

OneNote 可跟踪工作、家庭或学校的资料

收集、组织、搜索和分享
创意、想法、剪辑、列表、项目、
会议笔记、讲座、参考资料、
活动计划、说明、日记
等等

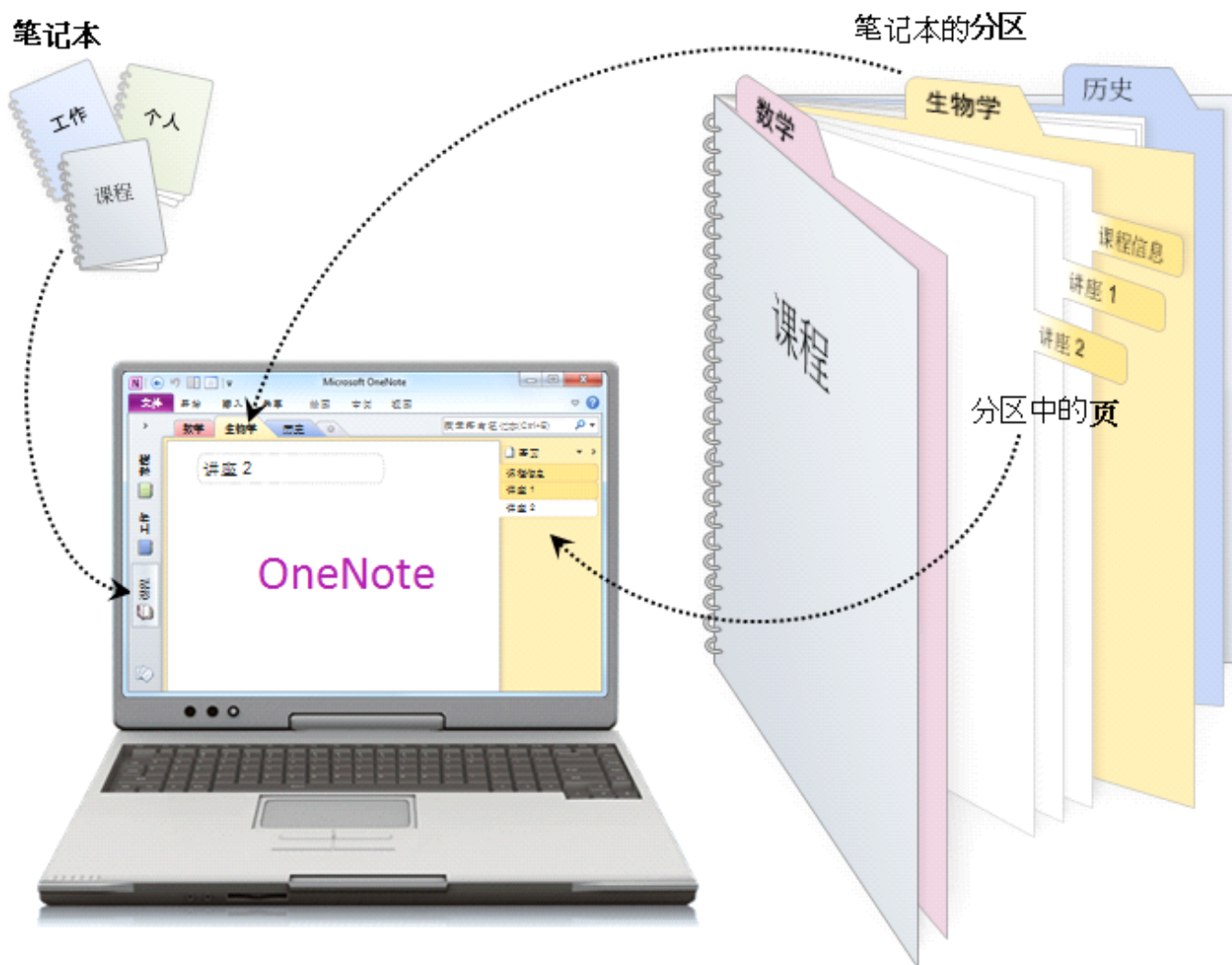
OneNote 页的不同之处在哪里？

您可以在页上的任何位置键入内容。

例如，在此处键入您的姓名：



各个选项卡都有什么用？



在工作中使用 OneNote

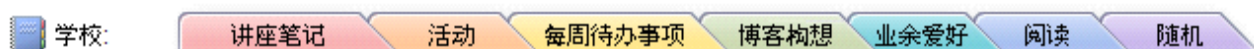




在家中使用 OneNote



在学校使用 OneNote





它是如何工作的? 观看此 1 分钟视频:



[单击此处播放](#)

同时签出右侧的下一页选项卡。

OneNote 基本功能

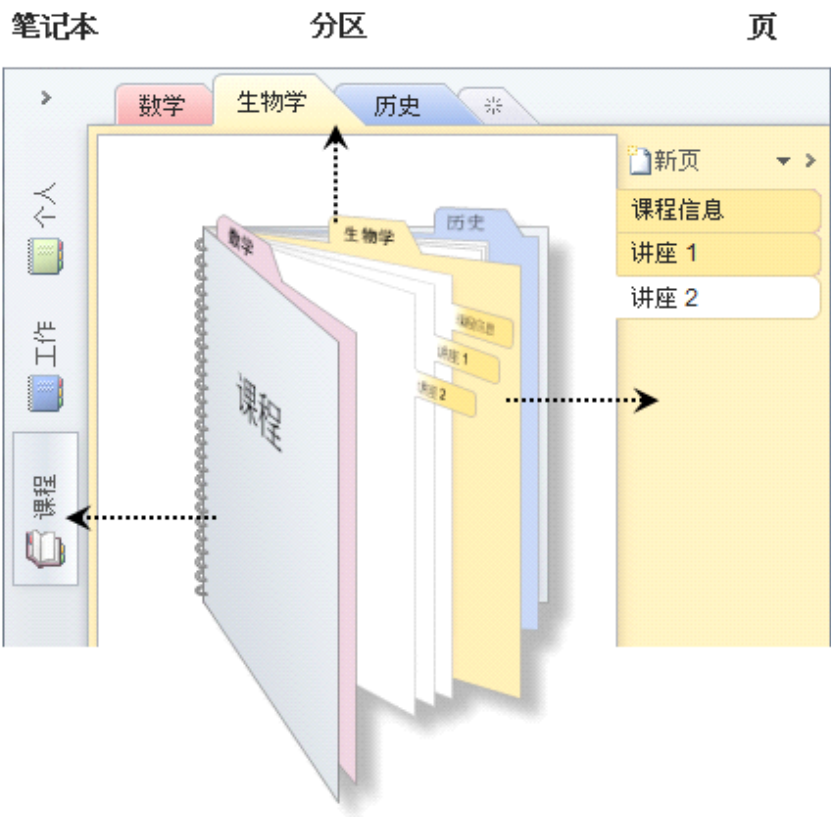


记录多种类型的笔记

查看“开始”>和“插入”>



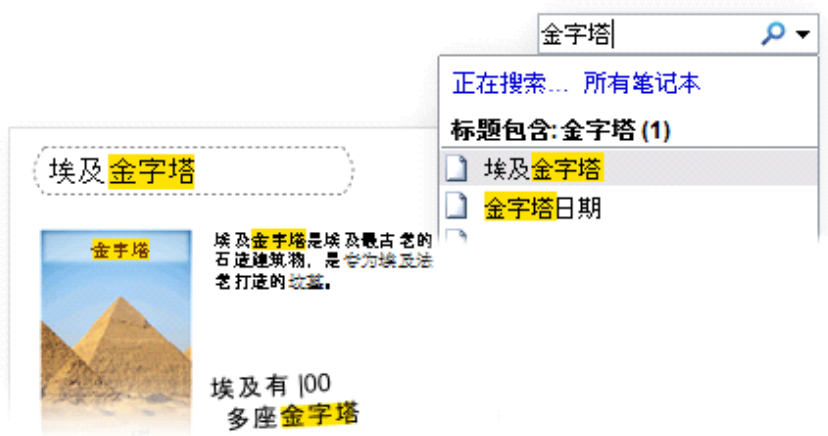
OneNote 可自动保存，因此您不会丢失任何内容。



组织

可创建无限个笔记本、分区和页面。

新建笔记本：
文件 > 新建



搜索

快速跳转至收藏页。

查找手写笔记
或图片中的文本！


与自己或他人分享




共享 >



随处访问您的笔记!

 自动同步。脱机时仍可编辑。

 小组笔记本示例:

会议笔记 | 项目信息 | 工作项目 | 培训 | 崇

小组总体状态

人员	工作项目	截止时间	状态
Jim	文稿撰写	10月21日	正在进行
Ann	摘要	10月25日	概念型 各位, 请在 此处 上签收!
Tom	市场	11月15日	未开始

高级计划
小组总体状态
详细信息
备注
概念型

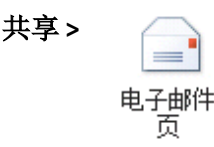
顶级使用

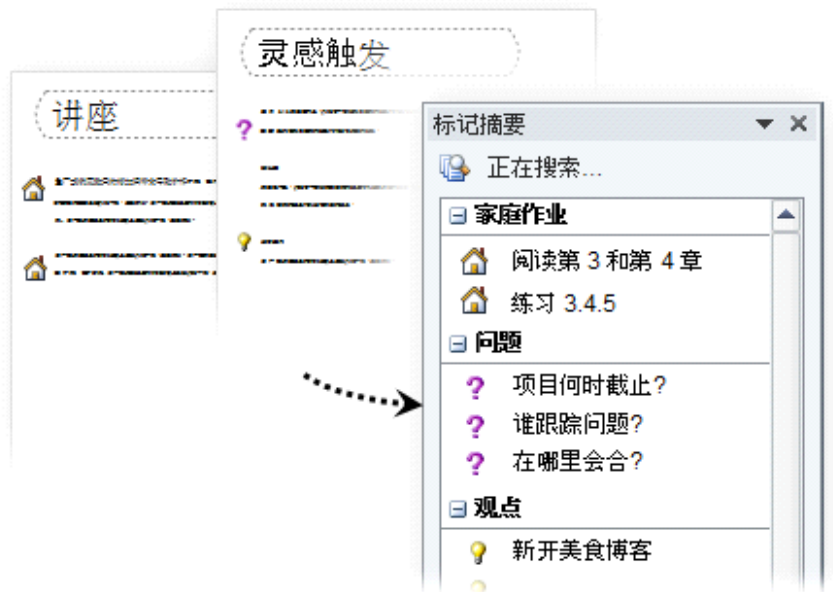


收集产品屏幕剪辑
剪辑地图、图表、汽车照片、天气、结婚礼服等。



通过电子邮件发送会议笔记、说明、状态、工作项目等。
收件人不需要 OneNote。



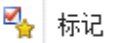


标记并查找重要项目

作业、意见、计费项目、客户请求、待办事项等。

您可以自定义您自己的标记。

右键单击某个笔记 >



标记

开始 >



查找
标记



从 Web 收集文章

新闻、研究、引述等。

OneNote 保留链接。

单击鼠标右键 >



发送至 OneNote



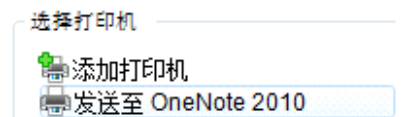
将项目文档与笔记放在一起



对打印样式的讲座幻灯片做批注

打印 OneNote 中对演示文稿、文档或其他内容所做的注释或批注。

从任何应用程序打印 >



将关键项目电子邮件与您的笔记一起保留

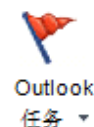
发送电子邮件至 OneNote。



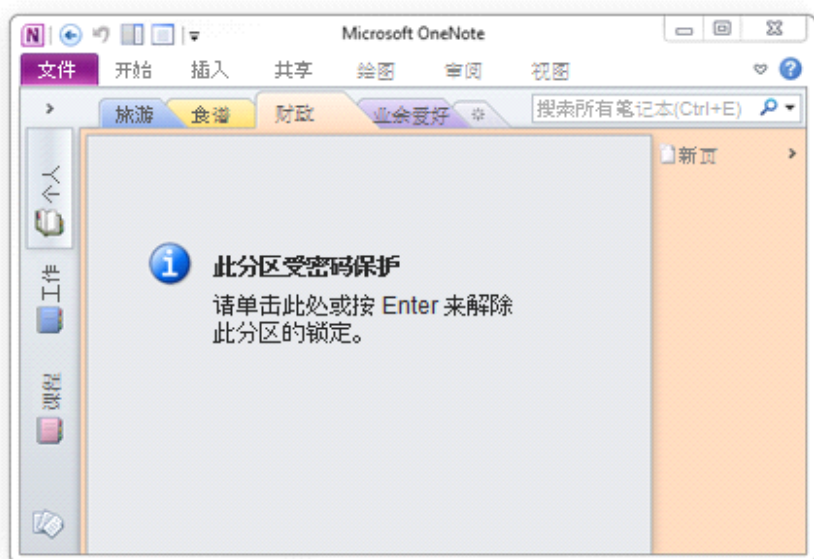
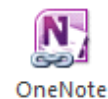


将会议笔记和任务与 **Outlook** 链接

开始 >



在 Outlook 中 >

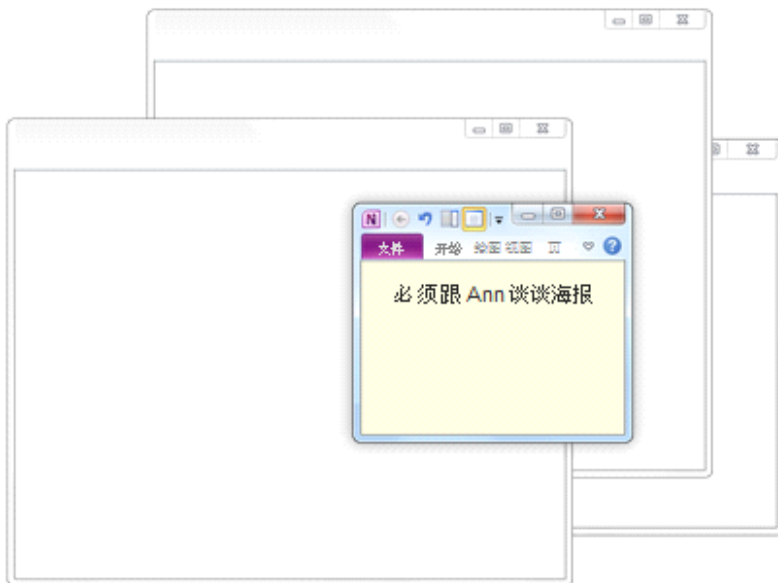


保护敏感信息

银行账户信息、网站登录、私人日记等。

右键单击分

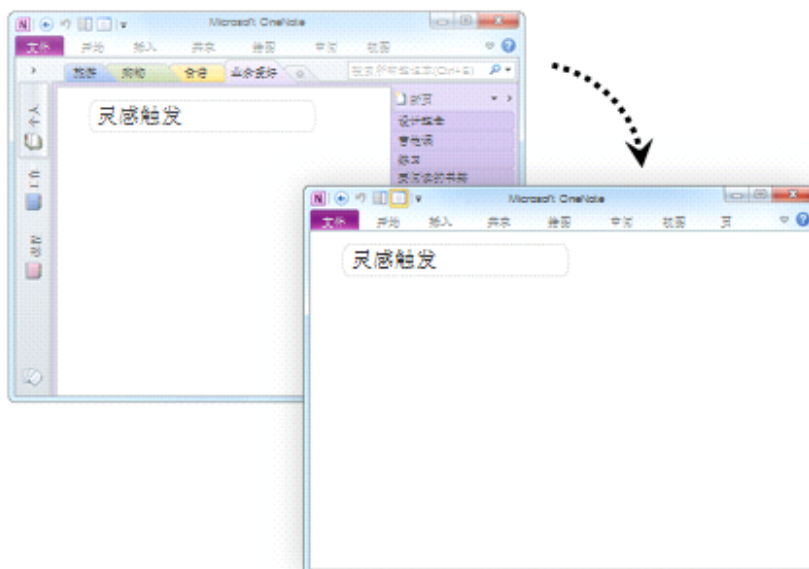
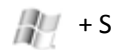
区选项卡 > 密码保护



快速记笔记

进入“未归档笔记”分区。

按住：

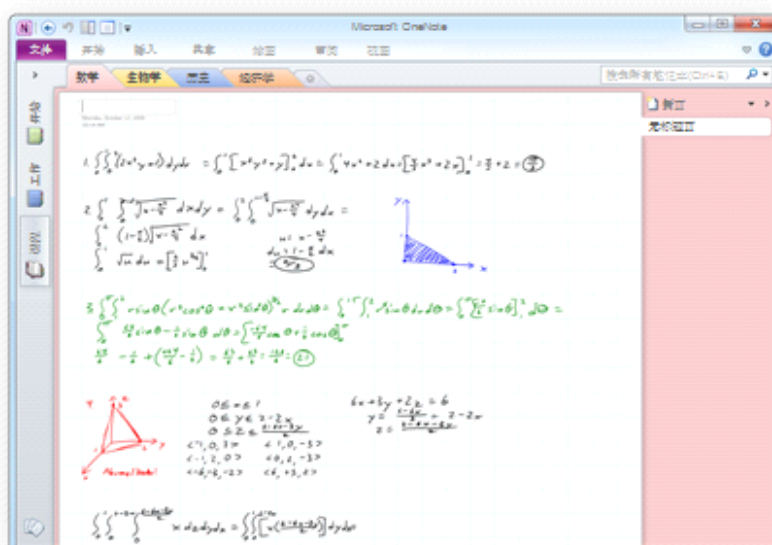


集体讨论不分心

视图 >



整页视图



手写课程或 Tablet PC 上的会议笔记

您甚至可以搜索手写，或将其转换为文本。

其他提示

- 只需键入一些文字并按 Tab，即可创建**表格**。
- **计算器**：在表达式的 “=” 后键入空格。例如：321*45+876=
- 尝试“新页”下拉菜单中的**页模板**。制作适用于会议或项目的自己的模板。
- 为会议**录音**，与您的笔记同步，或将音频或视频拖至页上。

新增功能

2010 中的新功能:

收集、组织和搜索

- 更好地组织页选项卡:
 - 多层子页
 - 折叠子页
- 通过快速搜索, 跳转至任何页
- 停靠到桌面
- 链接到其他笔记, 如 Wiki
- 用于制作标题的快速样式
- 将笔记自动链接到网页和文档
- 插入数学
- “在 Outlook 上做笔记” 任务
- 将内容发送至 OneNote 中的任何分区

共享和通用访问

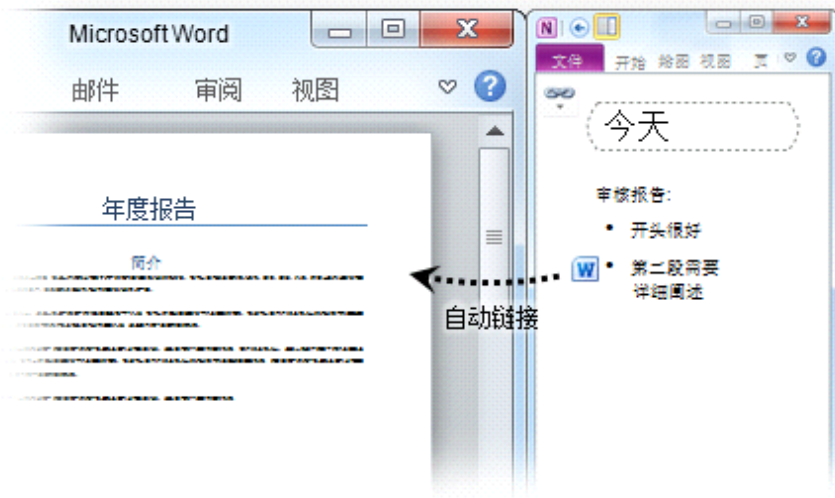
- 从任何位置访问:
 - 在 Web 上共享
 - 在浏览器中查看并编辑
 - 将笔记与 OneNote Mobile 同步
- 共享笔记:
 - 突出显示未读更改
 - 查看作者名缩写
 - 版本历史记录
 - 查找最近编辑
 - 按作者查找编辑
 - 更快与 SharePoint 同步

示例:



使用子页组织主题

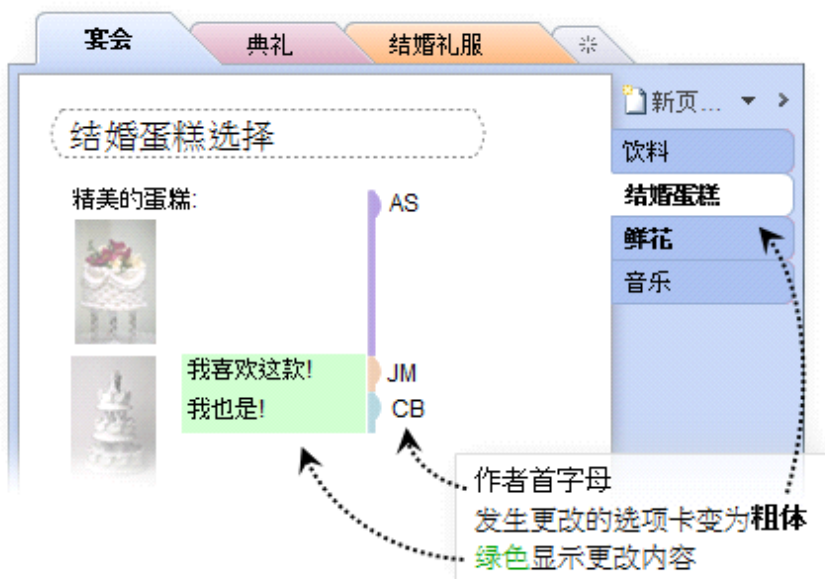
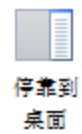
拖动选项卡以缩进页并组织分区内的页。



保持笔记在执行其他任务时可见

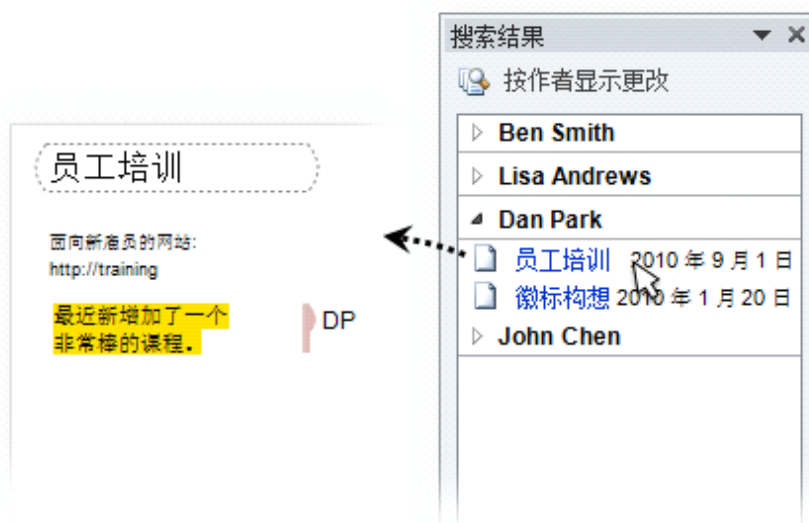
OneNote 将笔记链接到您查看的文档和网页。

视图 >



共享笔记本的新增功能

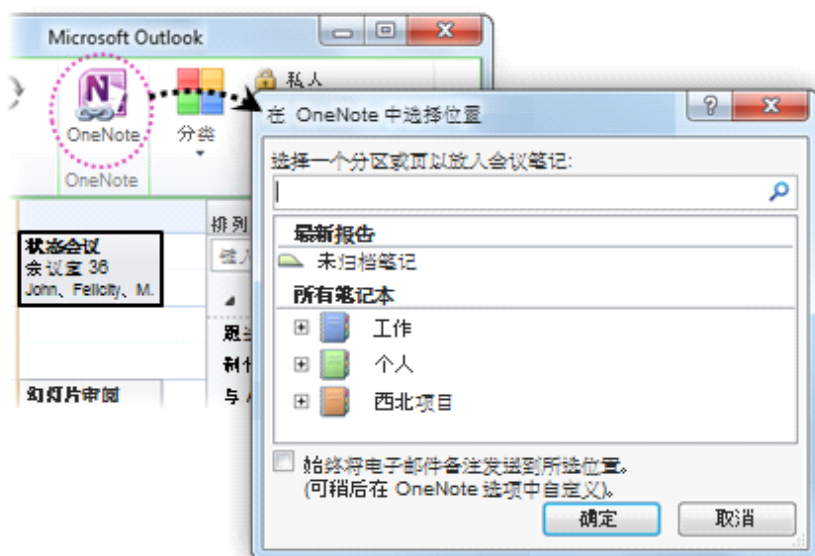
自动显示未读更改。



队友正在处理什么笔记?

共享 >

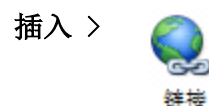




发送至 **OneNote** 时选择位置
从 Outlook 或 Internet Explorer 发送时



链接到自己和他人的信息



或键入 [[页名称]]

更多联机资源

[视频、模板、培训、帮助和讨论小组。](#)

Microsoft® Office OneNote® 2010 笔记本指南

版权所有 © 2009 Microsoft Corporation。保留所有权利。

此处提及的示例公司、组织、产品、域名、电子邮件地址、徽标、人员、地点和事件均为虚构。并不明示或暗示任何真实的公司、组织、产品、域名、电子邮件地址、徽标、人员、地点或事件。

###nodejs & npm proxy & http-server安装

2015年3月2日
16:20

找书籍神器:
<https://www.gitbook.com/explore?page=1>

谷歌搜索:
<https://www.souguge.net/>

张旭鑫es5新增特性:
<http://www.zhangxinxu.com/wordpress/2013/04/es5%E6%96%B0%E5%A2%9F%E6%95%B0%E7%BB%84%E6%96%B9%E6%B3%95/>

数组学习:
https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global_Objects/Array
Express:
<http://www.expressjs.com.cn/starter/installing.html>
举个栗子:
<http://www.expressjs.com.cn/starter/hello-world.html>

大黄总结:
<http://huangtengfei.com/archives/>

chrome浏览器小技巧干货:
<http://www.w3cplus.com/tools/dev-tips.html>

有关angular常问问题:
<https://github.com/huangtengfei/blog/issues/14>

- node安装

- + 下载[node安装包](<http://nodejs.org/download/>)
- + 安装
- + 设置环境变量 ./nodejs

- npm proxy设置

- + cmd 打开命令行
- + 输入 npm, 查看该命令是否存在, 如果不存在, 说明上一步没有nodejs没有安装或环境变量没有设置

- + 在控制台输入下面命令:

```
`$ npm config set proxy http://10.19.110.31:8080`
```

```
`$ npm config set https-proxy http://10.19.110.31:8080`
```

```
C:\Users\15041622>npm config set proxy http://10.19.110.31:8080
C:\Users\15041622>npm config set https-proxy http://10.19.110.31:8080
C:\Users\15041622>npm config list
; cli configs
user-agent = "npm/2.7.4 node/v0.12.2 win32 x64"

; userconfig C:\Users\15041622\npmrc
https-proxy = "http://10.19.110.31:8080/"
proxy = "http://10.19.110.31:8080/"
prxy = "http://10.19.110.31:8080"

; builtin config undefined
prefix = "C:\Users\15041622\AppData\Roaming\npm"

; node bin location = D:\Program Files\nodejs\node.exe
; cwd = C:\Users\15041622
; HOME = C:\Users\15041622
; 'npm config ls -l' to show all defaults.
```

- http-server安装

- > http-server只是一个机遇nodejs开发的小组件功能。利用nodejs的http服务在自己本地快速创建服务器

- + cmd命令输入 npm install -g http-server

```
C:\Users\15041622>npm install -g http-server
C:\Users\15041622\AppData\Roaming\npm\http-server -> C:\Users\15041622\AppData\Roaming\npm\node_modules\http-server\bin\http-server
C:\Users\15041622\AppData\Roaming\npm\hs -> C:\Users\15041622\AppData\Roaming\npm\node_modules\http-server\bin\http-server
http-server@0.8.0 C:\Users\15041622\AppData\Roaming\npm\node_modules\http-server
├── opener@1.4.1
├── corser@2.0.0
├── colors@1.0.3
├── optimist@0.6.1 <wordwrap@0.0.3, minimist@0.0.10>
├── union@0.4.4 <q@2.3.3>
├── portfinder@0.4.0 <async@0.9.0, mkdirp@0.5.1>
├── http-proxy@1.11.2 <eventemitter3@1.1.1, requires-port@0.1.1>
└── ecstatic@0.7.6 <url-join@0.0.1, mime@1.3.4, minimist@1.2.0, he@0.5.0>
```

http-server的使用

- cmd控制台找到需要部署服务器的目录, 在该目录下输入命令: http-server即可

```
C:\Users\15041622>http-server -p 8081
Starting up http-server, serving ./ on: http://0.0.0.0:8081
Hit CTRL-C to stop the server
```

css笔记

2015年4月22日
17:10

边框样式border-style:dotted solid double dashed;

源文档 <http://www.w3school.com.cn/cssref/pr_border-style.asp>

- 上边框是点状
- 右边框是实线
- 下边框是双线
- 左边框是虚线

position有4个属性:static relative absolute fixed,

默认的时候是static即静止定位,是按照元素在(x)html出现的顺序依次分配位置,static是不受top right left bottom影响的.

当父层a的position设为relative(相对定位) 子层b的position设为(绝对定位)时,b的偏移是相对于父层a来定位,例如top:10px的偏移量就使b的顶部和a的顶部有10px的距离.

但是如果a没有设置relative 而b设为absolute,b的定位就是相对于浏览器进行,此时top:10px会使b出现在离浏览器上方10px的地方.自己试下就最清楚了.

z-index

当你定义的CSS中有position属性值为absolute、relative或fixed,

用z-index此取值方可生效。

此属性参数值越大，则被层叠在最上面。

源文档 <http://zhidao.baidu.com/link?url=rK3PdDTtjaCqT2hKVfATRkzQsMkCOXv7ufsga6Q31Y_Gy4qdQx8KVL9DUTKrnzydFr1fsOclkh91eReHQLytq_>>

HTML:target=_blank

Ctrl+shift+p

sublime的快捷方式，查找需要的插件

_blank -- 在新窗口中打开链接

_parent -- 在父窗体中打开链接

_self -- 在当前窗体打开链接,此为默认值

_top -- 在当前窗体打开链接，并替换当前的整个窗体(框架页)

一个对应的框架页的名称 -- 在对应框架页中打开

源文档 <http://zhidao.baidu.com/link?url=dVBqh6lVp9W4jjlvZFmGm9LjVbkkQQJicldNQ4Ojou6mAcrMQi8lRrIfVMQte5x9cW78_wPvb0-4ddxnMnT_>>

图解css3

2015年4月27日
11:23

css选择器：基本选择器，层次选择器，伪类选择器（动态伪类选择器，目标~，语言~，UI~，结构~，否定~），伪元素，属性选择器。

基本选择器：通配，元素，id,类，群组

问题：

动态伪类选择器不是很了解用处
其实就类似于鼠标移动到一个菜单栏之后想要改变的样式，用的比较多的就是

表 2-13 结构伪类选择器使用语法

选择器	功能描述
E:first-child	作为父元素的第一个子元素的元素 E。与 E:nth-child(1) 等同
E:last-child	作为父元素的最后一个子元素的元素 E。与 E:nth-last-child(1) 等同
E:root	选择匹配元素 E 所在文档的根元素。在 HTML 文档中，根元素始终是 html，此时该选择器与 html 类型选择器匹配的内容相同
E F:nth-child(n)	选择父元素 E 的第 n 个子元素 F。其中 n 可以是整数（1、2、3）、关键字（even、odd）、可以是公式（2n+1、-n+5），而且 n 值起始值为 1，而不是 0

选择器	功能描述
E F:nth-last-child(n)	选择元素 E 的倒数第 n 个子元素 F。此选择器与 E F:nth-child(n) 选择器计算顺序刚好相反，但使用方法都是一样的，其中 :nth-last-child(1) 始终匹配的是最后一个元素，与 :last-child 等同
E:nth-of-type(n)	选择父元素内具有指定类型的第 n 个 E 元素
E:nth-last-of-type(n)	选择父元素内具有指定类型的倒数第 n 个 E 元素
E:first-of-type	选择父元素内具有指定类型的第一个 E 元素，与 E:nth-of-type(1) 等同
E:last-of-type	选择父元素内具有指定类型的最后一个 E 元素，与 E:nth-last-of-type(1) 等同
E:only-child	选择父元素只包含一个子元素，且该子元素匹配 E 元素
E:only-of-type	选择父元素只包含一个同类型的子元素，且该子元素匹配 E 元素
E:empty	选择没有子元素的元素，而且该元素也不包含任何文本节点

大家只需要记住一点：“:nth-child”选择的是某父元素的子元素，这个子元素并没有指定确切的类型，同时满足两个条件时方能有效果，其一是子元素，其二此子元素刚好好处在那个位置；“:nth-of-type”选择的是某父元素的子元素，而且这个子元素是指定类型。

选择器的优先权】

解释：

- 1. 内联样式表的权值最高 1000;
 - 2. ID 选择器的权值为 100
 - 3. Class 类选择器的权值为 10
 - 4. HTML 标签选择器的权值为 1
- A 选择器都有一个权值，权值越大越优先；
- B 当权值相等时，后出现的样式表设置要优于先出现的样式表设置；
- C 创作者的规则高于浏览者：即网页编写者设置的CSS 样式的优先权高于浏览器所设置的样式；
- D 继承的CSS 样式不如后来指定的CSS 样式；
- E 在同一组属性设置中标有 “!important”规则的优先级最大；

font:bold 30px/50px arial;（前面的30是size大小，后面的50表示line-height行高）

源文档 <<http://www.cnblogs.com/xugang/archive/2010/09/24/1833760.html>>

什么是内联对象，什么是块对象？

所有可视的文档对象都是块对象(block element)或者内联对象(inline element)。例如，div 是一个块对象，span 是一个内联对象。

块对象的特征是从新的一行开始且能包含其他块对象和内联对象。从新的一行开始：比如div就是前后断行；

内联对象被呈递时不会从新行开始，能够包含其他内联对象和数据。内联可以这样理解：从新的一行开始，直接从内容里面，接着往后走。。。是指它能被别的对象内联。。。。

源文档 <<http://zhidao.baidu.com/link?url=9vv4L2bNGnvFv7XbOKfHuoAdLD3RZ2iud5Hhb7d-otEch-feg21Wlkwq34wPrf69HydXHxVcEA4XP7oek7rja>>

需要记忆的属性

2015年4月28日
9:31

border-radius（圆角的设置）：

例子 2

```
border-radius: 2em 1em 4em / 0.5em 3em;
```

等价于：

```
border-top-left-radius: 2em 0.5em;  
border-top-right-radius: 1em 3em;  
border-bottom-right-radius: 4em 0.5em;  
border-bottom-left-radius: 1em 3em;
```

display:inline-block

简单来说就是将对象呈递为内联对象，但是对象的内容作为块对象呈递。旁边的内联对象会被呈递在同一行内，允许空格。但是这个属性目前不是所有的浏览器都支持，只有Opera和Safari支持，FireFox3和IE8据说将会支持，Firefox2和IE使用特殊办法可以实现这种效果，下面就来探讨一下

在Firefox2中有-moz-inline-stack和-moz-inline-box实现，但是这两个私有属性在某些情况下都会有异常，具体如下：

1、display:-moz-inline-stack

“当它所应用的外包装器（wrapper）的display为inline的时候，它所包含的a或button将无法点击或无法选取，需要通过position:relative还hack掉这个bug” --乌龙茶

2、display:-moz-inline-box

使用这一属性后，text-align就会出问题，必须使用Firefox的私有属性 -moz-box-align来解决

所以，建议不要使用-moz-inline-box，还是使用-moz-inline-stack

在IE（以下IE8除外）下，如果对内联元素，比如a或span元素，使用inline-block属性是有效的，似乎IE是支持的，其实是触发了IE的layout，从而拥有了inline-block属性的表症。这样我们就有了一种在IE下实现display:inline-block效果的两种方法：

1、先用display:inline-block属性触发块元素，然后再定义display:inline，让块元素呈递为内联对象（原理：这是IE的一个经典bug，如果先定义了display:inline-block，然后再设置display回inline或block，layout不会消失），代码如下：

```
div {display:inline-block;}
```

```
div {display:inline;}
```

2、直接让块元素设置为内联对象(display:inline)，然后通过zoom:1触发块元素的layout，代码如下：

```
div {display:inline; zoom:1;}
```

那么为了能够让所有浏览器支持display:inline-block，综合一下，最终的实现代码如下（参考怪飞：模拟兼容性的 inline-block 属性）：

display:inline-block; /*一方面Firefox3 beta、IE8 beta、Opera、Safari 支持，另一方面下触发IE下inline 元素的 hasLayout*/

display:-moz-inline-stack; /* Firefox 的私有属性，需要时还必须用到position:relative解决上面提到的bug */

zoom:1; /*同样是IE 下触发 hasLayout*/

*display:inline; /*一旦IE 下触发了 hasLayout，设置 block 元素为 inline 会使 display:inline 效果与 display:inline-block 相似*/

/*扩展一下一些其他可能用到的属性*/

text-indent:-9999px;

*text-indent:0;

font-size:0;

line-height:0; /* 如需隐藏文字，可用这四个属性 */

/*另外上面隐藏文字，还可以用更简化的方法：line-height:超级大值;font-size:0; */

overflow:hidden; /* 隐藏溢出的内容 */

vertical-align:middle; /* 行内垂直居中，针对 Opera 比较大的偏差 */

width: ? px; /* ? 为任意非auto值*/

list-style:

三个属性:

指定以下值（任意次序，最多三个）的字符串：

图像	可供 list-style-image 属性使用的图像值的任意范围。
位置	可供 list-style-position 属性使用的位置值的任意范围。
类型	可供 list-style-type 属性使用的类型值的任意范围。

对于<a>这两个标签——因为这两个标签非块元素，如果不用display：block定义一下，那么定义width、height等和长宽相关的css属性时会发现完全不生效。

源文档 <http://zhidao.baidu.com/link?url=70DJYKaD6SeWX-0HEqGZZ59uhJyIz6U-7kZtsI27_x8pX8SbbxiXEeU-rBbfwhKxcmmN1qu3991UI3amQ3VPK->

text-decoration下划线，设为none就取消掉

box-shadow

inset：阴影类型，可选值。如果不设置，其默认的投影方式是外阴影；如果取其唯一值“inset”，就是给元素设置内阴影。
x-offset：阴影水平偏移量，其值可以是正负值。如果取正值，则阴影在元素的右边，反之取负值，阴影在元素的左边。
y-offset：阴影垂直偏移量，其值可以是正负值。如果取正值，则阴影在元素的底部，反之取负值，阴影在元素的顶部。
blur-radius：阴影模糊半径，可选参数。其值只能是正值，如果取值为“0”时，表示阴影不具有模糊效果，如果取值越大，阴影的边缘就越模糊。
spread-radius：阴影扩展半径，可选参数。其值可以是正负值，如果取值为正值，则整个阴影都延展扩大，反之取值为负值，则整个阴影都缩小。

box-shadow: inset 4px 4px 4px 4px grey;
多层阴影（图解css3 p128）

background-origin

padding-box(图片相对于内边距)
border-box（图片相对于边框盒）
content-box（图片相对于内容框）

源文档 <http://www.w3school.com.cn/titv/c.asp?c=css_background-origin>

文本类型:

word-spacing: normal; /*词与词之间的间距*/
letter-spacing: normal; /*字符之间的间距*/
vertical-align: baseline; /*文本垂直对齐方式*/
text-decoration: none; /*文本修饰线*/
text-indent: 2px; /*文本首行缩进*/
text-align: left; /*文本水平对齐方式*/
line-height: 20px; /*文本行高*/
text-transform: uppercase; /*文本大小写*/
text-shadow: none; /*文本阴影效果*/
white-space: normal; /*文字之间和文本之间的空白符间距*/
direction: ltr; /*文本流入方向，默认从左到右*/

溢出文本属性:

text-overflow:clip是不显示省略号标记; ellipsis文本溢出显示省略号; 然而，ellipsis需要配合white-space: nowrap强制一行显示;和overflow: hidden;溢出内容隐藏配合使用。同时要明确给需要截取文本的容器设置宽度值。

CSS3盒模型

box-sizing:

content-box为默认值，元素维持标准盒模型。
border-box:重新定义盒模型组成的模式（可以拯救布局）
inherit：使元素继承父元素的盒模型模式。

CSS3内容溢出属性:

overflow-x: visible hidden scroll auto no-display no-content;	overflow-y: visible hidden scroll auto no-display no-content;
值	描述
visible	不裁剪内容，可能会显示在内容框之外。
hidden	裁剪内容 - 不提供滚动机制。
scroll	裁剪内容 - 提供滚动机制。
auto	如果溢出框，则应该提供滚动机制。
no-display	如果内容不适合内容框，则删除整个框。
no-content	如果内容不适合内容框，则隐藏整个内容。

overflow: hidden; /* 隐藏溢出的内容 */

vertical-align: middle; /* 行内垂直居中，针对 Opera 比较大的偏离 */

width: ? px; /* ? 为任意非auto值*/

height: ? px; /* ? 为任意非auto值*/

源文档 <http://zhidao.baidu.com/link?url=s3_90_vqt5FhAqvY5YX6cwL2Un5-9FgZ2BPX1XaoQ07GPjTbNYpxdfdbeG1GdbizlgDcnQ9anVtiI7g-c5-zXa>

border边框属性有三个子属性：线宽 线型 颜色。

border-width, border-style, border-color 定义

border-style : solid 是实线的意思

border-style: dotted solid double dashed (按照顶边，右边，下边，左边的顺序)；

- 上边框是点状
- 右边框是实线
- 下边框是双线
- 左边框是虚线

源文档 <<http://zhidao.baidu.com/link?url=X5YPqUOMVlQu-WH44m2NQGc44RJ9harcncfAAOHBlZFe3agx9tK1y0dkiugkpKHGNmJYqR60ASyP-wT2l5fpa>>

表 3-1 border-style 值列表

属性值	功能描述	示例代码	效果
none	定义无边框	.elm {border:none;}	none
hidden	与“none”相同。不过应用于表时除外，对于表，hidden 用于解决边框冲突	.elm {border:hidden;}	hidden
dotted	定义点状边框	.elm {border:3px dotted red ;}	<div>dotted</div>
dashed	定义虚线边框	.elm {border:3px dashed red;}	<div>dashed</div>
solid	定义实线边框	.elm {border:3px solid red;}	<div>solid</div>
double	定义双线。双线的宽度等于 border-width 的值	.elm {border:3px double red;}	<div>double</div>
groove	定义 3D 凹槽边框，其效果取决于 border-color 的值	.elm {border:3px groove red;}	<div>groove</div>
ridge	定义 3D 垄状边框，其效果取决于 border-color 的值	.elm {border:3px ridge red;}	<div>ridge</div>
inset	定义 3D inset 边框，其效果取决于 border-color 的值	.elm {border:8px inset red;}	<div>inset</div>
outset	定义 3D outset 边框，其效果取决于 border-color 的值	.elm {border:8px outset red;}	<div>outset</div>
inherit	规定应该从父元素继承边框样式。部分浏览器不支持这个属性值		

其实border-radius的水平和垂直半径也遵循前面介绍的规则，可以设置1~4个值的集合，同时它们分别遵循CSS赋值规则。但分开设置元素各个顶角的圆角的水平和垂直半径圆角效果时，不需要“/”，如下所示。

```
border-top-left-radius: 10px 50px;
border-top-right-radius: 20px 60px;
border-bottom-left-radius: 20px 60px;
border-bottom-right-radius: 30px 50px;
```

表格应用圆角

另外表格元素table使用border-radius是不一样的，当表格样式属性border-collapse是collapse时，表格不能正常显示，只有border-collapse属性值为separate时，表格圆角才能正常显示。

no-display	如果内容不适合内容框，则删除整个框。
no-content	如果内容不适合内容框，则隐藏整个内容。

CSS3自由缩放属性：
resize : none | both | horizontal | vertical | inherit

none: UserAgent没提供尺寸调整机制，用户不能操纵机制调节元素的尺寸。
both: UserAgent提供双向尺寸调整机制，让用户可以调节元素的宽度和高度。
horizontal: UserAgent提供单向水平尺寸调整机制，让用户可以调节元素的宽度。
vertical: UserAgent提供单向垂直尺寸调整机制，让用户可以调节元素的高度。
inherit: 默认css继承。

a标签居中：
a标签外层加一个DIV 然后DIV 设置 样式 text-align:center; 这样里面就居中了，这是第一种思路，第二种思路的话，就直接在你的源码上改 .a{text-decoration:none;font-size:20px;font-weight:bold;width:222px;height:100px;border:1px solid red;display:block; text-align:center} 这个就行了，因为A标签不是一个块级元素 所以 要先 display:block

问题

2015年5月5日
20:39

想要图片垂直居中:

- 1、可以先对图片的盒子, 设置行高等于当前盒子的高度
- 2、然后给图片设置 vertical-align: middle;
- 2、然后给图片设置 vertical-align: middle;

```
{ display: none; /* 不占据空间, 无法点击 */ }
{ visibility: hidden; /* 占据空间, 无法点击 */ }
{ position: absolute; top: -999em; /* 不占据空间, 无法点击 */ }
{ position: relative; top: -999em; /* 占据空间, 无法点击 */ }
{ position: absolute; visibility: hidden; /* 不占据空间, 无法点击 */ }
{ height: 0; overflow: hidden; /* 不占据空间, 无法点击 */ }
{ opacity: 0; filter: Alpha(opacity=0); /* 占据空间, 可以点击 */ }
{ position: absolute; opacity: 0; filter: Alpha(opacity=0); /* 不占据空间, 可以点击 */ }
zoom: 0.001;
-moz-transform: scale(0);
-webkit-transform: scale(0);
-o-transform: scale(0);
transform: scale(0);
/* IE6/IE7/IE9不占据空间, IE8/FireFox/Chrome/Opera 占据空间。都无法点击 */
position: absolute;
zoom: 0.001;
-moz-transform: scale(0);
-webkit-transform: scale(0);
-o-transform: scale(0);
/* 不占据空间, 无法点击 */ }
```

源文档 <http://zhidao.baidu.com/link?url=dlnX1MemYnDgfiRXREy1Vta-z1QRyDXvZRC9gKHepQDjaKVMvXlyd6iUFzgvYbphdVHlUqYo9128MIC7vWRHJ_>

有关hover的绑定处理:

JavaScript 下 .onmouseover() 和 onmouseout() 事件给我们的WEB页面交互带来了无数超炫的效果.而在 JQuery里.取而代之的是 hover(over,out)事件.

用法:

```
$("#id").hover(

function(){

//当鼠标放上去的时候,程序处理

},

function(){

//当鼠标离开的时候,程序处理

});
```

源文档 <<http://www.douban.com/note/202404884/>>

sit环境账号: <http://wjhxxsit.cnsuning.com/mpcp-web/home.htm>
易购账号: hqq2007happy@126.com a123

111111_happy2子账号: 111111_happy2 409102

易购账号: hqq2007happy@126.com a123456

111111_happy2 409102

问题描述:

浏览器不兼容问题:

@-moz-document url-prefix() { .selector { property: value; } }

例如:@-moz-document url-prefix(){
.service-btn {
margin-top: 19px;

}
}

Position:

- 1、relative :这个是相对节点本身的位移 (如果设置top:10px。就是离原先位置10px高)
- 2、fixed :是相对整个视窗的位移
- 3、absolute:先找寻父节点中含有relative属性的节点, 相对此节点位移, 若没有该属性, 则相对body位移。

有关换行, 写word-wrap: break-word;

打开新窗口 (window.open)

URL: 可选参数，在窗口中要显示网页的网址或路径。如果省略这个参数，或者它的值是空字符串，那么窗口就不显示任何文档。

窗口名称: 可选参数，被打开窗口的名称。

- 1. 该名称由字母、数字和下划线字符组成。
- 2. “_top”、“_blank”、“_self”具有特殊意义的名称。
 - _blank: 在新窗口显示目标网页
 - _self: 在当前窗口显示目标网页
 - _top: 框架网页中在上部窗口中显示目标网页
- 3. 相同 name 的窗口只能创建一个，要想创建多个窗口则 name 不能相同。
- 4. name 不能包含有空格。

参数字符串: 可选参数，设置窗口参数，各参数用逗号隔开。

源文档 <<http://www.imooc.com/code/412>>

参数	值	说明
top	Number	窗口顶部离开屏幕顶部的像素数
left	Number	窗口左端离开屏幕左端的像素数
width	Number	窗口的宽度
height	Number	窗口的高度
menubar	yes, no	窗口有没有菜单
toolbar	yes, no	窗口有没有工具条
scrollbars	yes, no	窗口有没有滚动条
status	yes, no	窗口有没有状态栏

事件表:

事件	说明
onclick	鼠标单击事件
onmouseover	鼠标经过事件
onmouseout	鼠标移开事件
onchange	文本框内容改变事件
onselect	文本框内容被选中事件
onfocus	光标聚集
onblur	光标离开
onload	网页导入
onunload	关闭网页

Date对象中处理时间和日期的常用方法：

方法名称	功能描述
get/setDate()	返回/设置日期
get/setFullYear()	返回/设置年份，用四位数表示
get/setYear()	返回/设置年份
get/setMonth()	返回/设置月份。 0:一月...11:十二月，所以加一。
get/setHours()	返回/设置小时，24小时制。
get/setMinutes()	返回/设置分钟数
get/setSeconds()	返回/设置秒钟数
get/setTime()	返回/设置时间（毫秒为单位）

getDay() 返回星期，返回的是0-6的数字，0表示星期天。如果要返回相对应“星期”，通过数组完成

get/setTime() 返回/设置时间，单位毫秒数

知识讲解：

split() 方法将字符串分割为字符串数组，并返回此数组。

语法：

```
stringObject.split(separator,limit)
```

参数说明:

参数	描述
separator ^❷	必需。从该参数指定的地方分割 stringObject。❶
limit ^❷	可选参数，分割的次数，如设置该参数，返回的子串不会多于这个参数指定的数组，如果无此参数为不限次数❶

Math 对象方法

方法❶	描述❶
abs(x)❶	返回数的绝对值。❶
acos(x)❶	返回数的反余弦值。❶
asin(x)❶	返回数的反正弦值。❶
atan(x)❶	返回数字的反正切值❶
atan2(y,x)❶	返回由 x 轴到点(x,y)的角度(以弧度为单位)❶
ceil(x)❶	对数进行上舍入。❶
cos(x)❶	返回数的余弦。❶
exp(x)❶	返回 e 的指数。❶
floor(x)❶	对数进行下舍入。❶
log(x)❶	返回数的自然对数（底为 e）。❶
max(x,y)❶	返回 x 和 y 中的最高值。❶
min(x,y)❶	返回 x 和 y 中的最低值。❶
pow(x,y)❶	返回 x 的 y 次幂。❶
random()❶	返回 0 ~ 1 之间的随机数。❶
round(x)❶	把数四舍五入为最接近的整数。❶
sin(x)❶	返回数的正弦。❶
sqrt(x)❶	返回数的平方根。❶
tan(x)❶	返回角的正切。❶
toSource()❶	返回该对象的源代码。❶
valueOf()❶	返回 Math 对象的原始值。❶

window对象是BOM的核心，window对象指当前的浏览器窗口。

window对象方法:

源文档 <<http://www.imooc.com/code/1146>>

数组方法:

方法	描述
concat()	连接两个或更多的数组，并返回结果。❶
join()	把数组的所有元素放入一个字符串。元素通过指定的分隔符进行分隔。❶
pop()	删除并返回数组的最后一个元素。
push()	向数组的末尾添加一个或更多元素，并返回新的长度。❶
reverse()	颠倒数组中元素的顺序。❶
shift()	删除并返回数组的第一个元素。
slice()	从某个已有的数组返回选定的元素。
sort()	对数组的元素进行排序。
splice()	删除元素，并向数组添加新元素。❶
toSource()	返回该对象的源代码。❶
toString()	把数组转换为字符串，并返回结果。❶
toLocaleString()	把数组转换为本地数组，并返回结果。❶
unshift()	向数组的开头添加一个或更多元素，并返回新的长度。❶
valueOf()	返回数组对象的原始值。

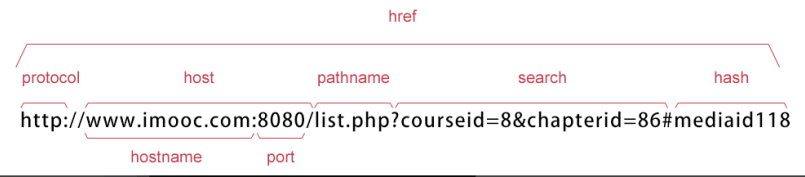
有关slice、substring和substr:

slice和substring接收的是起始位置和结束位置(不包括结束位置),而substr接收的则是起始位置和所要返回的字符串长度。

```
1 var test = 'hello world';
2
3 alert(test.slice(-3));           //rld
4 alert(test.substring(-3));      //hello world
5 alert(test.substr(-3));         //rld
6
7 alert(test.slice(3,-4));         //lo w
8 alert(test.substring(3,-4));    //hel
9 alert(test.substr(3,-4));       //空字符串
```

方法	描述
alert()	显示带有一段消息和一个确认按钮的警告框。
prompt()	显示可提示用户输入的对话框。
confirm()	显示带有一段消息以及确认按钮和取消按钮的对话框。
open()	打开一个新的浏览器窗口或查找一个已命名的窗口。
close()	关闭浏览器窗口。
print()	打印当前窗口的内容。
focus()	把键盘焦点给予一个窗口。
blur()	把键盘焦点从顶层窗口移开。
moveBy()	可相对窗口的当前坐标把它移动指定的像素。
moveTo()	把窗口的左上角移动到一个指定的坐标。
resizeBy()	按照指定的像素调整窗口的大小。
resizeTo()	把窗口的大小调整到指定的宽度和高度。
scrollBy()	按照指定的像素值来滚动内容。
scrollTo()	把内容滚动到指定的坐标。
setInterval()	每隔指定的时间执行代码。
setTimeout()	在指定的延迟时间之后来执行代码。
clearInterval()	取消 setInterval() 的设置。
clearTimeout()	取消 setTimeout() 的设置。

location对象：
location用于获取或设置窗体的URL，并且可以用于解析URL。



属性	描述
hash	设置或返回从井号 (#) 开始的 URL（锚）。
host	设置或返回主机名和当前 URL 的端口号。
hostname	设置或返回当前 URL 的主机名。
href	设置或返回完整的 URL。
pathname	设置或返回当前 URL 的路径部分。
port	设置或返回当前 URL 的端口号。
protocol	设置或返回当前 URL 的协议。
search	设置或返回从问号 (?) 开始的 URL（查询部分）。

属性	描述
assign()	加载新的文档。
reload()	重新加载当前文档。
replace()	用新的文档替换当前文档。

Navigator对象

Navigator 对象包含有关浏览器的信息，通常用于检测浏览器与操作系统的版本。

属性	描述
appName	浏览器代码名的字符串表示。
appName	返回浏览器的名称。
appVersion	返回浏览器的平台和版本信息。
platform	返回运行浏览器的操作系统平台。
userAgent	返回由客户机发送服务器的 user-agent 头部的值。

有关nodejs/fas/gulp/npm

2015年8月6日
20:34

第一个fas，就是个脚手架，能快速搭建一个项目原型,类似
express(<http://www.expressjs.com.cn/guide.html>)

- **scss**, **css**压缩合并
- **css3**自动加前缀
- **css**文件内图片资源根据内容自动加**md5**戳，并报警错误图片路径
- **jslint** **js**语法检测
- **imagemin** 图片压缩
- 资源定位
- 模块化**html**
- 集成了**ssi**的本地**serve**，取代**nginx**
- 数据**mock**

nodejs，就是一个javascript运行环境

nodejs，就是一个javascript运行环境,需要先装**nodejs**,然后装完会同时装好**npm**(**node package manager**,**node**的包管理工具,比如你要装**express**，就要用**npm**)，**node**的管理工具，可以按照各自开发库，
是这么个流程，装了**node**，配好环境变量以后，就相当于在命令行下有了一个**js**开发环境，**express**是基于**js**做的一个**web**框架

gulp，一个构建工具(**grunt**:<http://huangtengfei.com/2015/03/front-end-development-tool-yeoman/>)

DOM文本对象模型

2015年8月18日
17:29

HTML文档可以说由节点构成的集合，DOM节点有:

1. 元素节点: 上图中<html>、<body>、<p>等都是元素节点，即标签。

2. 文本节点: 向用户展示的内容，如...中的JavaScript、DOM、CSS等文本。

3. 属性节点: 元素属性，如<a>标签的链接属性href="http://www.imooc.com"。

节点属性:

方法	说明
nodeName	返回一个字符串，其内容是给定节点的名字。
nodeType	返回一个整数，这个数值代表给定节点的类型。
nodeValue	返回给定节点的当前值。

遍历节点树:

方法	说明
childNodes	返回一个数组，这个数组由给定元素节点的子节点构成
firstChild	返回第一个子节点
lastChild	返回最后一个子节点
parentNode	返回一个给定节点的父节点
nextSibling	返回给定节点的下一个子节点
previousSibling	返回给定节点的上一个子节点

DOM操作:

方法	说明
createElement(element)	创建一个新的元素节点
createTextNode()	创建一个包含着给定文本的新文本节点
appendChild()	指定节点的最后一个子节点列表之后添加一个新的子节点。
insertBefore()	将一个给定节点插入到一个给定元素节点的给定子节点的前面
removeChild()	从一个给定元素中删除一个子节点
replaceChild()	把一个给定父元素里的一个子节点替换为另外一个节点

有关获取Id, name, Tagname的区别:

方法	说明	获得
getElementById	通过指定 id 获得元素	一个
getElementsByName	通过元素名称 name 属性获得元素	一组
getElementsByTagName	通过标签名称获得元素	一组

有关DOM方法里的疑问:

var list = document.getElementsByTagName("div")[0].childNodes;
document.getElementsByTagName("div")所获取到的是一个对象的集合，类数组的集合，所以可以document.getElementsByTagName("div")[0]这样获取第几个节点，但是并不能真正使用数组的方法。

节点属性

在文档对象模型 (DOM) 中，每个节点都是一个对象。DOM 节点有三个重要的属性：

1. nodeName：节点的名称

2. nodeValue：节点的值

3. nodeType：节点的类型

一、nodeName 属性: 节点的名称，是只读的。

1. 元素节点的 nodeName 与标签名相同

2. 属性节点的 nodeName 是属性的名称

3. 文本节点的 nodeName 永远是 #text

4. 文档节点的 nodeName 永远是 #document

二、nodeValue 属性：节点的值

1. 元素节点的 nodeValue 是 undefined 或 null

2. 文本节点的 nodeValue 是文本自身

3. 属性节点的 nodeValue 是属性的值

三、nodeType 属性: 节点的类型，是只读的。以下常用的几种节点类型:

元素类型	节点类型
元素	1
属性	2
文本	3
注释	8
文档	9

日常累积

2015年9月23日
18:02

如果要一个东西在大屏下占一半，小屏占全部，

```
<div class= “col-lg-6 col-sm-12”  
两个都写上，这样拖动屏幕大小，他自己去选择合适的
```

比如，你电脑上想要一行显示4个格子，ipad上想要显示两个，手机显示一个
就是：col-lg-3 col-sm-6 col-xs-12
bootstrap是把col-xs定义写在最前面 其他的都写在@media里面了 难怪说xs不能自适应 又能同时写col-lg col-md col-sm
凡是要做自适应，需要设计的时候给出，不同屏幕长什么样，内部是media query，
(bootstrap是把col-xs定义写在最前面 其他的都写在@media里面了 难怪说xs不能自适应 又能同时写col-lg col-md col-sm)

```
3  
9 @media (min-width: 992px) {  
10     .container {  
11         width:970px  
12     }  
13 }  
14  
15 }
```

> media queries直译即媒体查询。用在html的`<link` 标签或CSS样式内，通过判断媒体类别或特征属性来使对应的CSS生效。
> 该标准的W3官网：<http://www.w3.org/TR/css3-mediaqueries/>

使用方法
常见的使用方式如下：

```
`<link rel="stylesheet" media="screen and (max-width: 400px)" href="small.css" />`
```

```
@media screen and (max-width: 400px) {  
span {  
width: 100%;  
}  
}
```

screen是媒体类型，目前浏览器仅支持'screen'(屏幕式样)、'pring'(打印式样)、'all'。

后面则通过`and`、`only`、`not`、`条件判断，判断内容。

Media features

> Media queries提供了以下特征条件来进行判断
> 绝大多数特征条件都提供`min-`和`max-`来表示`**大于等于**`和`**小于等于**`
>

- width
- 描述输出设备目标显示区域的视口(viewport)宽度
- `@media screen and (min-width: 400px) and (max-width: 700px){...}`
- height
- 同`width`
- device-width
- 描述输出设备本身的宽度
- `@media screen and (min-device-width: 800px) {...}`
- device-height
- 同`device-width`
- orientation
- 描述显示的方向。值`portrait`(纵向)表示高度大于等于宽度，值`landscape`(横向)表示宽度大于高度
- `@media all and (orientation:portrait) {...}`
- `@media all and (orientation:landscape) {...}`
- aspect-ratio
- 描述显示区域纵横比
- device-aspect-ratio
- 描述输出设备屏幕纵横比
- `@media screen and (device-aspect-ratio: 16/9) {...}`
- `@media screen and (device-aspect-ratio: 1280/720) {...}`
- `@media screen and (device-aspect-ratio: 2560/1440) {...}`
- color
- 描述屏幕支持的颜色位数，如果不是彩色设备，则值为0
- `@media all and (color) {...}`
- `@media all and (min-color: 1) {...}`
- color-index
- describes the number of entries in the color lookup table of the output device. If the device does not use a color lookup table, the value is zero.
- `@media all and (color-index) {...}`
- `@media all and (min-color-index: 256) {...}`
- monochrome
- describes the number of bits per pixel in a monochrome frame buffer. If the device is not a monochrome device, the output device value will be 0.
- `@media all and (monochrome) {...}`
- `@media all and (min-monochrome: 1) {...}`
- resolution
- `@media print and (min-resolution: 300dpi) {...}`
- scan
- `@media tv and (scan: progressive) {...}`
- grid

所以，什么都没有的时候，是写xs比较好的，因为后期再加东西，可以覆盖它。

在我们这里多数东西都引入了bootstrap，不必再自己关注media，比如面试的时候，要是问：解释一下media query

就是：col-lg-3 col-sm-6 col-xs-12

```
Angular:  
有关遍历的：  
angular.forEach(vm.data, function(data) {  
    console.log(vm.data);  
    var params = {  
  
        product_code:data.product_code  
  
        productId: data.sell_point  
    };  
    sellPointList.push(params);  
    return sellPointList;});
```

改变：
(将list用reduce的方式，将原先
product_code:111,product_name:2313213传值用111: 2313213方式传至后台)
sellPointList = vm.data.reduce(function(prev, next) {
 prev[next.product_code] = next.product_name;
 return prev;
}, {});
console.log(sellPointList);

```
reduce的另一种用法：  
vm.checkedNames = function () {  
    vm.checkedItems = vm.price_range.filter(function (it) {  
        return it.$checked;  
    });  
    if (vm.checkedItems.length > 0) {  
        var checkedString = vm.checkedItems.reduce(function (a, b) {  
            return a + b.name + ",";  
        }, "");  
        vm.checkedItemsStr = checkedString.substring(0, checkedString.length-1);  
    }  
};
```

将name属性拼成字符串传出：“name1、name2、name3”

遇到无法终止forEach循环时，用every替代：

forEach 遍历的范围在第一次调用 callback 前就会确定。调用forEach 后添加到数组中的项不会被 callback 访问到。如果已经存在的值被改变，则传递给 callback 的值是 forEach 遍历到他们那一刻的值。已删除的项不会被遍历到。

☒ 注意：没有办法中止forEach循环。如果要中止，可使用 Array.every 或 Array.some。见下面的例子+

forEach 为数组中的元素执行一次 callback 函数，不像 every 和 some，它总是返回 undefined。

```
angular里面引用setTimeout进行延时：  
angular.module("ng-admin").controller("OnDownloadDialogCtrl", ['$scope', "$timeout",  
function($scope,$timeout,AlertService, DialogService, service) {  
    var vm = $scope;  
    // 导入商品  
    vm.onBack = function() {  
        DialogService.hide("SaaS.Onsale.OnDownloadDialog");  
    };  
    vm.close = function() {  
        DialogService.dismiss("SaaS.Onsale.OnDownloadDialog");  
    };  
    vm.importGoods = function() {  
        console.log(vm.data);  
        $timeout(function() { vm.importList(); },10000);  
    };  
    vm.importList = function() {  
        var params = {};  
        var type = 1;  
        if(vm.data.autoImport != "1"){  
            type = vm.data.fixedOff;  
        }  
        params['type'] = type;  
        params = JSON.stringify(params);
```

```
跳转页面：EditInSaleGoods是配置的路由名，后面是携带的参数：  
a ui-sref="EditInSaleGoods({product_code: item.product_code,category_code:item.category_code});  
  
.state("EditInSaleGoods", {  
    url: "/EditInSaleGoods/{product_code}?category_code",  
    templateUrl: "modules/saas/templates/onSale/EditInSaleGoods.html"  
})  
  
* Created by 14081206 on 15-9-6.  
*/  
angular.module("ng-admin").controller("EditInSaleGoodsCtrl", ['$scope', '$filter', '$stateParams',  
function($scope, $filter, $stateParams, baseUrl, service, AlertService, DialogService, FileUp  
    var vm = $scope;  
    productCode = $stateParams.product_code,  
    //categoryCode = $stateParams.category_code,  
    EditInSaleGoods = baseUrl + 'index/editInSaleGoods';
```

论数组与对象的应用场景：
var packingList = [
{

```
map和list:  
List:  [{id:1,name:111}, {id:2, name:222}]  
  
Map:  {1:{id:1,name:111}, 2:{id:2,name:222}}
```

```
name: '商品1',  
qty: '10'  
}, {  
    name: '商品2',  
    qty: '20'  
}, {  
    name: '商品3',  
    qty: '30'  
}];  
  
var test = [];  
var pack = {};  
angular.forEach(packingList, function (data) {  
    pack={  
        nameVal:data.name,  
        amountVal:data.qty  
    }  
    test.push(pack);  
});
```


工具链接

2015年12月17日
19:38

有关模拟数据，可用deployd
<https://bintray.com/artifact/download/deployd/dpd-win-installer/deployd-win-0.8.0.exe>

静态http服务器：
<http://cncspace.newhua.com/down/EasyWebSvr1.9.zip>

```
nginx配置跨域
1、E:\nginx-1.9.9\conf\vhsts
配置：
server {
    listen 8180;
    server_name localhost;
    root "E:\snpc-frontwar\src\main\webapp\web"; //工程目录
    index index.html;

    location / {

        proxy_pass http://10.24.73.43/; //访问接口url
    }

    location ~ \.(js|css|html|jpeg|jpg|png|gif|ico|woff|woff2|ttf)$ {
        root "E:\snpc-frontwar\src\main\webapp\web";
    }

    access_log logs/access/user.log main; //日志记录
    error_log logs/access/user_err.log debug;
}
```

在目录：E:\nginx-1.9.9\conf
引入新增的跨域配置目录：
include vhosts/*.conf;

cd D:\Program Files\nginx-1.8.0> start .\nginx.exe 启动
cd D:\Program Files\nginx-1.8.0> .\nginx.exe -s stop 关闭
cd D:\Program Files\nginx-1.8.0> .\nginx.exe -s reload 重启

解决超时504报错：
在nginx.conf里，加入：
fastcgi_buffers 8 128k
这表示设置fastcgi缓冲区为8×128k
当然如果您在进行某一项即时的操作，可能需要nginx的超时参数调大点，例如设置成60秒：
send_timeout 60;
我只是调整了这两个参数，结果就是没有再显示那个超时，可以说效果不错
fastcgi_connect_timeout 300s;
fastcgi_buffers 8 128k;
send_timeout 60;

如果首页想要进来默认访问某个子项页面，可以将default指向的页面改为子页面，

```
$urlRouterProvider.otherwise('/console/basicInfo');
```

并且为了防止出现，删除其余后缀，又跳转到console.html页面，可以将其属性设置为如下：

```
$stateProvider
.state('Console', {
    url: '/console',
    templateUrl: 'modules/console/templates/console.html',
    controller: 'ConsoleCtrl',
    controllerAs: 'console',
    abstract: true
})
```

mongodb

2016年2月23日
16:41

1. 下载工具包
2. bin同级目录新建一个db目录



3. 定义数据库存放路径为此目录，启动mongodb

```
D:\mongodb\bin>mongod --dbpath="d:\mongodb\db"
```

有关github的运用

在遇到公司上网权限受限需要配置代理时：

在用户根目录下添加配置文件：.gitconfig
C:\Users\14081206\Documents\SuningImFiles\14081206

```
[gui]
  recentrepo = C:/Users/86057937/start
  recentrepo = C:/Users/86057937/learnig

[user]
  name = HTF
  email = 457296373@qq.com

[http]
  proxy = http://10.19.110.32:8080
  sslverify = false

[https]
  proxy = http://10.19.110.32:8080

[alias]
  last = log -l
  lg = log --color --graph --pretty=format:'%Cred%h%Creset -%C(yellow)%d%Creset %s %Cgreen(%cr) %C(bold blue)<an>%Creset' --abbrev-commit
[url "https://"]
  insteadOf = git://
[credential]
  helper = wincred
```

有关git代码提交的处理：

1. 初始化 git init

```
C:\Users\14081206\Desktop\helloworld>git init
Initialized empty Git repository in C:\Users\14081206\Desktop\helloworld\.git\
```

可以查看代码修改状态：

```
C:\Users\14081206\Desktop\helloworld>git status
On branch master
```

2. 开始本地仓库预提交

```
C:\Users\14081206\Desktop\helloworld>git add -a
```

3. 本地库代码提交

```
C:\Users\14081206\Desktop\helloworld>git commit -m "first commit"
[master (root-commit) ca97109] first commit
```

4. 提交远程网页托管（首次提交，需要输入库地址）

```
C:\Users\14081206\Desktop\helloworld>git remote add origin https://github.com/C
```

```
C:\Users\14081206\Desktop\helloworld>git push -u origin master
```

4. 第二次该库提交只需输入以下指令

```
C:\Users\14081206\Desktop\helloworld>git push origin master
```

补充：可以省去每次输入用户名和密码的方式：

```
git config --global credential.helper wincred
```

```
C:\Users\14081206\Desktop\helloworld>git config --global credential.helper wincred
```

参考链接：

<https://github.com/nuvsoft/Mock/wiki/Getting-Started>

node入门书籍

<http://nodebeginner.org/index-zh-cn.html>

Express中文手册：

<http://www.expressjs.com.cn/4x/api.html>

1. 安装node.js，并配置环境变量（新版本自带npm）

2. npm安装mockjs

```
npm install mockjs
```

3. npm安装express

```
npm install express
```

参考：<http://www.expressjs.com.cn/starter/installing.html>

4. helloWorld

参考：<http://www.expressjs.com.cn/starter/hello-world.html>

5. express的api

参考：<http://www.expressjs.com.cn/4x/api.html>

- 1.

因为express本身没提供html的渲染器

需要用ejs模板提供的

所以，先npm按照ejs

2. 需要获取前端依赖的包，就如npm管理后台的依赖包，前台需要bower这样的工具做类似的事

安装

```
npm install -g bower
```

```
bower install --save mockjs
```

Bower install angular 安装angular的包

需要下载bootstrap类似即可。可实现查找bower search bootstrap

源代码 <<https://github.com/nuvsoft/Mock/wiki/Getting-Started>>

需要注意的是，访问权限如果限制，需要设置代理

为 Bower 设置代理

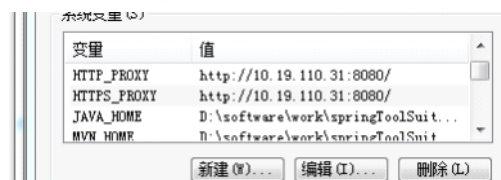
bower的安装和使用依赖于 npm 和 git，在完成前两者的代理设置后，有两种方法为 bower 设置代理。

方法一，直接修改 bower 的配置文件 .bowerrc，如果没有可先添加

```
1 "proxy": "http://10.24.48.6:8080",
2 "https-proxy": "http://10.24.48.6:8080"
```

方法二，在系统变量里添加 HTTP_PROXY 和 HTTPS_PROXY。

关于 bower 的使用可参考 此文。



可以直接这样设置即可。

安装好之后可以在bower_components这个目录下看到所下载下来的依赖的angular的包

- 3.

然后，需要一个总的启动文件，比如app.js

在这里面，你要定义 angular.module('myApp', [])

这样，index.html里面绑定的，ng-app="myApp"，才能找到对应的

然后，如果是多页面，肯定要有路由，毕竟angular是开发SAP应用，没有路由就没法跳转到相应页面

angular的路由有两种，一种是自带的，一种是ui.router

我们项目用的，都是后者

关于路由，你可以看下这篇

可以参考有关路由的链接：

<http://huangtengfei.com/2015/10/the-useage-of-ng-route-and-ui-router/>

资料

2016年2月29日
16:25

1、bfc

<http://www.cnblogs.com/pigtail/archive/2013/01/23/2871627.html>

<http://huangtengfei.com/2015/08/bfc-and-float/>

2、flex flexbox

3、em rem

4、重绘重排

5、http请求发送全过程

6、

4

2

1.ng-if和 ng-show

<http://huangtengfei.com/2015/09/scope-in-angularjs/>

有关路由方面的:

AngularJS 原生的 ngRoute 路由模块和第三方路由模块 ui.router 的用法进行简单对比

源文档 <<http://huangtengfei.com/2015/10/the-usage-of-ng-route-and-ui-router/>>

从服务提供者角度看:

ngRoute:

```
var app = angular.module('ngRouteApp', ['ngRoute']);

app.config(function($routeProvider){
    $routeProvider
        .when('/Main', {
            templateUrl: "main.html",
            controller: 'MainCtrl'
        })
        .otherwise({ redirectTo: '/tabs' });
});
```

ui.router:

```
1 var app = angular.module("uiRouteApp", ["ui.router"]);
2
3 app.config(function($urlRouterProvider, $stateProvider) {
4     $urlRouterProvider.otherwise("/index");
5     $stateProvider
6         .state("Main", {
7             url: "/main",
8             templateUrl: "main.html",
9             controller: 'MainCtrl'
10        })
11    });
```

可以看出:

\$routeProvider 服务提供者, 用来定义一个路由表, 即地址栏与视图模板的映射, 对应于 ui.router 中的 urlRouterProvider 和 stateProvider

因为ui.router是基于state的, 正常情况下, 每个view都对应一个state, 如果找不到这个view对应的state, 那么就可以通过urlRouterProvider的otherwise给出一个默认的url。

ui-router (左) : ngRoute (右)

- 应用程序内的一个区域 : 应用程序中的 url
- 可以嵌套的层次结构 : 只是平面层次结构
- 名称可以自定义 : 名称只能是 url
- 通过名称或 url 导航 : 只能通过 url 导航
- 可以存在多个视图 (ui-view) : 只能单一视图 (ng-view)
- 可以填充任何视图 : 只能填充一个视图
- 通过状态填充某一部件 : 通过指令将填充某一部件

可参考链接: <http://www.html-js.com/article/Front-end-source-code-analysis-original-ui-router-source-code-analysis>