

可以用 markdown 写在 GitHub 上;

爱·学习——第一周

目录

一、 Sublime、Atom 还是 Vscode?	3
1. Sublime.....	3
2. Vscode.....	5
3. Atom.....	6
4. 情况分析.....	8
5. 总结.....	8
二、 HTML5.....	8
1. H5 与 HTML5.....	8
2. 背景与应用.....	9
3. HTML5 API:	9
4. HTML5 识途.....	11
(1) 浏览器支持.....	11
(2) 新元素.....	11
(3) 移除的元素.....	12
(4) HTML5 Canvas.....	12
(5) HTML5 MathML.....	14
(6) HTML5 拖放 (Drag 和 drop)	14
(7) HTML5 Geolocation (地理定位)	14

(8) Video(视频)与 Audio(音频).....	15
(9) 新的 Input 类型.....	15
(10) HTML5 表单属性.....	15
(11) HTML5 语义元素.....	16
(12) HTML5 Web 存储.....	16
(13) HTML5 Web SQL.....	18
(14) HTML5 应用程序缓存.....	19
(15) HTML5 Web Workers.....	20
(16) HTML5 SSE.....	20
(17) HTML5 WebSocket.....	21
三、 CSS.....	23
1. css calc ()	23
(1) calc () 的产生与 box-sizing.....	23
(2) 什么是 calc ()?.....	23
(3) calc ()能做什么，怎么用?	24
(4) 使用场景.....	25
2. css transition 与 animate.....	26
3. css flex 布局.....	30
四、 git 基本操作.....	30

一、Sublime、Atom 还是 Vscode?

有一个传言:世界上有三种程序猿,一种是用 Emacs 的,一种是用 Vim 的,一种是用其他编辑器的。Emacs,著名的集成开发环境和文本编辑器。Emacs 被公认是最受专业程序员喜爱的代码编辑器之一,另外一个 vim。

因为 Vim 和 Emacs 学习曲线比较陡,作为第三种程序员,还是来看看当前一些简单好用的主流编辑器吧!

1. [Sublime](#)

功能概览:

[Sublime Text](#) (中文官网) 是一个代码编辑器 (Sublime Text 2 (最新版 sublime text 3) 是收费软件,但可以无限期试用),也是 HTML 和散文先进的文本编辑器。Sublime Text 是由程序员 Jon Skinner 于 2008 年 1 月份所开发出来,它最初被设计为一个具有丰富扩展功能的 Vim。

Sublime Text 具有漂亮的用户界面和强大的功能,例如代码缩略图,Python 的插件,代码段等. 还可自定义键绑定,菜单和工具栏. Sublime Text 的主要功能包括: 拼写检查,书签,完整的 Python API , Goto 功能,即时项目切换,多选择,多窗口等等. Sublime Text 是一个跨平台的编辑器,同时支持 Windows、Linux、Mac OS X 等操作系统.

插件安装 [\(知乎插件推荐\)](#)

(1) 安装 Package Control

按 Ctrl+` 调出 console, 粘贴安装代码到底部命令行并回车。

重启 Sublime Text。

如果在 Preferences->package settings 中看到 package control 这一项, 则安装成功。

(2) 用 Package Control 安装其他插件

按下 Ctrl+Shift+P 调出命令面板

输入 install 调出 Install Package 选项并回车, 然后在列表选中要安装的插件。

主要优点

(1) 主流前端开发编辑器

(2) 体积较小, 运行速度快

(3) 文本功能强大

(4) 支持编译功能且可在控制台看到输出

(5) 内嵌 python 解释器支持插件开发以达到可扩展目的

(6) Package Control: ST 支持的大量插件可通过其进行管理

主要槽点

(1) 收费闭源, 收费的问题虽然不影响使用, 但是毕源带来的问题就是一个 bug 千年都不见修复. 很影响使用。

(2) 虽然有 Package Control 管理插件, 但是安装搜索插件时, 只能看到简单的一句话描述, 而且之后的插件配置不方便。

(3) 当插件多了后, sublime 会时不时崩溃, 用户也不能清晰的知道是哪个插件引起的. 只能一个个去卸载、判断。

2. [Vscode](#)

功能概览:

Vscode 同样是一个跨平台的文本编辑器, 内置了对 JavaScript, TypeScript and Node.js, C++, C#, Python, PHP 等语言的强大支持。据 vscode 的作者介绍, 这款产品可能是微软第一款支持 Linux 的产品。

插件安装

Vscode 更新到目前位置, 内置了插件安装功能, 在编辑器右侧有一个插件中心的按钮, 可以方便的安装, 升级, 卸载插件, 此外可以在 vscode 的插件商店查找适合自己的插件 vscode 插件, 相比 sublime 来说, vscode 的插件管理还是很方便的, 而且插件数量也相当可观, 其生态圈日趋成熟, 毕竟背后是微软帝国。

主要优点

- (1) 好看, 分分钟逼死 notepad++
- (2) 加载大文件几乎秒开, 试过打开 100M 的工程, 无压力
- (3) C#支持高亮, 已经编译过的还支持引用
- (4) JS、HTML 等支持高亮和补全
- (5) 全平台
- (6) 免费, 这是必须的好评

(7) 占用内存低

(8) 基于 atom, 但是性能秒 atom

主要槽点

(1) 还不够稳定, 经常会崩溃

(2) 插件还不是很完善, 但是发展速度相当快

(3) debug 的灵活性还不够

(4) 对 C#支持还不够好

(5) 不支持工程加载, 只支持文件夹加载, 引用比较复杂的大工程支持力度低

3. [Atom](#)

功能概览:

Atom 代码编辑器支持 Windows、Mac、Linux 三大桌面平台, 完全免费, 并且已经在 GitHub 上开放了全部的源代码. 在经过一段长时间的迭代开发和不断改进后, Atom 正式版在性能和稳定性方面都有着显著的改善。

开发团队将 Atom 称为一个“为 21 世纪创造的可配置的编辑器”, 它拥有非常精致细腻的界面, 并且可配置项丰富, 加上它提供了与 SublimeText 上类似的 Package Control (包管理) 功能, 更重要的是 atom 的包管理工具可视化了插件的配置, 以及插件的使用帮助以及对应的 github 的地址, 这很大程度上方便了开发人员将 Atom 打造成真正适合自己的开发工具。

作为一个现代的代码编辑器, Atom 有着各种流行编辑器都有的特性, 功能上非常丰富, 支持各种编程语言的代码高亮 (HTML / CSS / Javascript / PHP / Python / C / C++ / Objective C / Java / JSON / Perl / CoffeeScript / Go / Sass / YAML / Markdown 等等), 与大多数其他编辑器相比, Atom 的语言支持已经算是覆盖非常全面了. 另外, 它的代码补全功能 (也叫 Snippets) 也非常好用, 你只需输入几个字符即可展开成各种常用代码, 可以极大提高编程效率。

插件安装

Atom 具有交互性很好的 GUI 插件管理中心, 在这里用户可以很方便的搜索, 安装, 升级, 卸载, 配置插件。

主要优点

- (1) 开发维护团队强大, 且是开源项目, 因此修复 bug 速度快, 生态圈成长速度快。
- (2) 快捷键支持特别好, 熟悉了各种快捷键后可以成吨提高生成效率.
- (3) 比较稳定, 很少出现崩溃.
- (4) 插件管理很到位, 能准确定位出问题的插件.
- (5) 插件的生态圈发展速度特别快, 一大堆好用的插件等着用户去探索

主要槽点

- (1) 性能问题, 启动速度很慢
- (2) 打开大文件是会出现 cpu 占用过高的问题

(3) 目前相比于 sublime 技术层面还不够成熟, 有不少 bug

4. 情况分析

个人情况:

Sublime、Notepad++、VC++6.0、Codeblock、Hbuilder (云打包)。

项目使用:

《前端规范》(链接): *IDE 工具 atom 或者 vscode。对于其他 IDE 或者编译器 code 出来的代码如有出现格式混乱问题, 不予 merge, 直至修改为止。*

5. 总结

这三个编辑器, 各有所长, 也各有自己的不足, 针对不同的开发人员, 可能需要根据自己的喜好来选择, 不过幸运的是, 这三个编辑器的使用方式大同小异, 基本上可以无痛的迁移到任何一个编辑器上。

从长远发展角度来讲, Atom 和 vscode 是由两大巨头维护主推的开源项目, 开源的力量是巨大的, 因此其发展速度注定会远超 sublime。

受目前个人笔记本配置限制, 暂用 Sublime, 后期根据项目实际逐渐向 Atom 迁移。

二、HTML5

1. H5 与 HTML5

[H5:](#) 一个解决方案, 包含一定特效、看起来酷炫的移动端 onepage

网站的解决方案。（「HeiHei Hai Hui Hua（嘿嘿还会滑）」）。

H5 是一个技术合集，却被意会成了一项技术，变成可以在质上而不是量上描述的概念。

HTML5：互联网领域缩写词，全称 HyperText Markup Language 5（超文本标记语言 5），HTML5 并不是一项技术，而是一个标准。

2. 背景与应用

HTML5 的设计目的是为了在移动设备上支持多媒体。

HTML5 是 W3C 与 WHATWG 合作的结果，WHATWG 指 Web Hypertext Application Technology Working Group。

WHATWG 致力于 web 表单和应用程序，而 W3C 专注于 XHTML 2.0。在 2006 年，双方决定进行合作，来创建一个新版本的 HTML。

应用：多媒体、动画、本地数据存储、web APP 缓存引用、拖放、表单验证、XHTMLHttpRequest 等。

个人认为，HTML5 更多的是对 HTML 的增强，结合 CSS3，将一些原本需要使用 JavaScript 等来实现的东西，通过 HTML 来完成，追求学习的简易、使用的高效、功能的强大。

3. [HTML5 API:](#)

为了更好地处理今天的互联网应用，HTML5 添加了很多新元素及功能，比如：图形的绘制，多媒体内容，更好的页面结构，更好的形式处理，和几个 api 拖放元素，定位，包括网页、应用程序缓存，存

储，网络工作者等。

HTML5API 盘点：Canvas、Audio 和 Video、Geolocation（地理定位功能）、WebSockets（实现了浏览器与服务器全双工通信）、Web storage（localStorage、sessionStorage 和离线存储 manifest，localStorage 可用于不同标签页数据传递；离线存储详见下图）、Communication（跨文档消息通信，postMessage API）、Web Workers（后台处理、多线程支持、监听由后台服务器广播的新闻消息）、requestAnimationFrame（以前不够了解，[参见这里](#)）。

首先，需要在页面头加入manifest属性：

```
1 <!DOCTYPE HTML>
2 <html manifest = "cache.manifest">
3 ...
4 </html>
```

然后cache.manifest文件的书写方式为：

```
CACHE MANIFEST
#v0.11

CACHE:

js/app.js
css/style.css

NETWORK:
resource/logo.png

FALLBACK:
/offline.html
```

离线存储的manifest一般由三个部分组成：

- 1.CACHE:表示需要离线存储的资源列表，由于包含manifest文件的页面将被自动离线存储，所以不需要把页面自身也列出来。
- 2.NETWORK:表示在它下面列出来的资源只有在在线的情况下才能访问，他们不会被离线存储，所以在离线情况下无法使用这些资源。不过，如果在CACHE和NETWORK中有一个相同的资源，那么这个资源还是会被离线存储，也就是说CACHE的优先级更高。
- 3.FALLBACK:表示如果访问第一个资源失败，那么就使用第二个资源来替换他，比如上面这个文件表示的就是如果访问根目录下任何一个资源失败了，那么就去访问offline.html。

除以上列出的 API，HTML5 新标签、新特性还包括：HTML5 `<!DOCTYPE>`；新的特殊内容元素，比如 `article`、`footer`、`header`、`nav`、`section`；新的表单控件、新表单元素、新属性、新输入类型，自动验证等。

4. HTML5 识途

参考：<http://www.php.cn/course/34.html>

（1）浏览器支持

IE8 及以下不支持 H5、CSS3 Media，可使用 `html5shiv.min.js` 及 `Response.min.js` 静态包。



所有浏览器（包括旧的和最新的），对无法识别的元素会作为内联元素自动处理，为了能让旧版本的浏览器正确显示这些元素，可以设置 CSS 的 `display` 属性值为 `block`。同理，也可以为 HTML 添加新的元素，并使用 CSS 设置样式。

（2）新元素

<canvas>：定义图形，基于 JavaScript 的绘图 API（标签只是图形容器，必须使用脚本来绘制图形。）；

新多媒体元素：`<audio>`、`<video>`、`<source>` (`<video>` / `<audio>`)

元素支持多个 `<source>` 元素. `<source>` 元素可以链接不同的文件。浏览器将使用第一个可识别的格式）、`<embed>`（嵌入的内容，如插件）、`<track>`（规定外部文本轨道）；

新表单元素：[`<datalist>`](#)（选项列表，定义 input 可能的值）、[`<keygen>`](#)（密钥对生成器字段）、[`<output>`](#)（不同类型的输出）；

新的语义和结构元素：`<article>`、`<aside>`、`<command>`、`<details>`、`<dialog>`、`<summary>`、`<figure>`、`<figcaption>`、`<footer>`、`<header>`、`<meter>`、`<nav>`、`<progress>`、`<section>`、`<time>`、`<wbr>`（如果单词太长，或者担心浏览器会在错误的位置换行，可以使用 `<wbr>` 元素来添加 Word Break Opportunity（单词换行时机））；

（3）移除的元素

`<acronym>`：HTML5 中使用 `<abbr>` 标签；`<basefont>`：标签定义文档中所有文本的默认颜色、大小和字体，改用 CSS 设置；`<big>`、`<center>`、``、`<frame>`、`<frameset>`、`<noframes>`、`<strike>`（添加删除线，用 `` 标签代替）、`<tt>`（打字机文本，CSS 代替）；

（4）HTML5 Canvas

①创建一个画布（Canvas）（默认情况下 `<canvas>` 元素没有边框和内容。）；

②使用 JavaScript 来绘制图像

找到 `<canvas>` 元素；创建对象：`getContext("2d")` 对象是内建的 HTML5 对象，拥有多种绘制路径、矩形、圆形、字符以及添加图像的方法；绘制（`beginPath`、`moveTo`、`lineTo`、`fill`、`stroke` 绘制当前路径、`arc(x, y, r, start, stop)`、`Gradient` 渐变、`drawImage` 等）；

SVG 与 Canvas

SVG：基于 XML 的可伸缩矢量图形（Scalable Vector Graphics）；GitHub 官网的 logo 用的 SVG 生成。

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

SVG 与 Canvas 的区别

SVG 是一种使用 XML 描述 2D 图形的语言。
Canvas 通过 JavaScript 来绘制 2D 图形。
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 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	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
--	---------

(5) HTML5 MathML

HTML5 可以在文档中使用 MathML 元素，对应的标签是 `$...$`;

MathML 是数学标记语言，是一种基于 XML（标准通用标记语言的子集）的标准，用来在互联网上书写数学符号和公式的置标语言。

(6) HTML5 拖放 (Drag 和 drop)

设置元素为可拖放 (`draggable="true"`) ;

拖动什么 - `ondragstart` 和 `setData()`;

放到何处 - `ondragover`;

进行放置 - `ondrop` (把被拖元素通过 `appendChild` 追加到放置元素中) ;

需使用 `preventDefault()` 阻止默认事件;

(7) HTML5 Geolocation (地理定位)

使用 `getCurrentPosition()` 方法来获得用户的位置 (需要用户授权) ; 检测浏览器支持情况 (`navigator.geolocation`) 、 `position.coords.latitude` 、 `position.coords.longitude` 、 `coords.accuracy`、`coords.altitude`、处理错误和拒绝 (错误代码: `Permission denied` - 用户不允许地理定位、`Position unavailable`

- 无法获取当前位置、Timeout - 操作超时）、在地图中显示结果；

Geolocation 对象其他方法：

watchPosition() - 返回用户的当前位置，并继续返回用户移动时的更新位置（就像汽车上的 GPS）；

clearWatch() - 停止 watchPosition() 方法；

（8） Video(视频)与 Audio(音频)

产生：大多数视频/音频是通过插件（比如 Flash）来显示的，然而，并非所有浏览器都拥有同样的插件，HTML5 规定了一种通过 video/audio 元素来包含视频/音频的标准方法。元素提供播放、暂停和音量控件。

（9）新的 Input 类型

HTML5 拥有多个新的表单输入类型。这些新特性提供了更好的输入控制和验证：color、date、datetime、datetime-local、email、month、number（max、min、Step、value）、range（max、min、Step、value）、search、tel（手机端实现点击跳转到电话拨号页）、time、url、week 等。

（10）HTML5 表单属性

HTML5 的 <form> 和 <input> 标签添加了几个新属性。

<form>新属性：autocomplete、novalidate；

<input>新属性：autocomplete、autofocus、form、formaction、formenctype、formmethod、formnovalidate、formtarget、height and width、list、min and max、multiple、pattern（regexp）、

placeholder、required、step 等。

(11) HTML5 语义元素

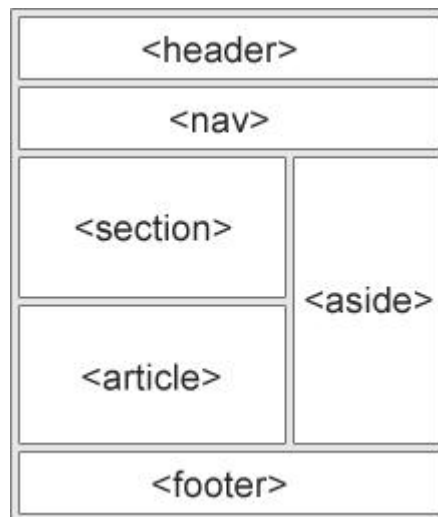
语义= 意义.

语义元素 = 元素的意义.

一个语义元素能够清楚的描述其意义给浏览器和开发者。

无语义元素实例：<div> 和 – 无需考虑内容.

语义元素实例：<form>, <table>, and – 清楚的定义了它的内容。



<figure>标签规定独立的流内容（图像、图表、照片、代码等等），元素的内容应该与主内容相关，但如果被删除，则不应对文档流产生影响。

<figcaption>（非块元素） 标签定义 <figure> 元素的标题，<figcaption>元素应该被置于 “figure” 元素的第一个或最后一个子元素的位置。

(12) HTML5 Web 存储

前端 HTML5 几种存储方式的总结：

<https://www.cnblogs.com/LuckyWinty/p/5699117.html>

关于 Cookie、session 和 Web Storage:

<https://juejin.im/post/5ad5b9116fb9a028e014fb19>

Web Storage--HTML5 本地存储:

<https://segmentfault.com/a/1190000007926921>

聊一聊前端存储那些事儿:

<https://segmentfault.com/a/1190000005927232>

LocalStorage 用法:

背景: HTML5 之前, 存储主要是用 cookies。cookies 缺点有在请求头上带着数据, 大小是 4k 之内。主 Domain 污染。主要应用如购物车、客户登录等。对于 IE 浏览器有 UserData, 大小是 64k, 只有 IE 浏览器支持。

HTML 存储方式目标: 解决 4k 的大小问题; 解决请求头常带存储信息的问题; 解决关系型存储的问题; 跨浏览器。

存储方式: 以键值对 (Key-Value) 的方式存储, 永久存储, 永不失效, 除非手动删除; 大小每个域名 5M。

检测方法: `window.localStorage`;

常用的 API:

<code>localStorage.getItem(key)</code>	取记录
<code>localStorage.setItem(key, value)</code>	设置记录
<code>localStorage.removeItem(key)</code>	移除记录
<code>localStorage.key(index)</code>	取 key 所对应的值

localStorage.clear()	清除记录
-----------------------	------

存储的内容：只要是能序列化成字符串的内容都可以存储, 如：数组，图片，json，样式，脚本等；

HTML5 的本地存储 API 中的 localStorage 与 sessionStorage 在使用方法上是相同的, 区别在于 sessionStorage 在关闭页面后即被清空, 而 localStorage 则会一直保存。

(13) HTML5 Web SQL

关系数据库，通过 SQL 语句访问。

三个核心方法：

openDatabase：这个方法使用现有的数据库或者新建的数据库创建一个数据库对象。

打开数据库：

```
var db = openDatabase('mydb', '1.0', 'Test DB', 2 * 1024 * 1024);
```

//openDatabase() 方法对应的五个参数分别为：数据库名称、版本号、描述文本、数据库大小、创建回调

transaction：这个方法让我们能够控制一个事务，以及基于这种情况执行提交或者回滚。

executeSql：这个方法用于执行实际的 SQL 语句（基本上都是用 SQL 语句进行数据库的相关操作，前期已对 SQL 语句有过一些接触，能够看懂和写出简单的 SQL 语句）。

实例：

```
var db = openDatabase('mydb', '1.0', 'Test DB', 2 * 1024 * 1024);
db.transaction(function (tx) {
    tx.executeSql('CREATE TABLE IF NOT EXISTS LOGS (id unique, log)');
    tx.executeSql('INSERT INTO LOGS (id, log) VALUES (1, "菜鸟教程")');
    tx.executeSql('INSERT INTO LOGS (id, log) VALUES (2, "www.runoob.com")');
});
```

索引数据库 (IndexedDB) API (作为 HTML5 的一部分) 对创建具有丰富本地存储数据的数据密集型 的离线 HTML5 Web 应用程序很有用。同时它还有助于本地缓存数据，使传统在线 Web 应用程序（比如移动 Web 应用程序）能够更快地运行和响应。

(14) [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。

缓存带来的优势：

离线浏览 – 用户可在应用离线时使用

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。

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

注意：以 “#” 开头的是注释行，但也可满足其他用途。应用的缓存

会在其 manifest 文件更改时被更新。如果您编辑了一幅图片，或者修改了一个 JavaScript 函数，这些改变都不会被重新缓存。更新注释行中的日期和版本号是一种使浏览器重新缓存文件的办法。

关于应用程序缓存的说明：

请留心缓存的内容；一旦文件被缓存，则浏览器会继续展示已缓存的版本，即使您修改了服务器上的文件。为了确保浏览器更新缓存，您需要更新 manifest 文件。

离线缓存与传统浏览器缓存区别：

- ①离线缓存是针对整个应用，浏览器缓存是单个文件
- ②离线缓存断网了还是可以打开页面，浏览器缓存不行
- ③离线缓存可以主动通知浏览器更新资源

（15）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。您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，一般用于更耗费 CPU 资源的任务。

（16）HTML5 SSE

HTML5 服务器发送事件（server-sent event）允许网页获得来自服务器的更新。

Server-Sent 事件指的是网页自动获取来自服务器的更新，以前也可能做到这一点，前提是网页不得不询问是否有可用的更新。通过

服务器发送事件，更新能够自动到达。

EventSource 对象用于接收服务器发送事件通知。

SSE 还需要能够发送数据更新的服务器（比如 PHP 和 ASP）。服务器端事件流的语法是非常简单的。把 “Content-Type” 报头设置为 “text/event-stream”，就可以开始发送事件流了。

可以用 onmessage（当接收到消息）、onopen（当通往服务器的连接被打开）、onerror（当发生错误） 事件来获取消息。

（17）HTML5 WebSocket

WebSocket 是 HTML5 开始提供的一种在单个 TCP 连接上进行全双工通讯的协议。

在 WebSocket API 中，浏览器和服务器只需要做一个握手的动作，然后，浏览器和服务器之间就形成了一条快速通道。两者之间就直接可以数据互相传送。

浏览器通过 JavaScript 向服务器发出建立 WebSocket 连接的请求，连接建立以后，客户端和服务器端就可以通过 TCP 连接直接交换数据。

当你获取 Web Socket 连接后，你可以通过 **send()** 方法来向服务器发送数据，并通过 **onmessage** 事件来接收服务器返回的数据。

为了建立一个 WebSocket 连接，客户端浏览器首先要向服务器发起一个 HTTP 请求，这个请求和通常的 HTTP 请求不同，包含了一些附加头信息，其中附加头信息 “Upgrade: WebSocket” 表明这是一个申请协议升级的 HTTP 请求，服务器端解析这些附加的头信息然后产生应答信息返回给客户端，客户端和服务器的 WebSocket 连接就建立起来了，双方就可以通过这个连接通道自由的传递信息，并且这个

连接会持续存在直到客户端或者服务器端的某一方主动的关闭连接。

WebSocket 属性：

Socket.readyState:

只读属性 **readyState** 表示连接状态，可以是以下值：

0	表示连接尚未建立；
1	表示连接已建立，可以进行通信；
2	表示连接正在进行关闭；
3	表示连接已经关闭或者连接不能打开。

Socket.bufferedAmount:

只读属性 **bufferedAmount** 已被 `send()` 放入正在队列中等待传输，但是还没有发出的 UTF-8 文本字节数。

WebSocket 事件

以下是 WebSocket 对象的相关事件。

事件	事件处理程序	描述
open	Socket.onopen	连接建立时触发
message	Socket.onmessage	客户端接收服务端数据时触发
error	Socket.onerror	通信发生错误时触发
close	Socket.onclose	连接关闭时触发

WebSocket 方法

Socket.send() 使用连接发送数据

Socket.close() 关闭连接

在标准 HTML5 中，`<html>` 和 `<body>` 标签是可以省略的；省略

<html> 或 <body> 在 DOM 和 XML 软件中会崩溃（缺少根元素？）。

在标准 HTML5 中，<head> 标签是可以省略的；默认情况下，浏览器会将 <body> 之前的内容添加到一个默认的 <head> 元素上。

三、CSS

1. css calc ()

参考：《css 揭秘》、[CSS3 的 calc\(\) 使用](#)、[calc\(\) 如何工作](#)。

(1) calc () 的产生与 box-sizing

平时在制作页面的时候，总会碰到有的元素是 100% 的宽度，如果元素宽度为 100% 时，其自身如果有别的盒模型属性设置，那将导致盒子撑破。比如说，有一个边框，或者说有 margin 和 padding，这些都会让你的盒子撑破（标准模式下，除 IE 怪异模式）。平时可能只能通过改变结构来实现。就算你通过繁琐的方法实现了，但有益于浏览器的兼容性而导致最终效果不一致。CSS3 属性中的 box-sizing (content-box | border-box | inherit) 在一定程度上解决了这样的问题，其实 calc () 函数功能实现上面的效果更简单。

(2) 什么是 calc ()?

calc 是英文单词 calculate (计算) 的缩写，是 css3 的一个新增的功能，用来指定元素的长度。比如说，你可以使用 calc () 给元素的 border、margin、padding、font-size 和 width 等属性设置动态值。为何说是动态值呢？因为我们使用的表达式来得到的值。不过 calc ()

最大的好处就是用在**流体布局**上，可以通过 `calc()` 计算得到元素的宽度。

(3) `calc()` 能做什么，怎么用？

`calc()` 能让你给元素的做计算，你可以给一个 `div` 元素，使用百分比、`em`、`px` 和 `rem` 单位值计算出其宽度或者高度，比如说 “`width:calc(50% + 2em)`”，这样一来你就不用考虑元素 `DIV` 的宽度值到底是多少，而把这个烦人的任务交由浏览器去计算。

运算规则：

`calc()` 使用通用的数学运算规则，但是也提供更智能的功能：

- 使用 “+”、“-”、“*” 和 “/” 四则运算；
- 可以使用百分比、`px`、`em`、`rem` 等单位；
- 可以混合使用各种单位进行计算；
- 表达式中有 “+” 和 “-” 时，其前后必须要有空格，如 “`widht: calc(12%+5em)`” 这种没有空格的写法是错误的；
- 表达式中有 “*” 和 “/” 时，其前后可以没有空格，但建议留有空格。

`calc()` 函数可以用来对数值属性执行**四则运算**。比如，`<length>`，`<frequency>`，`<angle>`，`<time>`，`<number>` 或者 `<integer>` 数据类型，例：

```
.foo {  
  width: calc(50vmax + 3rem);  
  padding: calc(1vw + 1em);  
  transform: rotate( calc(1turn + 28deg) );  
  background: hsl(100, calc(3 * 20%), 40%);  
  font-size: calc(50vw / 3);  
}
```

`calc()` 与 SASS:

[SASS 常用用法](#) 包括：变量（SASS 允许使用变量，所有变量以 `$` 开头）、

计算功能、嵌套[选择器嵌套、&的嵌套（引用父元素）、属性嵌套]、继承（`@extend`）、Mixin（`@mixin` 定义、`@include` 复用，mixin 的强大之处，在于可以指定参数和缺省值，这样我们就可以复用一些样式，只需要传递一个参数，就像调用一个函数一样！）、导入文件（`@import`）、注释（`/*`会保留到编译后的文件，单行注释`//`不会）。[详见中文文档。](#)

calc()：首先，`calc()` 能够组合不同的单元。特别是，我们可以混合计算绝对单位（比如百分比与视口单元）与相对单位（比如像素）。例如，我们可以创建一个表达式，用一个百分比减掉一个像素值。第二，使用 `calc()`，计算值是表达式它自己，而非表达式的结果。当使用 CSS 预处理器做数学运算时，给定值为表达式的结果。这意味着浏览器中的值可以更加灵活，能够响应视口的改变。我们能够给元素设定一个高度为视口的高度减去一个绝对值，它将随视口的改变进行调节。

注意：`calc()` 函数可以嵌套；对不支持 `calc()` 的，使用一个固定值作为降级方案；实际使用时，同样需要添加浏览器的前缀。



请不要忘记在 `calc()` 函数内部的 `-` 和 `+` 运算符的两侧各加一个空白符，否则会产生解析错误！这个规则如此怪异，是为了向前兼容：未来，在 `calc()` 内部可能会允许使用关键字，而这些关键字可能会包含连字符（即减号）。

（4）使用场景

知道元素尺寸的垂直居中（`top`、`left`：50%减去半宽或半高）；

使用 `rem`, `calc()` 函数能够用来创建一个基于视口的栅格;
`calc()` 使计算更加清晰。

2. css transition 与 animate

(1) [animation、transition、transform、translate 傻傻分不清:](#)

属性	含义
animation (动画)	用于设置动画属性, 他是一个简写的属性, 包含6个属性
transition (过渡)	用于设置元素的样式过渡, 和animation有着类似的效果, 但细节上有很大的不同
transform (变形)	用于元素进行旋转、缩放、移动或倾斜, 和设置样式的动画并没有什么关系, 就相当于color一样用来设置元素的 “外表”
translate (移动)	translate只是transform的一个属性值, 即移动。

(2) [Transition \(过渡\)](#)

什么叫过渡? 字面意思上来讲, 就是元素从这个属性(如 `color`)的某个值如(`red`)过渡到这个属性(`color`)的另外一个值(`green`), 这是一个状态的转变, 需要一种条件来触发这种转变, 比如我们平时用到的:`:hover`、`:focus`、`:checked`、媒体查询或者 JavaScript。常见触发方式及用法[详见这里](#)。

在执行的过程中声明关键帧的动画; `transition` 产生动画的条件是 `transition` 设置的 `property` 发生变化, 这种动画的特点是需要“一个驱动力去触发”, 有着以下几个不足:

①这类动画必须要事件触发, 所以没法在网页加载时自动发生

②是一次性的，不能重复发生，除非一再触发

③只能定义开始状态和结束状态，不能定义中间状态，也就是说只有两个状态

④一条 transition 规则，只能定义一个属性的变化，不能涉及多个属性。

语法：**transition: property duration timing-function delay;**

值	描述
transition-property	规定设置过渡效果的 CSS 属性的名称
transition-duration	规定完成过渡效果需要多少秒或毫秒
transition-timing-function	规定速度效果的速度曲线
transition-delay	定义过渡效果何时开始

transition-property 指定需要过渡的 CSS 属性。并不是所有属性都能过渡的，只有能数字量化的 CSS 属性才能过渡[颜色系、数字系、01 系（如 visibility，0 表示隐藏，1 表示显示）、渐变、阴影、all（表示所有属性都将获得过渡效果）]。

transition-duration 过渡需要的时间，单位可指定 s 秒，也可指定 ms 毫秒。默认值是 0，表示立刻变化。如果设置了多个过渡属性，但每个属性的过渡时间都一样，设一个值即可，该值会应用到所有过渡属性上。

transition-timing-function 过渡函数，有 linear, ease, ease-in, ease-out, ease-in-out, cubic-bezier(n, n, n, n), steps。其实它们都是贝赛尔曲线。如果预定义好的这些函数满足不了你的需求，可以通过 cubic-bezier(n, n, n, n) 自定义平滑曲线。

steps 可以把过渡阶段分割成等价的几步。

transition-delay 延迟开始过渡的时间，默认值是 0，表示不延迟，立即开始过渡动作。单位是 s 秒或 ms 毫秒。该属性还能设负时间，意思是让过渡动作从该时间点开始启动，之前的过渡动作不显示。

隐式过渡（比如设成 em 的属性）、开关过渡和永久过渡（最终状态）、auto 过渡（必须靠 JS，通过 getComputedStyle 将 auto 转换为固定值，通过 style 设置该值，然后再触发 CSS 的过渡效果。）

（3）Animation（动画）

参考：[CSS3 动画](#)、[CSS Animation for Beginners](#)

在官方的介绍上介绍这个属性是 transition 属性的扩展，弥补了 transition 的很多不足（animation 是由多个 transition 的效果叠加，并且可操作性更强，能够做出复杂酷炫的效果）。

语法：**animation: name duration timing-function delay iteration-count direction play-state fill-mode;**

值	描述
name	用来调用@keyframes定义好的动画，与@keyframes定义的动画名称一致
duration	指定元素播放动画所持续的时间
timing-function	规定速度效果的速度曲线，是针对每一个小动画所在时间范围的变换速率
delay	定义在浏览器开始执行动画之前等待的时间，值整个animation执行之前等待的时间
iteration-count	定义动画的播放次数，可选具体次数或者无限(infinite)
direction	设置动画播放方向：normal(按时间轴顺序),reverse(时间轴反方向运行),alternate(轮流，即来回往复进行),alternate-reverse(动画先反运行再正方向运行，并持续交替运行)
play-state	控制元素动画的播放状态，通过此来控制动画的暂停和继续，两个值：running(继续)，paused(暂停)
fill-mode	控制动画结束后，元素的样式，有四个值：none(回到动画没开始时的状态)，forwards(动画结束后动画停留在结束状态)，backwards(动画回到第一帧的状态)，both(根据animation-direction轮流应用forwards和backwards规则)，注意与iteration-count不要冲突(动画执行无限次)

animation 与 transition 不同的是，keyframes 提供更多的控制，尤其是时间轴的控制，这点让 css animation 更加强大，使得 flash 的部分动画效果可以由 css 直接控制完成，而这一切，仅仅只需要几行代码，也因此诞生了大量基于 css 的动画库，用来取代 flash 的动画部分。

@keyframes 规则：@keyframes 定义关键帧，即动画每一帧执行的是什么，animation 是来定义动画的每一帧如何执行的。创建动画的原理是，将一套 CSS 样式逐渐变化为另一套样式。在动画过程中，可以多次改变这套 CSS 样式。

语 法 : @keyframes animationname {keyframes-selector {css-styles;}}

animationname 是动画名字; keyframes-selector 帧选择器, 用百分比定义, 0%-100%, 与 from-to 等价, 作用是定义不同时间段的不同样式。

拓展: [\[译\]详解 React Native 动画](#)

简单一次性的动画可以使用 transition, 比较逻辑清晰, 可维护性较好。比较复杂的动画, 这个时候便可以拿出 animation 开始你的表演, 其实不仅仅用 css 能实现动画, 用 js 同样可以操控元素的样式实现动画(setTimeout, setInterval, 其中 requestAnimationFrame 能够更高性能地执行动画)。

3. css flex 布局

css transition 逐渐消失

css animate 淡入淡出

四、git 基本操作