的是CEF 1

Hex

- 结合了node.js和chromium
- 便于进行GUI界面的桌面App开发
- 同一份前端代码运行于win\mac\linux多系统平台
- 可以写C++扩展，增强API
- 词典客户端基础



javascript : not just for browsers any more !

目标：

- 构建js生态圈，能使用js开发web后端服务、桌面端应用、命令行工具、hybrid应用和浏览器内应用；
- 开发人员能够使用Commonjs的API开发应用，该应用可以在不同的js宿主环境和解释器内运行

组成：

- CommonJS = APIs + Specifications + Libs + [代码风格]

进行中工作：

- **Modules**(1.1.1)
- **Packages**(1.0)(例如nodejs的package.json文件)
- Package Mappings(Package的附加部分，描述用到的模块的扩展信息)
- Web Server Gateway Interface(JSGI)
- **Promises**

Module (CMD = Commonjs Module Definition)

- 全局方法:require
- 全局变量:exports , module (上面带有id)
- node.js实现的是CMD方式
- 常用的seajs实现该标准api

Module (CMD = Commonjs Module Definition)

```
1 // -----
2 // math.js
3 exports.add = function() {
4     var sum = 0, i = 0, args = arguments, l = args.length;
5     while (i < l) {
6         sum += args[i++];
7     }
8     return sum;
9 };
10 // -----
11 // increment.js
12 var add = require('math').add;
13 exports.increment = function(val) {
14     return add(val, 1);
15 };
16 // -----
```

Module (CMD = Commonjs Module Definition)

```
1 // common.js模块定义
2 define(function (require, exports, module) {
3
4     var a = require('a'),
5         b = require('b');
6
7     exports.action = function () {
8         // ...
9     };
10 });
11
```

AMD = Asynchronous Module Definition

- 异步加载
- 更符合浏览器场景
- 常用的实现：
 - require.js
 - curl
- 支持AMD载入的库
 - jquery / underscore
 - backbone 等等

```
// 一个完整的模块定义包含模块名称，模块的依赖和回调函数
define("adder", ["math"], function (math) {
    return {
        addTen : function (x) {
            return math.add(x, 10);
        }
    };
});
// 无依赖的情况下，默认依赖为require, exports 和 module
define("adder", function (require, exports) {
    exports.addTen = function (x) {
        return x + 10;
    };
});
// 省略id的情况下为匿名模块
define(["math"], function (math) {
    return {
        addTen : function (x) {
            return math.add(x, 10);
        }
    };
});
```

UMD: Universal Module Definition

```
1  function (root, factory) {
2      if (typeof define === 'function' && define.amd) {
3          // AMD
4          define(['jquery'], factory);
5      } else if (typeof exports === 'object') {
6          // Node, CommonJS-like
7          module.exports = factory(require('jquery'));
8      } else {
9          // Browser globals (root is window)
10         root.returnExports = factory(root.jQuery);
11     }
12 }(this, function ($) {
13     //    methods
14     function myFunc(){};
15
16     //    exposed public method
17     return myFunc;
18 }));
});
```

ES6 Module

```
//---- lib.js ----
export const sqrt = Math.sqrt;
export function square(x) {
    return x * x;
}
export function diag(x, y) {
    return sqrt(square(x) + square(y));
}

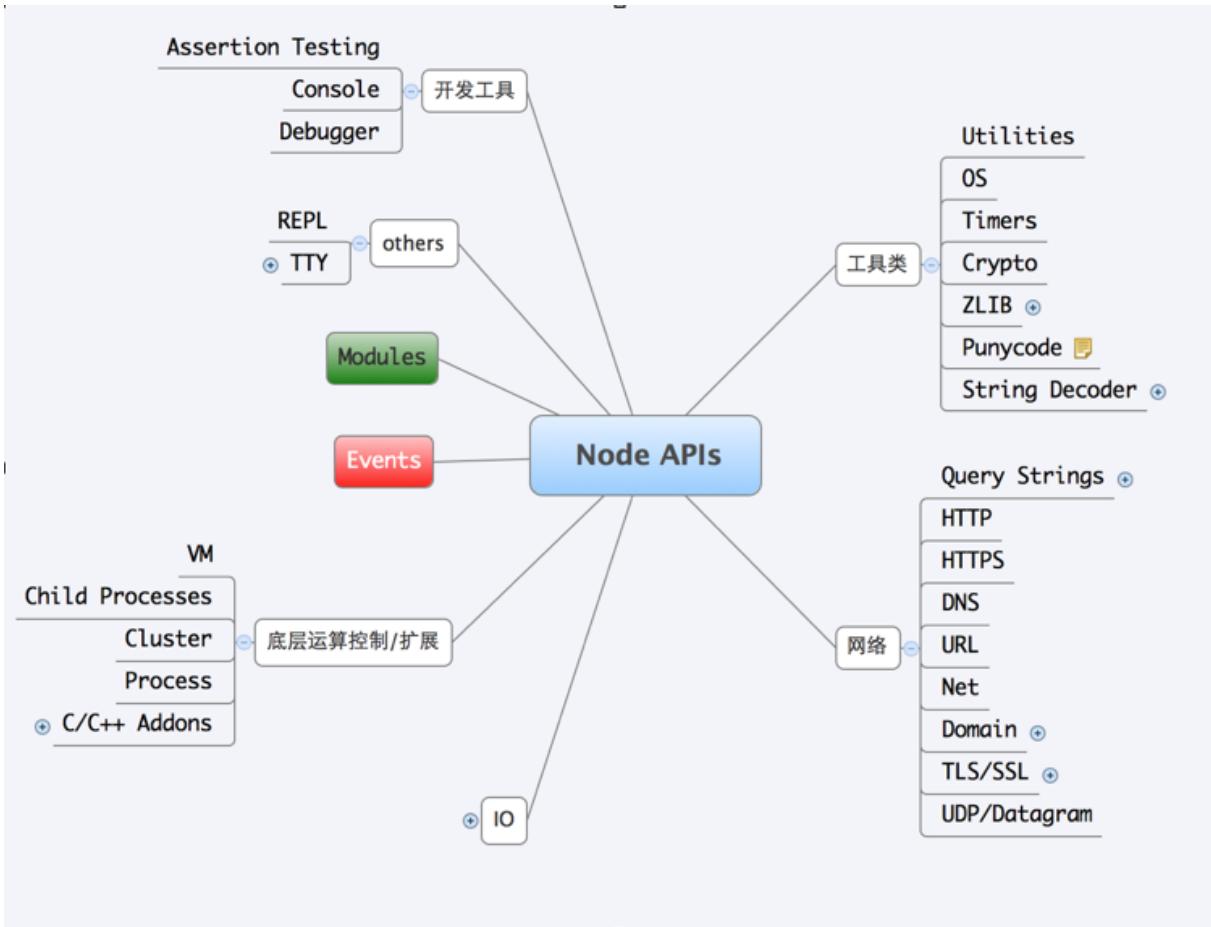
//---- main.js ----
import { square, diag } from 'lib';
console.log(square(11)); // 121
console.log(diag(4, 3)); // 5
```



a platform built on Chrome's JavaScript runtime
for easily building fast, scalable network
applications.

- common.js最有名的实现
- 遵循CMD模块载入
- package.json
- 基于V8引擎
- 实现诸多功能API
- 拥有大量可复用模块
 - NPM社区

```
{  
  "name": "underscore",  
  "description": "JavaScript's functional programming helper library.",  
  "homepage": "http://underscorejs.org",  
  "keywords": [  
    "util",  
    "functional"  
  ],  
  "author": {  
    "name": "Jeremy Ashkenas",  
    "email": "jeremy@documentcloud.org"  
  },  
  "repository": {  
    "type": "git",  
    "url": "git://github.com/jashkenas/underscore.git"  
  },  
  "main": "underscore.js",  
  "version": "1.7.0",  
  "devDependencies": {  
    "docco": "0.6.x"  
  },  
  "scripts": {},  
  "licenses": [  
    {  
      "type": "MIT",  
      "url": "https://raw.github.com/jashkenas/underscore/master/LICENSE"  
    }  
  ],  
  "files": [  
    "underscore.js",  
    "underscore-min.js",  
    "LICENSE"  
  ],  
  "gitHead": "da996e665deb0b69b257e80e3e257c04fde4191c",  
  "bugs": {  
    "url": "https://github.com/jashkenas/underscore/issues"  
  }  
}
```



IO.js

JavaScript I/O

将 ES6 带入 Node 社区！

io.js 是一个衍生自 Node.js™，并兼容 npm 的开发平台。



1.8.1 版本

下载 Linux, Win32, Win64, Mac & 其他 版本。

[更新日志](#)

[每周播报](#) 核心特性和社区动态 (Medium)

每日构建版本 可用于测试。

[常见问题](#)





chengmu

[Edit Profile](#) | [Log out](#)

Node Packaged Modules

Total Packages: 96 937

8 992 316 downloads in the last day

129 981 954 downloads in the last week

502 282 397 downloads in the last month

Patches welcome!

Any package can be installed by using `npm install`.

Add your programs to this index by using `npm publish`.

Recently Updated

- 2m [generator-frontend-app](#)
- 3m [metalsmith-jekyll-dates](#)
- 3m [angular-numeraljs](#)
- 4m [achilles](#)

Most Depended Upon

- 7169 [underscore](#)
- 6601 [async](#)
- 5743 [request](#)
- 5157 [lodash](#)

WHO'S HIRING

NUTANIX

+ 13 MORE...

npm Enterprise

Try the on-premises
solution for private
npm.

package类似于java中的jar包；

npm实际上形成了一个共享代码的社区：

Recently Updated

- 2m generator-frontend-app
- 3m metalsmith-jekyll-dates
- 3m angular-numeraljs
- 4m achilles
- 6m grunt-rtlcss
- 8m jeefo-sqlite3-orm
- 9m bitshadowmachine
- 14m speakingurl
- 19m node-simple-queue
- 19m eitrium-shell
- More...

Most Depended Upon

- 7169 underscore
- 6601 async
- 5743 request
- 5157 lodash
- 3720 commander
- 3628 express
- 2722 optimist
- 2664 coffee-script
- 2658 colors
- 2313 mkdirp
- More...

```
# chengmu at userdeMacBook-Pro.local in ~/res/YD_SeeWorld [17:  
$ npm install express  
  
# chengmu at userdeMacBook-Pro.local in ~/res/YD_SeeWorld [17:  
$ npm publish
```

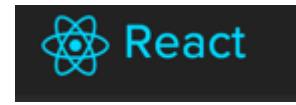
Q3: 现在的html+css+js可以做什么？



Rich Interactive MV* App.. client-side template ...



BACKBONE.JS



ANGULARJS
by Google®



网易有道

博客分类

所有文章

语言技术

交互设计

视觉设计



“破网”利器—箩筐——移动互联网分析利器
wangmh | 技术沙龙, 用户研究 | 2014-10-7 23:59

Server Side App...

app.js — blog UNREGISTERED

FOLDERS

- blog
- models
- public
 - fonts
 - images
 - javascripts
 - stylesheets
- routes
 - index.js
 - user.js
- views
 - artical.ejs
 - article.ejs
 - edit.ejs
 - footer.ejs
 - header.ejs
 - index.ejs
 - post.ejs
 - reg.ejs
 - upload.ejs
 - user.ejs

Line 1, Column 1

```
1 |
2 /**
3  * Module dependencies.
4 */
5
6 var express = require('express');
7 var routes = require('./routes');
8 var user = require('./routes/user');
9 var http = require('http');
10 var path = require('path');
11 var MongoStore = require('connect-mongo')(express);
12 var settings = require('./settings');
13 var flash = require('connect-flash');
14
15 var app = express();
16
17 // all environments
18 app.set('port', process.env.PORT || 3000)
19 ;
20 app.set('views', path.join(__dirname, 'views'));
21 app.set('view engine', 'ejs');
22 app.use(flash());
23 app.use(express.favicon());
24 app.use(express.logger('dev'));
25 app.use(express.bodyParser());
```

Spaces: 4 JavaScript

Command Line Tools...

```
slap README.md
 8,5 (43) Help: f2
 README.md
 ...
 .git/
 lib/
 node_modules
 test/
 .gitignore
 index.js
 LICENSE
 package.json
 README.md
 screenshot.p
 slap.ini
 slap.log

1 ! [Screenshot](https://raw.githubusercontent.com/ -editor/)

2
3 & slap
4 -----
5
6 CLI text editor for the masses
7
8 slap is a Sublime-like editor that strives to make editing fro
9 easier. It has **first-class mouse support**, **GUI editor lik
10 **copy/paste support**, **syntax highlighting for [many langua
11 and many other features that will make you leave nano, vim, an
12
13 Installation
14 -----
15
16 Make sure [NodeJS](http://nodejs.org/download/) is installed,
17
18 $ sudo npm install -g slap
19
20 Usage
21 -----
22
23 $ slap file.c
```



摆脱了浏览器的APP开发



网上应用店



Mindjet



All Cheat Sheets



StackEdit



Evernote Web



Postman - REST Client



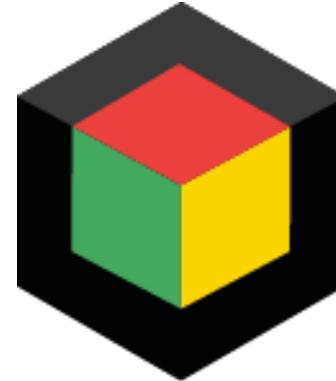
马克飞象



Google Keep: 记事和...



Bookmax



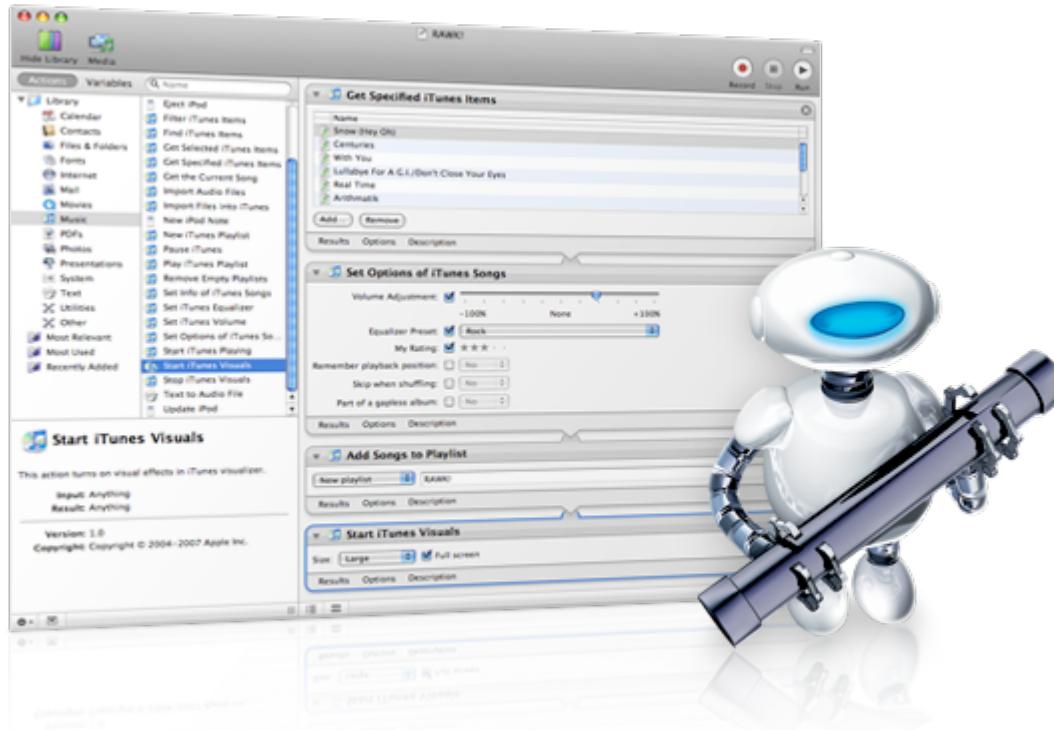
Easily create apps using the web technologies you know and love:
HTML, CSS, and JavaScript

PhoneGap is a free and open source framework that allows you to create mobile apps using standardized web APIs for the platforms you care about.

[Install PhoneGap ▶](#)

[Getting Started Guides ▶](#)

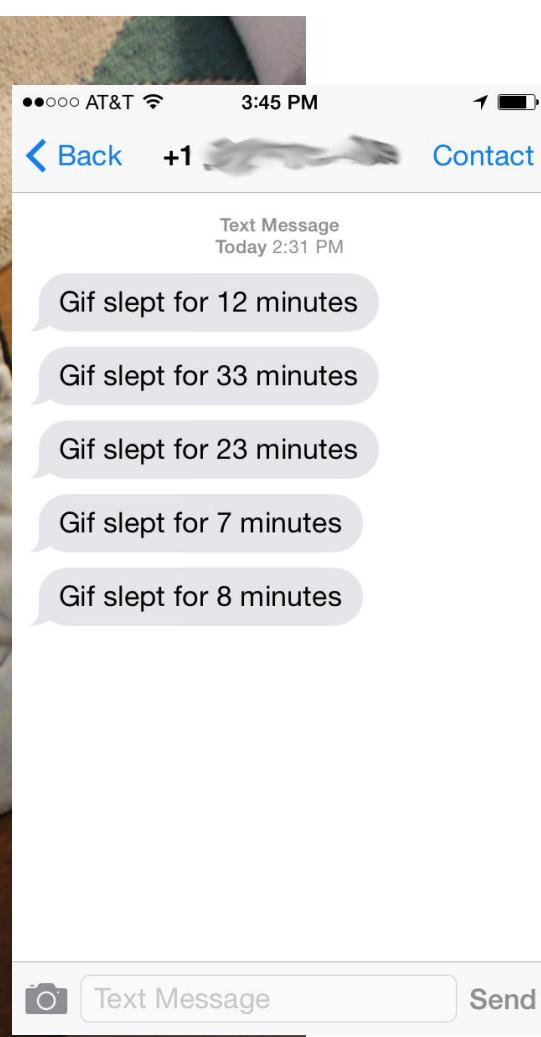
use JavaScript for OS X Automation



Slideshow is a [Node](#)/JavaScript Application Programming Interface (API) and Command Line Interface (CLI) for observing and controlling the slideshow presentation applications [Microsoft PowerPoint 2010 for Windows](#), [Microsoft PowerPoint 2011 for Mac OS X](#) and [Apple KeyNote 5 for Mac OS X](#). It can determine the current state of the application, gather information about the slides and control the application's slideshow mode. It is implemented as a thin Node/JavaScript API layer on top of platform-specific WSH/JScript and AppleScript connectors. No native code is required.

```
# CLI variant
slideshow powerpoint boot
slideshow powerpoint open sample.pptx
slideshow powerpoint start
slideshow powerpoint goto 2
sleep 2
slideshow powerpoint stop
slideshow powerpoint close
slideshow powerpoint quit
```

```
// API variant
var SlideShow = require("slideshow")
var slideshow = new SlideShow("powerpoint")
slideshow.boot()
```



```
$ npm install cylon
```

Arduino

In this example below, we will use Cylon.js with an Arduino controlled via the Firmata protocol. First, install the [cylon-firmata](#) module:

```
$ npm install cylon-firmata
```

Now, this example code will connect to an Arduino, and every second turn the LED either on, or off.

```
var Cylon = require("cylon");

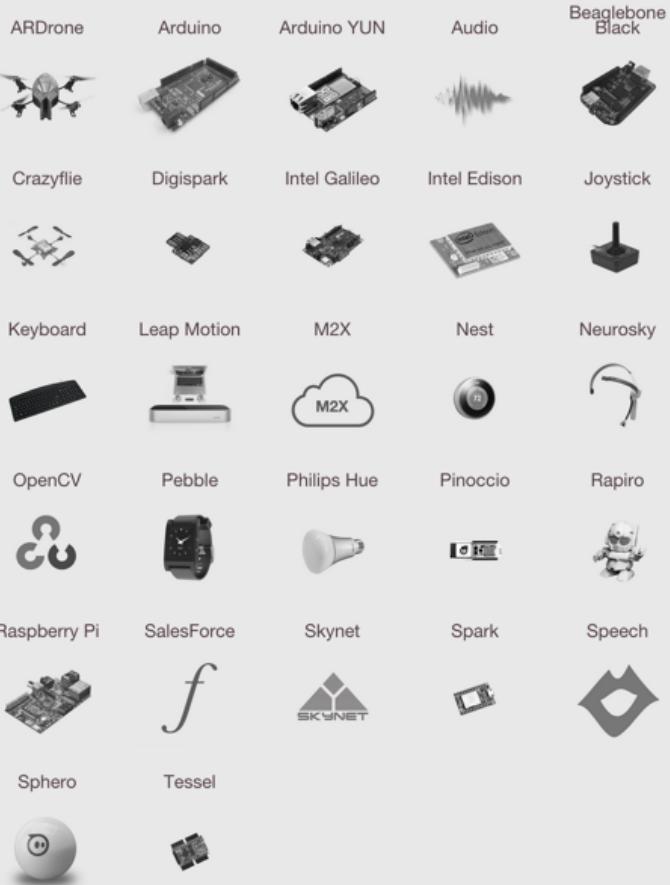
// Initialize the robot
var robot = Cylon.robot({
  // Change the port to the correct port for your Arduino.
  connection: { name: 'arduino', adaptor: 'firmata', port: '/dev/ttyACM0' },
  device: { name: 'led', driver: 'led', pin: 13 },

  work: function(my) {
    // we do our thing here
    every((1).second(), function() { my.led.toggle(); });
  }
});

// start working
robot.start();
```

Platform Support

Cylon.js has a extensible system for connecting to hardware devices. The following robotics and physical computing systems, and software platforms, are currently supported:



WebRTC



Apache CouchDB™ is a database
that uses **JSON** for documents,
JavaScript for **MapReduce** indexes,
and regular **HTTP** for its **API**

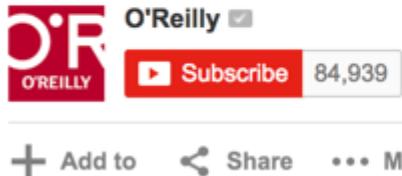
3D, animation.....

[Three.js](#)



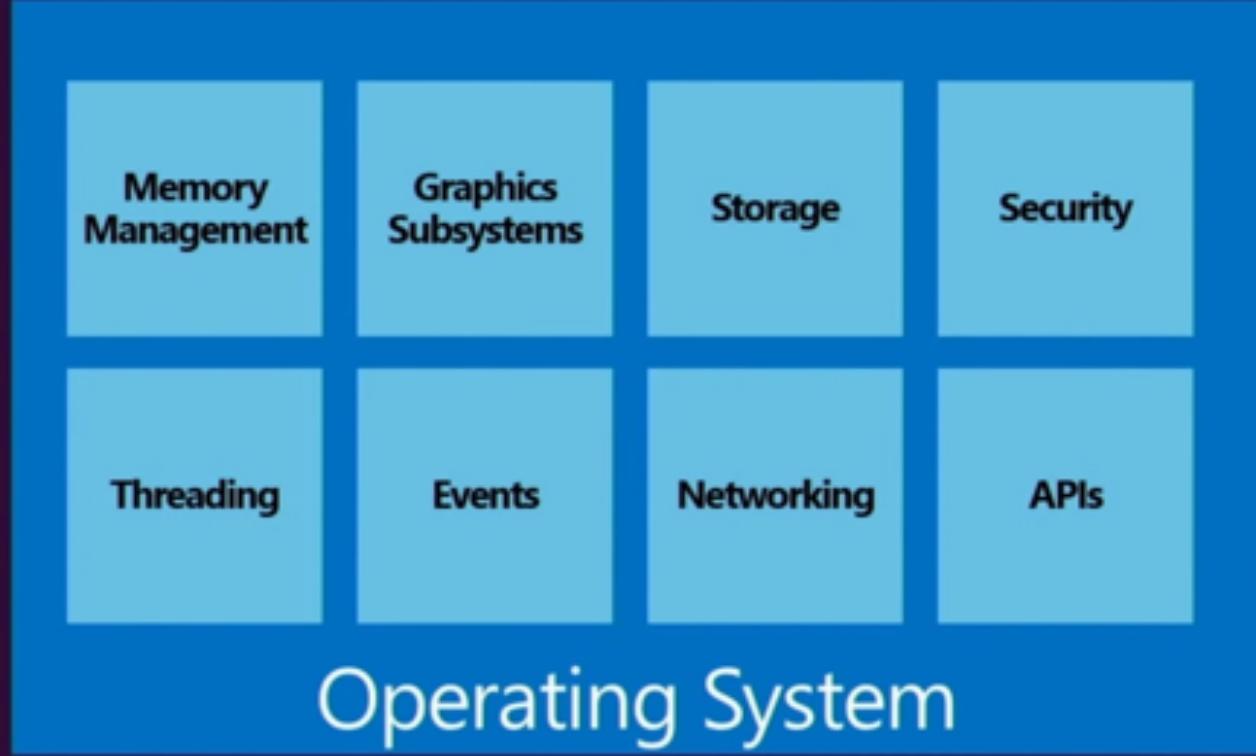
Javascript is the assembly language of the Web.

Scott Hanselman, "Virtual Machines, JavaScript and Assembler" -
Fluent 2014 Keynote



52,493

1,096 12



Hardware



Memory
Management

Garbage
Collection

Graphics
Subsystems

HTML, CSS,
Canvas, Web GL,
Audio, Video

Storage

Cookies,
IndexedDb,
File API

Security

Sandbox, SSL,
CORS

Threading

Web Workers

Events

DOM Events,
Callbacks

Network

WebSockets,
XHR, Offline,
Realtime

APIs

GeoLocation,
Sensors

JavaScript (OS)

Browser (Hardware Layer)

FOR EVERYONE

HP Chromebook 11. Made with Google.
Everything you need in one laptop. For \$279.

Learn more about the HP Chromebook 11

Buy now >

View details



**even
Node OS...**



页面建构

js开发 游戏开发

服务端 / 移动客户端开发

机器人、飞行器等等硬件开发

动画专家

交互体验专家

○ ○ ○

1. 前端定义

2. Web标准

Q1: HTML5到底指的是什么？

3. 浏览器

Q2: 为什么会有兼容性问题？

4. 浏览器之外

Q3: 现在的html+css+js可以做什么？

5. 工作流

workflow

1. 预处理

- Sass / Less / Coffeescript

2. 前端工程化 : 构建工具

- npm / grunt / gulp

3. 代码管理 - gitflow

4. 编辑器 or IDE

- Sublime Text / Vim / WebStorm / IntelliJ

4. IDE : Chrome Dev Tools

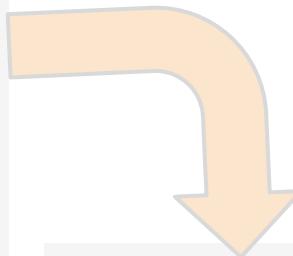
用写代码的方式写CSS :

Sass /

{less}

```
@base: #f938ab;

.box-shadow(@style, @c) when (iscolor(@c)) {
  -webkit-box-shadow: @style @c;
  box-shadow:         @style @c;
}
.box-shadow(@style, @alpha: 50%) when (isnumber(@alpha)) {
  .box-shadow(@style, rgba(0, 0, 0, @alpha));
}
.box {
  color: saturate(@base, 5%);
  border-color: lighten(@base, 30%);
  div { .box-shadow(0 0 5px, 30%) }
}
```



```
.box {
  color: #fe33ac;
  border-color: #fdcdea;
}
.box div {
  -webkit-box-shadow: 0 0 5px rgba(0, 0, 0, 0.3);
  box-shadow:     0 0 5px rgba(0, 0, 0, 0.3);
}
```



CoffeeScript

```
# Assignment:  
number = 42  
opposite = true  
  
# Conditions:  
number = -42 if opposite  
  
# Functions:  
square = (x) -> x * x  
  
# Arrays:  
list = [1, 2, 3, 4, 5]  
  
# Objects:  
math =  
  root: Math.sqrt  
  square: square  
  cube: (x) -> x * square x  
  
# Splat:  
race = (winner, runners...) ->  
  print winner, runners
```

```
var cubes, list, math, num, number, opposite, race, square,  
  __slice = [].slice;  
  
number = 42;  
  
opposite = true;  
  
if (opposite) {  
  number = -42;  
}  
  
square = function(x) {  
  return x * x;  
};  
  
list = [1, 2, 3, 4, 5];  
  
math = {  
  root: Math.sqrt,  
  square: square,  
  cube: function(x) {  
    return x * square(x);  
  }  
}
```

前端工程化

- 开发前：脚手架

- 下载底层库
- 下载模板
- 下载框架
- ...



```
# chengmu at localhost in ~/res [20:31:11]
```

```
$ yo
```

```
? 'Allo chengmu! What would you like to do? Angularexpress
```

```
Make sure you are in the directory you want to scaffold into.
```

```
This generator can also be run with: yo angularexpress
```

```
[?] Would you like to include Twitter Bootstrap? Yes
```

```
[?] Would you like to use the SCSS version of Twitter Bootstrap wit
```

```
[?] Which modules would you like to include? angular-resource.js, a
```

```
[?] Would you like to include Jade template engine? Yes
```

```
  create app/styles/main.scss
```

```
  create app/jade/index.jade
```

```
I'm all done. Running bower install & npm install for you to instal
```

```
invoke karma:app
```

```
npm WARN package.json res@0.0.0 No description
```

```
npm WARN package.json res@0.0.0 No repository field.
```



```
# chengmu at localhost in ~/res/angular_express [20:35:46]
```

```
$ tree -L 2
```

```
.
├── Gruntfile.js
├── app
│   ├── 404.html
│   ├── favicon.ico
│   ├── jade
│   ├── robots.txt
│   └── scripts
│       └── styles
└── bower.json
    ├── karma-e2e.conf.js
    ├── karma.conf.js
    ├── node_modules
    │   └── grunt
    │       ├── grunt
    │       └── grunt-karma
    │           └── karma-ng-scenario
    ├── npm-debug.log
    ├── package.json
    └── server
        ├── main.js
        └── routes
└── test
    ├── runner.html
    └── spec
```

```
12 directories, 11 files
```

前端工程化

- 开发中：
 - 本地服务
 - 监控自动编译预处理：
 - sass, less, coffeescript..
 - 监控 自动编译模板：
 - jade, doT..
 - 自动重加载:livereload..
 - js lint / css lint
 - . . .

NPM package.json 脚本

```
{  
  "title": "my-awesome-module",  
  "scripts": {  
    "lint": "npm run lint-js && npm run lint-html",  
    "lint-js": "jshint ./src",  
    "lint-html": "htmlhint -c .htmlhintrc ./src/**/*.*.html"  
  },  
  "devDependencies": {  
    "jshint": "^2.5.1",  
    "htmlhint": "^0.9.5"  
  }  
}
```

Grunt



GRUNT

grunt

```
$ grunt
Running "jshint:gruntfile" (jshint) task
>> 1 file lint free.

Running "jshint:src" (jshint) task
>> 1 file lint free.

Running "jshint:test" (jshint) task
>> 1 file lint free.

Running "qunit:files" (qunit) task
Testing test/tiny-pubsub.html....OK
>> 4 assertions passed (23ms)

Running "clean:files" (clean) task
Cleaning "dist"...OK

Running "concat:dist" (concat) task
File "dist/ba-tiny-pubsub.js" created.

Running "uglify:dist" (uglify) task
File "dist/ba-tiny-pubsub.min.js" created.
Uncompressed size: 389 bytes.
Compressed size: 119 bytes gzipped (185 bytes minified).

Done, without errors.

$ _
```

前端工程化

- 开发后

- lint代码
- 单元测试
- 编译, 各种编译
- 压缩和连接
- 图片处理, spirit
- 性能优化
- 部署到各个不同的环境
-

```
// 打包测试版，上传到测试服务器
// >> grunt test
grunt.registerTask('test',['zspack:test', 'uglify', 'scp:upload']);

grunt.registerTask('testJS',['zspack:test', 'scp:uploadJS']);

grunt.registerTask('testCSS',['zspack:test', 'scp:uploadCSS']);

// 打包上线版本，并压缩js代码
// >> grunt online
grunt.registerTask('online',['zspack:online', 'uglify']);

// 打包本地版本，利用本地调试插件，测试线上展示 {注：需要先启动server任务}
// >> grunt debug & grunt connect &
grunt.registerTask('debug',['module', 'zspack:locate', 'watch']);

// 新建模块
// 需要的参数：模块名称，必须提供Readme，然后将公用的一些东西放入其中
// 使用方法：grunt new --name=new_module --disc=这里写模块说明，建议写法为：版本号 + 功能说明
// 注意：为了后期的维护性，一定两个参数都要跟上。
grunt.registerTask('new',['zsgenerator', 'module','zspack:locate']);

// 启动server服务，提供本地开发环境
// >> grunt &
grunt.registerTask('default',['connect']);
```

编辑器

- 我爱sublime text
- 有很多非常棒的扩展插件
 - jshint / emmet / jsformat / snippets / docBlockr ...
- 前端组大家有用vim的, emacs, IntelliJ。。。
- 原则 : 爱用啥用啥
- 只要别用dreamweaver。。。

Chrome Dev Tools

- “各公司应该每个月都培训一次”
 - [《你用的Web Inspector 是怎么实现的？》](#)
- 总体概览
- 好玩但你可能不知道的：
 - break on ... / source snippets / console /
 - emulation / timeline / heap snapshot

Q&A

Thanks!