[在github上写个人简历——最简单却又不容易的内容罗列](http://www.cnblogs.com/dolphinX/p/3439376.html)

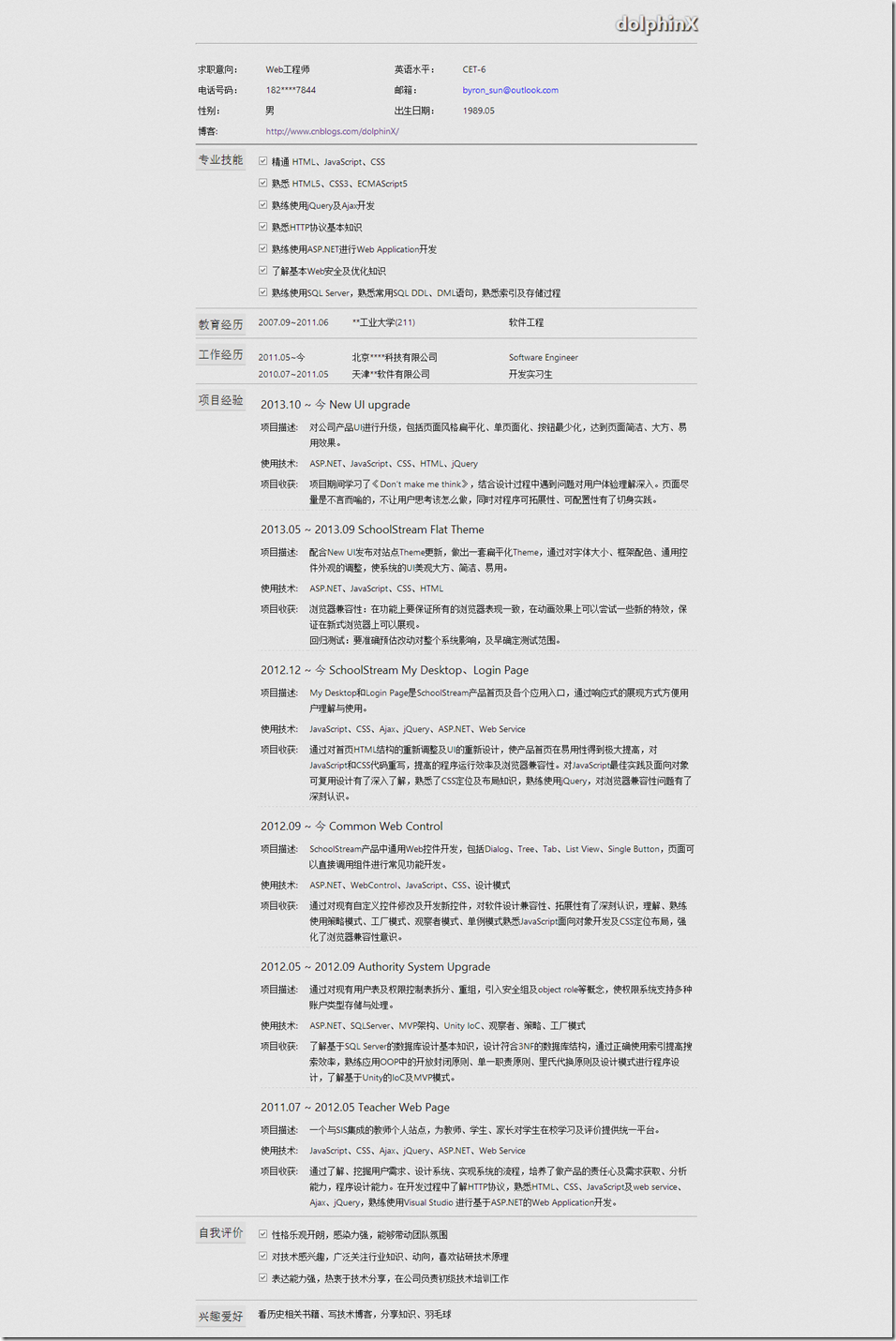
前篇博客分享了一下自己初入github的过程，傻瓜式一步步搭建好主页后，终于该做正事儿了——写简历。在脑袋中构思了很多版本，最后终于决定，先写一个最传统的版本，于是我在箱子中翻出我word版本的简历，对照写一个html版本的。

**word版本看起来是这个样子的**

[](http://images.cnitblog.com/blog/349217/201311/25164314-498ee8913b364b5486b4b38ca4ee0f8d.png)

**html版看起来是这样子的**

本来以为很简单的事情，没想到折腾好久，反反复复修改，终于弄出一个满意的版本,有兴趣的同学可以到[我的github](http://dolphinx.github.io/" \t "_blank)上看看

[](http://images.cnitblog.com/blog/349217/201311/25164317-6bdeecdd5d08414a85ee31a827774197.png)

**技术点儿分析**

源码

http://images.cnblogs.com/OutliningIndicators/ContractedBlock.gifView Code

看起来平淡无奇，实际确实没什么高深的，都是最简单的html与css。。。不过最开始我图快全都用简单的table布局实现，后来在不断修改中，也用了一些心思，学到了一些新知识。为了用一些html5和CSS3的东西，只好舍弃了浏览器兼容性，目前我只在最新版本Chrome和Firefox上测试过，而且由于家里没有Windows环境，Mac上滚动条表现形式和Windows上不太一样，所以。。。。

**页面结构**

上面提到最开始我除了主框架全部使用table布局，后来改为html5版本，舍弃了大部分table(并不是不用table了，table在数据展示方面还是很有优势的)，页面布局实际上是这样的了

[复制代码](javascript:void(0);)

1 <!DOCTYPE html>

2 <html>

3 <head></head>

4 <body>

5 <header></header>

6 <article>

7 <section></section>

8 <section></section>

9 <section></section>

10 <section></section>

11 <aside></aside>

12 </article>

13 </body>

14 </html>

[复制代码](javascript:void(0);)

**reset**

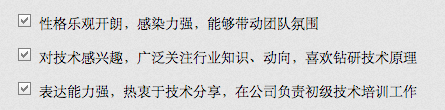
细心的朋友可以注意到页面引用了一个reset.css，这是根据[雅虎的reset.css](http://yuilibrary.com/yui/docs/cssreset/" \t "_blank)简单改动的一个版本，自己以前盲目的写了很多，不但凌乱不堪，而且大部分无用，会导致浏览器repaint，降低页面渲染效率

http://images.cnblogs.com/OutliningIndicators/ContractedBlock.gifView Code

**关于使用section、aside等标签不是装逼的辩解**

其实最开始我也很看不惯html5的section、nav、aside等标签，明明和div一样，觉得使用这个纯属装逼，后来接触到一些关于html结构化和语义化的知识后才认识到自己的浅薄，使用了这些标签后能够使计算机更容易理解网页内容，这些标签都是有语义的，比如nav标签里放的是页面导航，footer放的是页面的copyright等，搜索引擎等机器可以更针对性的查找、理解其需要的内容，关于这部分知识有兴趣可以看看[web语义化](http://www.cnblogs.com/dolphinX/p/3311588.html)

**一直被忽略的::before/::after伪元素**



每条前面的对勾以前我会直接逐个写到html中，看了一些网站源码才发现::berore/::after的妙处

li::before{

content:url(./images/icons/yes.png);

padding-right:6px;

vertical-align: middle;

}

简简单单这样就搞定了，content可以制定内容，文字、图片都可以

**关于::与:也就是伪元素与伪类区别**

简单讲伪元素是为了标示某些元素如first-line等，而伪类是为了区别一个元素的不同类别如:hover,:visited,越说越不明白，看看[这个](http://www.jb51.net/css/67317.html" \t "_blank)吧

**使用disply:box, box-flex 属性布局**

****

看到这种两栏儿布局，大家第一反应是什么，我反正上来想到的是table，然后是float，然后是display:inline-block,这几种布局的局限性很明显，不再赘述，看看CSS3中的解决方案

<section style="display:box">

<section style="box-flex:2;">lefft</section>

<section style="box-flex:3;">right</section>

</section>

display:box属性赋予了我们划分容器的能力，父容器有了display:box（真正使用需要添加浏览器前缀）属性后，其子元素div或section就不再向标准block元素那样独占一行了，几个div可以共处一行，每个div占多少呢？我们可以使用width属性为期赋予绝对宽度，也可以使用box-flex（使用需要浏览器前缀）属性按比例划分，上面例子中left section占父容器宽度的2/(2+3)也就是2/5，右边是3/5。关于相关属性box-ordinal-group, box-orient, box-pack, box-align有兴趣同学可以去网上查查资料

**标题怎么凸出来**

[image](http://images.cnitblog.com/blog/349217/201311/25164826-0d48043e42154909bcc931c6bf08cecd.png)

这个简单的效果让我折腾了一个小时，悲剧的是text－shadow我很早就使用过，用法可以看看[CSS3的文字阴影—text-shadow](http://www.w3cplus.com/blog/52.html" \t "_blank)，我不明白的是怎么弄就能让文字凸出来或者凹下去，每次就是瞎试，今天小研究了一下貌似明白了什么。

首先是文字颜色与背景颜色的统一，两个颜色类似才能方便做出这种效果，然后是阴影的颜色用反色，白色文字就用黑色阴影，至于凹凸就要看位移的正负了。我的这个的css是这样的

[复制代码](javascript:void(0);)

color:#e8e8e8; /\*页面背景也是这个颜色\*/

text-shadow: 2px 2px 2px #333, 0px 0px 5px #666;/\*移位两次，正数就会凸出来，阴影颜色接近反色\*/

font-size: 30px;

font-weight:bold;

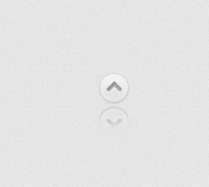
text-align:right;

margin-top:15px;

[复制代码](javascript:void(0);)

**回到顶部图标**

由于使用截屏工具，那个图标不明显，其实页面滚动条不再顶部的时候，页面上有个现在网站很常见的回到顶部图标的



难住我的倒不是这个图标的显示控制，这个很简单，把元素设position设为fixed，然后根据页面滚动条位置决定是否出现，Mac下滚动条是可以为负数的，这块儿得处理一下。

[复制代码](javascript:void(0);)

1 <aside id="wrap" style="position:fixed; bottom:100px; right:100px;display:none; opacity:0.5;">

2 <section style="background-image:url(images/icons/top.png); width:32px;height:32px;"></section>

3 <section class="reflect"></section>

4 <section class="shadow"></section>

5 </aside>

[复制代码](javascript:void(0);)

[复制代码](javascript:void(0);)

1 $(function(){

2 $('#wrap').on('click', function(event) {

3 var obj=document.body.scrollTop>0? document.body:document.documentElement;

4 $(obj).animate({"scrollTop":0}, 1000);

5 $(this).animate({"opacity":0.5}, 1000);

6 }).on('mouseover', function(event) {

7 $(this).css('opacity',1);

8 }).on('mouseout',function(event) {

9 $(this).css('opacity',0.5);

10 });

11 });

12 $(document).scroll(function(event) {

13 var goTop=$('#wrap');

14 var scrollTop=document.body.scrollTop || document.documentElement.scrollTop || 0;

15 if(scrollTop>0){

16 if(goTop.css('display')=='none'){

17 $('#wrap').fadeIn(500);

18 }

19 }else{

20 if(goTop.css('display')!='none'){

21 $('#wrap').fadeOut(500);

22 }

23 }

24 });

[复制代码](javascript:void(0);)

难住我的是倒影部分，关于这个倒影纯属那个啥了，本来处理很简单，其实就是我源代码中保留的部分，页面放一个aside加上style完事儿

[复制代码](javascript:void(0);)

1 #goTop{

2 position:fixed;

3 right:100px;

4 bottom:100px;

5 background-image:url(images/icons/top.png);

6 width:32px;

7 height:32px;

8 -webkit-box-reflect: below 0px -webkit-gradient(linear, center top, center bottom, from(transparent),color-stop(0.2, transparent), to(white));

9 opacity: 0.5;

10 }

[复制代码](javascript:void(0);)

效果和上面截图类似，悲剧的是只有Chrome支持，最起码的节操要求我至少把Firefox搞定，反复尝试终于找到了解决办法，具体说来是这样的

[复制代码](javascript:void(0);)

1 <aside id="wrap" style="position:fixed; bottom:100px; right:100px;display:none; opacity:0.5;">

2 <section style="background-image:url(images/icons/top.png); width:32px;height:32px;"></section>

3 <section class="reflect"></section>

4 <section class="shadow"></section>

5 </aside>

[复制代码](javascript:void(0);)

id为wrap的aside就是整个回到顶部部分，第一个section是上面那个没有倒影的图片，class为reflect的section是倒影图片

[复制代码](javascript:void(0);)

1 .reflect{

2 background-image:url(images/icons/top.png);

3 width:32px;

4 height:32px;

5 -webkit-transform: scaleY(-1);

6 -moz-transform: scaleY(-1);

7 -ms-transform: scaleY(-1);

8 transform: scaleY(-1);

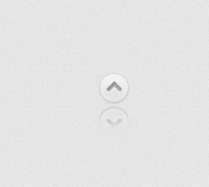
9 filter:alpha(opacity='80');

10 opacity: 0.8;

11 }

[复制代码](javascript:void(0);)

其实就是插入同样图片，然后使 transform: scaleY(-1); 倒置图片，加上一定的透明，可是这还没完，这样的效果是这样的

 对比一下预期效果

倒是有倒影了，但是倒影没有消失最后，好像有个向下的按钮似的。。。所以加入了class为shadow的section，就是为了让倒影逐渐消失

[复制代码](javascript:void(0);)

1 .shadow{

2 position: relative;

3 top:-32px;

4 left:0px;

5 height:32px;

6 width:32px;

7 background-image: -moz-linear-gradient(center bottom, rgb(227,227,227) 30%, rgba(255,255,255,0) 100%);

8 background-image: -webkit-gradient(linear, center bottom, center top, color-stop(0.3, rgb(227,227,227)), color-stop(0.7, rgba(255,255,255,0)));

9 filter: progid:DXImageTransform.Microsoft.Gradient(gradientType=0,startColor=#e8e8e8, EndColorStr=#ffffff);

10 }

[复制代码](javascript:void(0);)

把section上移32px使其处于reflect正上方，然后对这层做一个透明渐变，使reflect处于相应位置的图像部分也透明渐变，终于达到预期效果。关于-moz-linear-gradient/-webkit-gradient的使用可以看看理解[CSS3线性渐变](http://www.qianduan.net/understand-the-css3-gradient.html)

**最后**

本来就是本着玩玩儿的目的写这个在线简历的，没想到收获比我预期的大得多，熟悉了很多自己一知半解的东西，而且制作过程中产生了很多新想法，建议有想法的同学们也试试，即使不想换工作，制作过程中会有想不到的收获。

接下来会陆续做一些类似的，不同样式的简历，可能要反复修改，应该进度会比较慢，主要是锻炼一下JavaScript, html5, CSS3技巧，可以在[我的github](https://github.com/DolphinX/dolphinx.github.com)上看看我的进度，尽量尽快push 更新。

PS.虽然现在不急着换工作，但朋友看了我的简历内容后都觉得写的弱爆了，都是空谈，看不出具体会什么，希望热心的朋友也能给出这方面的修改意见