需求：

①写一个web版的360工具箱，示意图如下：



②无左上返回按钮，右上按钮有皮肤切换，下拉框（但无点击逻辑）；

③按钮点击有事件，但事件是console.log(按钮名)；

④可以在全部工具和我等工具自由切换；

⑤可以点击左下角的编辑，然后根据实际表现设置；

⑥可以在全部工具里面，点击按钮，然后添加到我的工具这边来；

⑦效果尽量与原图相同。

效果网址：

<http://jianwangsan.cn/toolbox>

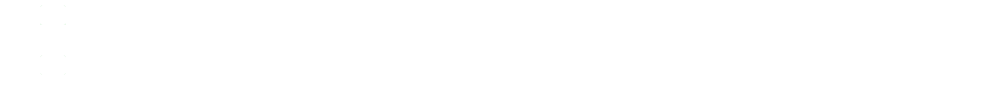
（一）tab页切换

①切图：

先切图，如图：（不想用他的绿色的）



再切按钮图（自行ps）：（下图白色，所以直接是看不见的）



②页面制作：

html：我这里发的是jade版，如果需要看html版的话，请打开demo的网址：<http://jianwangsan.cn/toolbox>，然后查看源代码自行复制

**link**(rel='stylesheet', href='./stylesheets/toolboxes.css')  
**script**(type="text/javascript",src='javascripts/toolboxes.js')  
**div**.*back* **div**.*tooltop* **div**.*tooltop-img* **div**.*toolTab* **div**.*tool*#alltool  
 **span**.*img*.*alltool*.*select* **span**.*text* **全部工具  
 div**.*hover* **div**.*tool*#mytool  
 **span**.*img*.*mytool* **span**.*text* **我的工具**

css：

a[href='/toolbox'] {  
 color: #555;  
 text-decoration: none;  
 background-color: #e5e5e5;  
 -webkit-box-shadow: inset 0 3px 8px rgba(0, 0, 0, 0.125);  
 -moz-box-shadow: inset 0 3px 8px rgba(0, 0, 0, 0.125);  
 box-shadow: inset 0 3px 8px rgba(0, 0, 0, 0.125);  
}  
  
.back {  
 height: 600px;  
 width: 100%;  
 position: relative;  
}  
  
.back \* {  
 border: 0;  
 padding: 0;  
 margin: 0;  
}  
  
.back .tooltop {  
 height: 120px;  
 width: 100%;  
 background-color: white;  
 position: relative;  
}  
  
.back .tooltop-img {  
 height: 100%;  
 width: 100%;  
 background-image: url(../img/toolboxBackground.png);  
 background-size: cover;  
}  
  
.back .toolTab {  
 position: absolute;  
 left: 0;  
 bottom: 10px;  
 height: 35px;  
 width: 100%;  
}  
  
.back .toolTab .tool {  
 margin-left: 20px;  
 width: 140px;  
 height: 100%;  
 line-height: 30px;  
 color: white;  
 font-size: 22px;  
 font-weight: 900;  
 -webkit-box-sizing: border-box;  
 -moz-box-sizing: border-box;  
 box-sizing: border-box;  
 display: inline-block;  
 cursor: default;  
 position: relative;  
 -webkit-user-select: none;  
 -moz-user-select: none;  
 -ms-user-select: none;  
 user-select: none;  
}  
  
.back .toolTab .tool .img {  
 height: 27px;  
 width: 27px;  
 background-repeat: no-repeat;  
 display: inline-block;  
 vertical-align: middle;  
 background-image: url(../img/toolbox.png);  
}  
  
.back .toolTab .tool .img.alltool {  
 background-position: 0 0;  
}  
  
.back .toolTab .tool .img.alltool.select {  
 background-position: 0 -50px;  
}  
  
.back .toolTab .tool .img.mytool {  
 background-position: -40px 0;  
}  
  
.back .toolTab .tool .img.mytool.select {  
 background-position: -40px -50px;  
}  
  
.back .toolTab .tool .text {  
}  
  
.back .toolTab .hover {  
 height: 2px;  
 width: 125px;  
 background-color: white;  
 position: absolute;  
 bottom: -2px;  
 left: 0;  
}

js逻辑：

**$**(**document**).ready(function () {  
 //这里是点击切换显示页面  
 **$**(".tool").click(function () {  
 //这里是上面的图标的逻辑变换  
 if (!(**$**(this.children[0]).hasClass("select"))) {  
 **$**(".select").removeClass("select");  
 **$**(this.children[0]).addClass("select");  
  
 //这里是hover的横线的位置变化  
 var node = **$**(".tool .hover");  
 node.remove();  
 //当动画需要停止的时候，让他停止  
 if ('stop' in node) {  
 node.stop();  
 }  
 node.**css**("left", "0px");  
 **$**(this).append(node);  
  
 //以下应该是切换页面的逻辑  
  
 }  
 })  
  
 **$**(".tool").mouseenter(function (evt) {  
 //只有当鼠标移动到非当前选中的tab上时，才会移动  
 if (!(**$**(this.children[0]).hasClass("select"))) {  
  
 var self = this;  
 var node = **$**(".tool .hover");  
 var start = null;  
 var end = null;  
  
 var tools = **$**(".toolTab")[0].children  
 for (var i = 0; i < tools.**length**; i++) {  
 if (self == tools[i]) {  
 end = i;  
 } else if (**$**(".select")[0].parentNode == tools[i]) {  
 start = i;  
 }  
 }  
  
 //停止之前的动画  
 if ('stop' in node) {  
 node.stop();  
 }  
 //现在开始动画效果  
 node.animate({**"left"**: (end - start) \* 160 + "px"})  
 }  
 })  
 **$**(".tool").mouseleave(function () {  
 //只有当鼠标移动到非当前选中的tab上时，才会移动  
 if (!(**$**(this.children[0]).hasClass("select"))) {  
 var node = **$**(".tool .hover");  
 //停止之前的动画  
 if ('stop' in node) {  
 node.stop();  
 }  
 node.animate({**"left"**: "0px"})  
 }  
 })  
})

到目前为止，tab按钮的动画和切换可以了（缺少和页面的配合，但是那个很简单）。

（二）全部工具里面的按钮和样式

我将他拆成五部分：



第一部分是上面的大按钮那一排；

第二部分是小按钮；

第三部分是一条颜色很淡的线，他只在app有多行的情况下，在行间有；

第四部分是标题；

第五部分是右边的滚动条；

①为了方便模块化开发，我先制作模板，然后实际编写代码的时候，根据样式已经设置好的模板，来添加内容，这样方便调试和制作

②大按钮：

先分为整体和具体每个按钮；

每个按钮分为背景图、遮罩层（下半部分的灰色阴影），文字title，文字说明，还有移动上去可能显示的添加按钮，共计五部分。

html：

**div**.*contentbox  
 //下面这个是全部工具里页面最顶级的，目前共有两个，通过displayNONE这个类是否存在，来决定显示哪个* **div**.*toolbox-all  
 //这个是最上面的大图标那一行* **div**.*firstRow  
 //以下是单个按钮* **div**.*BigTool  
 //背景图* **span**.*img*(style='background-image: url(../img/bigImg01.png)')  
 *//阴影遮罩* **span**.*mask  
 //文字* **div**.*text* **div**.*title* **微信清理  
 div**.*description* **定期清理微信，节省手机空间** *//添加按钮* **div**.*addButton* **添　加  
 div**.*BigTool* **span**.*img*(style='background-image: url(../img/bigImg02.png)')  
 **span**.*mask* **div**.*text* **div**.*title* **雷电OS  
 div**.*description* **雷电OS Editor 旧机变新机  
 div**.*addButton* **添　加  
 div**.*BigTool* **span**.*img*(style='background-image: url(../img/bigImg03.png)')  
 **span**.*mask* **div**.*text* **div**.*title* **手机相册扩容  
 div**.*description* **无损处理照片，腾出50%空间  
 div**.*addButton* **添　加**

css：

.back .contentbox > div {  
 width: 100%;  
 height: 100%;  
 box-sizing: border-box;  
}  
  
.back .contentbox .toolbox-all {  
 background: white;  
 padding: 30px;  
 overflow: auto;  
}  
  
.back .contentbox .toolbox-my {  
 background: green;  
}  
  
.back .contentbox .toolbox-all .firstRow {  
 width: 100%;  
 height: 140px;  
 display: flex;  
 justify-content: space-between;  
}  
  
.back .contentbox .toolbox-all .firstRow .BigTool {  
 width: 300px;  
 height: 140px;  
 position: relative;  
 cursor: pointer;  
}  
  
.back .contentbox .toolbox-all .firstRow .BigTool .img {  
 display: inline-block;  
 position: absolute;  
 width: 100%;  
 height: 100%;  
}  
  
.back .contentbox .toolbox-all .firstRow .BigTool .mask {  
 display: inline-block;  
 position: absolute;  
 width: 100%;  
 height: 100%;  
 background-image: linear-gradient(rgba(0, 0, 0, 0) 0%, rgba(0, 0, 0, 0) 40%, rgba(0, 0, 0, 0.5) 60%, rgba(0, 0, 0, 0.8) 100%);  
}  
  
.back .contentbox .toolbox-all .firstRow .BigTool .text {  
 position: absolute;  
 bottom: 0;  
 left: 0;  
 right: 0;  
 height: 55px;  
 padding: 0 10px;  
}  
  
.back .contentbox .toolbox-all .firstRow .BigTool .text .title {  
 font-weight: 600;  
 font-size: 18px;  
 color: white;  
}  
  
.back .contentbox .toolbox-all .firstRow .BigTool .text .description {  
 font-size: 14px;  
 margin-top: 10px;  
 color: rgb(218, 218, 218);  
}  
  
.back .contentbox .toolbox-all .firstRow .BigTool .addButton {  
 display: none;  
 position: absolute;  
 bottom: 10px;  
 right: 12px;  
 width: 60px;  
 height: 22px;  
 background-image: linear-gradient(rgb(98, 227, 25) 0%, rgb(68, 208, 27) 100%);  
 font-size: 12px;  
 color: white;  
 text-align: center;  
 line-height: 20px;  
 border: 1px solid rgb(65, 199, 36);  
 -webkit-border-radius: 1px;  
 -moz-border-radius: 1px;  
 border-radius: 1px;  
}  
  
.back .contentbox .toolbox-all .firstRow .BigTool:hover .addButton {  
 display: block;  
}

③小按钮：

分为左边的图片，右边的文字（title和description），添加按钮

所有小按钮放在同一个commonRow类里面，这个commonRow是弹性布局。

commonRow和上面的firstRow是平级的

HTML：

**div**.*commanRow* **div**.*normalTool* **div**.*img* **div**.*text* **div**.*title* **手游模拟器  
 div**.*description* **电脑玩手游，挂机辅助神器  
 div**.*addButton* **添　加  
 div**.*normalTool* **div**.*img* **div**.*text* **div**.*title* **手游模拟器  
 div**.*description* **电脑玩手游，挂机辅助神器  
 div**.*addButton* **添　加  
 div**.*normalTool* **div**.*img* **div**.*text* **div**.*title* **手游模拟器  
 div**.*description* **电脑玩手游，挂机辅助神器  
 div**.*addButton* **添　加  
 div**.*normalTool* **div**.*img* **div**.*text* **div**.*title* **手游模拟器  
 div**.*description* **电脑玩手游，挂机辅助神器  
 div**.*addButton* **添　加**

CSS：

.back .contentbox .commanRow {  
 width: 100%;  
 display: flex;  
 justify-content: space-between;  
 margin-top: 10px;  
 flex-flow: wrap;  
}  
  
.back .contentbox .commanRow .normalTool {  
 width: 300px;  
 height: 70px;  
 position: relative;  
 cursor: pointer;  
 padding: 5px;  
 -webkit-box-sizing: border-box;  
 -moz-box-sizing: border-box;  
 box-sizing: border-box;  
}  
  
.back .contentbox .commanRow .normalTool:hover {  
 outline: 1px solid #dadada;  
}  
  
.back .contentbox .commanRow .normalTool .img {  
 position: relative;  
 display: inline-block;  
 width: 60px;  
 height: 60px;  
 background-color: blue;  
}  
  
.back .contentbox .commanRow .normalTool .text {  
 position: absolute;  
 left: 75px;  
 right: 5px;  
 top: 5px;  
 bottom: 5px;  
}  
  
.back .contentbox .commanRow .normalTool .text .title {  
 line-height: 35px;  
 font-size: 16px;  
}  
  
.back .contentbox .commanRow .normalTool .text .description {  
 line-height: 25px;  
 font-size: 12px;  
 color: #aaa;  
}  
  
.back .contentbox .commanRow .normalTool .addButton {  
 display: none;  
 position: absolute;  
 top: 10px;  
 right: 15px;  
 width: 60px;  
 height: 22px;  
 background-image: linear-gradient(rgb(98, 227, 25) 0%, rgb(68, 208, 27) 100%);  
 font-size: 12px;  
 color: white;  
 text-align: center;  
 line-height: 20px;  
 border: 1px solid rgb(65, 199, 36);  
 -webkit-border-radius: 1px;  
 -moz-border-radius: 1px;  
 border-radius: 1px;  
}  
  
.back .contentbox .commanRow .normalTool:hover .addButton {  
 display: block;  
}

④一条颜色很淡的线，分为两种情况，顶层的，和间层的。

直接插入一个div，利用

border: 1px dotted rgb(209, 209, 209);

属性来设置，

html：

**div**.*dotted*

在顶层的时候，和firstRow平级，在间层的时候，和.normalTool平级

css：

.back .contentbox .dotted {  
 border: 1px dotted rgb(209, 209, 209);  
 margin-top: 10px;  
 width: 100%;  
 margin-bottom: 10px;  
}

⑤标题：

两部分，左边的绿色竖线，右边的文字，简单暴力

html

**div**.*titleRow* **span**.*titleRow-left* **span**.*titleRow-text* **电脑安全**

CSS：

.back .contentbox .titleRow {  
 width: 100%;  
 height: 20px;  
 margin-top: 25px;  
}  
  
.back .contentbox .titleRow .titleRow-left {  
 display: inline-block;  
 width: 2px;  
 height: 100%;  
 background-color: rgb(42, 191, 29);  
}  
  
.back .contentbox .titleRow .titleRow-text {  
 height: 100%;  
 display: inline-block;  
 line-height: 20px;  
 margin-left: 10px;  
 vertical-align: top;  
 font-size: 18px;  
 color: #000;  
}

⑥滚动条

分为四部分：滚动条整体、滚动条背景、滚动条、两端的按钮

不需要html代码；

CSS代码限制为当前页面生效：

/\* 有这行才有效，滚动条的宽度 \*/  
.back .contentbox ::-webkit-scrollbar {  
 width: 12px;  
}  
  
/\* 滚动条的背景 \*/  
.back .contentbox ::-webkit-scrollbar-track {  
 background-color: rgb(242, 242, 242);  
}  
  
/\*滚动条\*/  
.back .contentbox ::-webkit-scrollbar-thumb {  
 -webkit-border-radius: 5px;  
 border-radius: 5px;  
 background: rgb(218, 218, 218);  
}  
  
.back .contentbox ::-webkit-scrollbar-button {  
 width: 12px;  
}

总结：

目前情况是小按钮没有背景图，暂时不做，只起到占位效果，背景图等做按钮类时，再根据实际需要设置和添加。

后附目前为止的代码：

HTML：（jade格式）

extends layout  
**block content  
 link**(rel='stylesheet', href='./stylesheets/toolboxes.css')  
 **script**(type="text/javascript",src='javascripts/toolboxes.js')  
 **div 博客地址：  
 a**(href='http://blog.csdn.net/qq20004604?viewmode=list' target='\_blank') **http://blog.csdn.net/qq20004604?viewmode=list  
 br  
 br  
 div**.*back* **div**.*tooltop* **div**.*tooltop-img* **div**.*toolTab* **div**.*tool*#alltool  
 **span**.*img*.*alltool*.*select* **span**.*text* **全部工具  
 div**.*hover* **div**.*tool*#mytool  
 **span**.*img*.*mytool* **span**.*text* **我的工具  
 div**.*contentbox  
 //下面这个是全部工具里页面最顶级的，目前共有两个，通过displayNONE这个类是否存在，来决定显示哪个* **div**.*toolbox-all  
 //这个是最上面的大图标那一行* **div**.*firstRow  
 //以下是单个按钮* **div**.*BigTool  
 //背景图* **span**.*img*(style='background-image: url(../img/bigImg01.png)')  
 *//阴影遮罩* **span**.*mask  
 //文字* **div**.*text* **div**.*title* **微信清理  
 div**.*description* **定期清理微信，节省手机空间** *//添加按钮* **div**.*addButton* **添　加  
 div**.*BigTool* **span**.*img*(style='background-image: url(../img/bigImg02.png)')  
 **span**.*mask* **div**.*text* **div**.*title* **雷电OS  
 div**.*description* **雷电OS Editor 旧机变新机  
 div**.*addButton* **添　加  
 div**.*BigTool* **span**.*img*(style='background-image: url(../img/bigImg03.png)')  
 **span**.*mask* **div**.*text* **div**.*title* **手机相册扩容  
 div**.*description* **无损处理照片，腾出50%空间  
 div**.*addButton* **添　加** *//横线那一行，如果是多行app，应考虑使用另外一个* **div**.*dotted* **div**.*commanRow* **div**.*normalTool* **div**.*img* **div**.*text* **div**.*title* **手游模拟器  
 div**.*description* **电脑玩手游，挂机辅助神器  
 div**.*addButton* **添　加  
 div**.*normalTool* **div**.*img* **div**.*text* **div**.*title* **手游模拟器  
 div**.*description* **电脑玩手游，挂机辅助神器  
 div**.*addButton* **添　加  
 div**.*normalTool* **div**.*img* **div**.*text* **div**.*title* **手游模拟器  
 div**.*description* **电脑玩手游，挂机辅助神器  
 div**.*addButton* **添　加  
 div**.*dotted* **div**.*normalTool* **div**.*img* **div**.*text* **div**.*title* **手游模拟器  
 div**.*description* **电脑玩手游，挂机辅助神器  
 div**.*addButton* **添　加  
 div**.*titleRow* **span**.*titleRow-left* **span**.*titleRow-text* **电脑安全  
 div**.*commanRow* **div**.*normalTool* **div**.*img* **div**.*text* **div**.*title* **手游模拟器  
 div**.*description* **电脑玩手游，挂机辅助神器  
 div**.*addButton* **添　加  
 div**.*toolbox-my*.*displayNONE*

CSS代码：

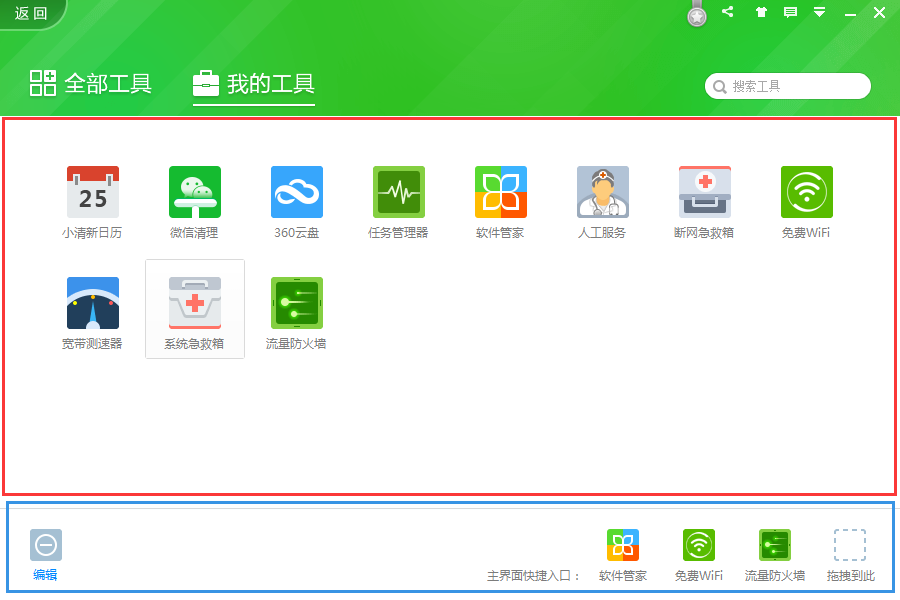
a[href='/toolbox'] {  
 color: #555;  
 text-decoration: none;  
 background-color: #e5e5e5;  
 -webkit-box-shadow: inset 0 3px 8px rgba(0, 0, 0, 0.125);  
 -moz-box-shadow: inset 0 3px 8px rgba(0, 0, 0, 0.125);  
 box-shadow: inset 0 3px 8px rgba(0, 0, 0, 0.125);  
}  
  
.back {  
 height: 600px;  
 width: 100%;  
 position: relative;  
 -webkit-box-shadow: 0px 0px 2px #555;  
 -moz-box-shadow: 0px 0px 2px #555;  
 box-shadow: 0px 0px 2px #555;  
 min-width: 1000px;  
}  
  
.back \* {  
 border: 0;  
 padding: 0;  
 margin: 0;  
}  
  
.back .tooltop {  
 height: 120px;  
 width: 100%;  
 background-color: white;  
 position: relative;  
}  
  
.back .tooltop-img {  
 height: 100%;  
 width: 100%;  
 background-image: url(../img/toolboxBackground.png);  
 background-size: cover;  
}  
  
.back .toolTab {  
 position: absolute;  
 left: 0;  
 bottom: 10px;  
 height: 35px;  
 width: 100%;  
}  
  
.back .toolTab .tool {  
 margin-left: 20px;  
 width: 140px;  
 height: 100%;  
 line-height: 30px;  
 color: white;  
 font-size: 22px;  
 font-weight: 900;  
 -webkit-box-sizing: border-box;  
 -moz-box-sizing: border-box;  
 box-sizing: border-box;  
 display: inline-block;  
 cursor: default;  
 position: relative;  
 -webkit-user-select: none;  
 -moz-user-select: none;  
 -ms-user-select: none;  
 user-select: none;  
}  
  
.back .toolTab .tool .img {  
 height: 27px;  
 width: 27px;  
 background-repeat: no-repeat;  
 display: inline-block;  
 vertical-align: middle;  
 background-image: url(../img/toolbox.png);  
}  
  
.back .toolTab .tool .img.alltool {  
 background-position: 0 0;  
}  
  
.back .toolTab .tool .img.alltool.select {  
 background-position: 0 -50px;  
}  
  
.back .toolTab .tool .img.mytool {  
 background-position: -40px 0;  
}  
  
.back .toolTab .tool .img.mytool.select {  
 background-position: -40px -50px;  
}  
  
.back .toolTab .tool .text {  
}  
  
.back .toolTab .hover {  
 height: 2px;  
 width: 125px;  
 background-color: white;  
 position: absolute;  
 bottom: -2px;  
 left: 0;  
}  
  
.back .contentbox {  
 position: absolute;  
 top: 120px;  
 bottom: 0;  
 left: 0;  
 right: 0;  
 background-color: white;  
}  
  
.back .contentbox > div {  
 width: 100%;  
 height: 100%;  
 box-sizing: border-box;  
}  
  
.back .contentbox .toolbox-all {  
 background: white;  
 padding: 30px;  
 overflow: auto;  
}  
  
.back .contentbox .toolbox-my {  
 background: green;  
}  
  
.back .contentbox .toolbox-all .firstRow {  
 width: 100%;  
 height: 140px;  
 display: flex;  
 justify-content: space-between;  
}  
  
.back .contentbox .toolbox-all .firstRow .BigTool {  
 width: 300px;  
 height: 140px;  
 position: relative;  
 cursor: pointer;  
}  
  
.back .contentbox .toolbox-all .firstRow .BigTool .img {  
 display: inline-block;  
 position: absolute;  
 width: 100%;  
 height: 100%;  
}  
  
.back .contentbox .toolbox-all .firstRow .BigTool .mask {  
 display: inline-block;  
 position: absolute;  
 width: 100%;  
 height: 100%;  
 background-image: linear-gradient(rgba(0, 0, 0, 0) 0%, rgba(0, 0, 0, 0) 40%, rgba(0, 0, 0, 0.5) 60%, rgba(0, 0, 0, 0.8) 100%);  
}  
  
.back .contentbox .toolbox-all .firstRow .BigTool .text {  
 position: absolute;  
 bottom: 0;  
 left: 0;  
 right: 0;  
 height: 55px;  
 padding: 0 10px;  
}  
  
.back .contentbox .toolbox-all .firstRow .BigTool .text .title {  
 font-weight: 600;  
 font-size: 18px;  
 color: white;  
}  
  
.back .contentbox .toolbox-all .firstRow .BigTool .text .description {  
 font-size: 14px;  
 margin-top: 10px;  
 color: rgb(218, 218, 218);  
}  
  
.back .contentbox .toolbox-all .firstRow .BigTool .addButton {  
 display: none;  
 position: absolute;  
 bottom: 10px;  
 right: 12px;  
 width: 60px;  
 height: 22px;  
 background-image: linear-gradient(rgb(98, 227, 25) 0%, rgb(68, 208, 27) 100%);  
 font-size: 12px;  
 color: white;  
 text-align: center;  
 line-height: 20px;  
 border: 1px solid rgb(65, 199, 36);  
 -webkit-border-radius: 1px;  
 -moz-border-radius: 1px;  
 border-radius: 1px;  
}  
  
.back .contentbox .toolbox-all .firstRow .BigTool:hover .addButton {  
 display: block;  
}  
  
.back .contentbox .dotted {  
 border: 1px dotted rgb(209, 209, 209);  
 margin-top: 10px;  
 width: 100%;  
 margin-bottom: 10px;  
}  
  
.back .contentbox .commanRow {  
 width: 100%;  
 display: flex;  
 justify-content: space-between;  
 margin-top: 10px;  
 flex-flow: wrap;  
}  
  
.back .contentbox .commanRow .normalTool {  
 width: 300px;  
 height: 70px;  
 position: relative;  
 cursor: pointer;  
 padding: 5px;  
 -webkit-box-sizing: border-box;  
 -moz-box-sizing: border-box;  
 box-sizing: border-box;  
}  
  
.back .contentbox .commanRow .normalTool:hover {  
 outline: 1px solid #dadada;  
}  
  
.back .contentbox .commanRow .normalTool .img {  
 position: relative;  
 display: inline-block;  
 width: 60px;  
 height: 60px;  
 background-color: blue;  
}  
  
.back .contentbox .commanRow .normalTool .text {  
 position: absolute;  
 left: 75px;  
 right: 5px;  
 top: 5px;  
 bottom: 5px;  
}  
  
.back .contentbox .commanRow .normalTool .text .title {  
 line-height: 35px;  
 font-size: 16px;  
}  
  
.back .contentbox .commanRow .normalTool .text .description {  
 line-height: 25px;  
 font-size: 12px;  
 color: #aaa;  
}  
  
.back .contentbox .commanRow .normalTool .addButton {  
 display: none;  
 position: absolute;  
 top: 10px;  
 right: 15px;  
 width: 60px;  
 height: 22px;  
 background-image: linear-gradient(rgb(98, 227, 25) 0%, rgb(68, 208, 27) 100%);  
 font-size: 12px;  
 color: white;  
 text-align: center;  
 line-height: 20px;  
 border: 1px solid rgb(65, 199, 36);  
 -webkit-border-radius: 1px;  
 -moz-border-radius: 1px;  
 border-radius: 1px;  
}  
  
.back .contentbox .commanRow .normalTool:hover .addButton {  
 display: block;  
}  
  
.back .contentbox .titleRow {  
 width: 100%;  
 height: 20px;  
 margin-top: 25px;  
}  
  
.back .contentbox .titleRow .titleRow-left {  
 display: inline-block;  
 width: 2px;  
 height: 100%;  
 background-color: rgb(42, 191, 29);  
}  
  
.back .contentbox .titleRow .titleRow-text {  
 height: 100%;  
 display: inline-block;  
 line-height: 20px;  
 margin-left: 10px;  
 vertical-align: top;  
 font-size: 18px;  
 color: #000;  
}  
  
/\* 有这行才有效，滚动条的宽度 \*/  
.back .contentbox ::-webkit-scrollbar {  
 width: 12px;  
}  
  
/\* 滚动条的背景 \*/  
.back .contentbox ::-webkit-scrollbar-track {  
 background-color: rgb(242, 242, 242);  
}  
  
/\*滚动条\*/  
.back .contentbox ::-webkit-scrollbar-thumb {  
 -webkit-border-radius: 5px;  
 border-radius: 5px;  
 background: rgb(218, 218, 218);  
}  
  
.back .contentbox ::-webkit-scrollbar-button {  
 width: 12px;  
}

JS代码：

无新增，略。

（三）我的工具页面布局

如图：



首先将其分为二部分；

第一部分是上方整体红色方框区域；

* 包含若干个独立按钮，按钮分为图片和下方文字两部分；

第二部分是下方蓝色方框区域；

* 包含左方的编辑按钮和右方的四个快捷按钮区域；
* 左方是图标和文字，图标分为按下和非按下状态；
* 右方是左边的文字和右侧的按钮，按钮又分为图标和文字。按钮在无图标时有占位图标。

先上模板：

*//我的工具，和之前的div.toolbox-all平级***div**.*toolbox-my*.*displayNONE  
 //上方区域* **div**.*toolbox-content  
 //独立按钮* **div**.*tool-my* **div**.*img* **div**.*text* **小清新日历** *//下方区域* **div**.*toolbox-foot  
 //编辑按钮* **div**.*edit* **div**.*img* **div**.*text* **编辑** *//右方区域* **div**.*shortcut  
 //左边的描述文字* **div**.*description* **div**.*text* **主界面快捷入口：** *//右边四个按钮* **div**.*tool-foot* **div**.*img* **div**.*text* **系统急救箱  
 div**.*tool-foot*.*placeholder* **div**.*placeholder-img* **div**.*text* **拖拽到此  
 div**.*tool-foot*.*placeholder* **div**.*placeholder-img* **div**.*text* **拖拽到此  
 div**.*tool-foot*.*placeholder* **div**.*placeholder-img* **div**.*text* **拖拽到此**

然后是CSS的样式：（会涉及图片，后补，图标图片除外）

.back .contentbox .toolbox-my {  
 background-color: white;  
 padding: 30px 40px 90px 40px;  
 position: relative;  
}  
  
.back .contentbox .toolbox-my .toolbox-content {  
 width: 100%;  
 height: 100%;  
 overflow-x: hidden;  
 overflow-y: auto;  
 display: flex;  
 flex-wrap: wrap;  
}  
  
.back .contentbox .toolbox-my .toolbox-content .tool-my {  
 width: 100px;  
 height: 100px;  
 display: inline-block;  
 position: relative;  
 border: 1px solid transparent;  
 -webkit-box-sizing: border-box;  
 -moz-box-sizing: border-box;  
 box-sizing: border-box;  
}  
  
.back .contentbox .toolbox-my .toolbox-content .tool-my:hover {  
 -webkit-border-radius: 2px;  
 -moz-border-radius: 2px;  
 border-radius: 2px;  
 border: 1px solid #DADADA;  
}  
  
.back .contentbox .toolbox-my .toolbox-content .tool-my .img {  
 position: absolute;  
 top: 18px;  
 left: 23px;  
 right: 23px;  
 bottom: 28px;  
 background-image: url("../img/toolsImg.png");  
 background-position: 0 0;  
}  
  
.back .contentbox .toolbox-my .toolbox-content .tool-my .text {  
 position: absolute;  
 bottom: 9px;  
 width: 100%;  
 text-align: center;  
 line-height: 12px;  
 height: 12px;  
 font-size: 12px;  
 color: #7c7c7c;  
}  
  
.back .contentbox .toolbox-my .toolbox-foot {  
 position: absolute;  
 left: 0;  
 right: 0;  
 bottom: 0;  
 height: 95px;  
 -webkit-box-sizing: border-box;  
 -moz-box-sizing: border-box;  
 box-sizing: border-box;  
 border-top: 1px solid rgb(218, 218, 218);  
}  
  
.back .contentbox .toolbox-my .toolbox-foot .edit {  
 width: 32px;  
 height: 52px;  
 position: absolute;  
 top: 22px;  
 left: 30px;  
}  
  
.back .contentbox .toolbox-my .toolbox-foot .edit .img {  
 width: 32px;  
 height: 32px;  
 background-image: url(../img/toolbox.png);  
 background-position: -120px 0;  
}  
  
.back .contentbox .toolbox-my .toolbox-foot .edit .img:hover {  
 background-position: -120px -50px;  
}  
  
.back .contentbox .toolbox-my .toolbox-foot .edit .text {  
 width: 32px;  
 height: 20px;  
 line-height: 20px;  
 vertical-align: bottom;  
 color: rgb(0, 138, 225);  
 font-size: 12px;  
 text-align: center;  
 cursor: default;  
}  
  
.back .contentbox .toolbox-my .toolbox-foot .shortcut {  
 position: absolute;  
 right: 13px;  
 top: 9px;  
 bottom: 15px;  
 width: 450px;  
 display: flex;  
 justify-content: flex-end;  
 align-items: flex-end;  
}  
  
.back .contentbox .toolbox-my .toolbox-foot .shortcut .description {  
 width: 98px;  
 height: 22px;  
}  
  
.back .contentbox .toolbox-my .toolbox-foot .shortcut .description .text {  
 height: 22px;  
 line-height: 22px;  
 font-size: 11px;  
 color: #7c7c7c;  
 vertical-align: top;  
}  
  
.back .contentbox .toolbox-my .toolbox-foot .shortcut .tool-foot {  
 width: 76px;  
 height: 70px;  
 border: 1px solid transparent;  
 -webkit-border-radius: 2px;  
 -moz-border-radius: 2px;  
 border-radius: 2px;  
 -webkit-box-sizing: border-box;  
 -moz-box-sizing: border-box;  
 box-sizing: border-box;  
 position: relative;  
}  
  
.back .contentbox .toolbox-my .toolbox-foot .shortcut .tool-foot:hover {  
 border: 1px solid #dadada;  
}  
  
.back .contentbox .toolbox-my .toolbox-foot .shortcut .tool-foot .img {  
 position: absolute;  
 top: 13px;  
 left: 21px;  
 right: 21px;  
 height: 34px;  
 -webkit-border-radius: 2px;  
 -moz-border-radius: 2px;  
 border-radius: 2px;  
 -webkit-box-sizing: border-box;  
 -moz-box-sizing: border-box;  
 box-sizing: border-box;  
 border: 1px solid transparent;  
 background-image: url("../img/toolsImg.png");  
 background-size: 615px 615px;  
 /\*  
 这个用于计算位置，和实际位置需要乘以61.5%  
 background-position: 0 0;  
 \*/  
}  
  
.back .contentbox .toolbox-my .toolbox-foot .shortcut .tool-foot .text {  
 position: absolute;  
 bottom: 3px;  
 width: 100%;  
 text-align: center;  
 line-height: 12px;  
 height: 12px;  
 font-size: 12px;  
 color: #7c7c7c;  
}  
  
.back .contentbox .toolbox-my .toolbox-foot .shortcut .tool-foot.placeholder:hover {  
 border: 1px solid transparent;  
}  
  
.back .contentbox .toolbox-my .toolbox-foot .shortcut .tool-foot.placeholder .placeholder-img {  
 position: absolute;  
 top: 13px;  
 left: 21px;  
 right: 21px;  
 height: 34px;  
 -webkit-border-radius: 2px;  
 -moz-border-radius: 2px;  
 border-radius: 2px;  
 -webkit-box-sizing: border-box;  
 -moz-box-sizing: border-box;  
 box-sizing: border-box;  
 border: 1px dotted #7c7c7c;  
}

然后对之前的一些css和html修改：

首先，删除掉

**div**.*BigTool* **span**.*img*(style='background-image: url(../img/bigImg03.png)')

注意，以上共三个，最后的图片链接不同，修改为：

**div**.*BigTool* **span**.*img*

修改以下同名样式为：

.back .contentbox .toolbox-all .firstRow .BigTool .img {  
 display: inline-block;  
 position: absolute;  
 width: 100%;  
 height: 100%;  
 background-image: url("../img/bigImg.png");  
 background-position: 0 0;;  
}

然后修改该样式：

.back .contentbox .commanRow .normalTool .img {  
 position: relative;  
 display: inline-block;  
 width: 60px;  
 height: 60px;  
 background-image: url("../img/toolsImg.png");  
 background-position: 0 0;;  
}

于是，我们需要三个图片：

toolbox.png 放零碎的图标，

bigImg.png 放大图标，图标尺寸为300x160

toolsImg.png 放普通图标，尺寸为52x52

下载链接为：

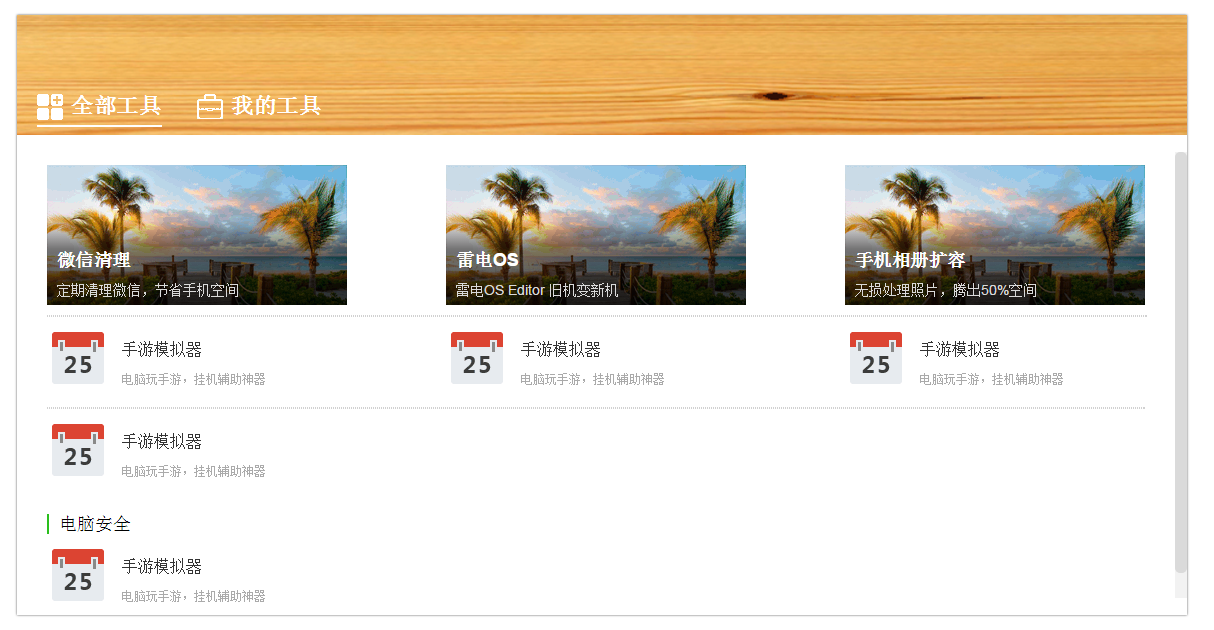
<http://jianwangsan.cn/img/toolbox.png>

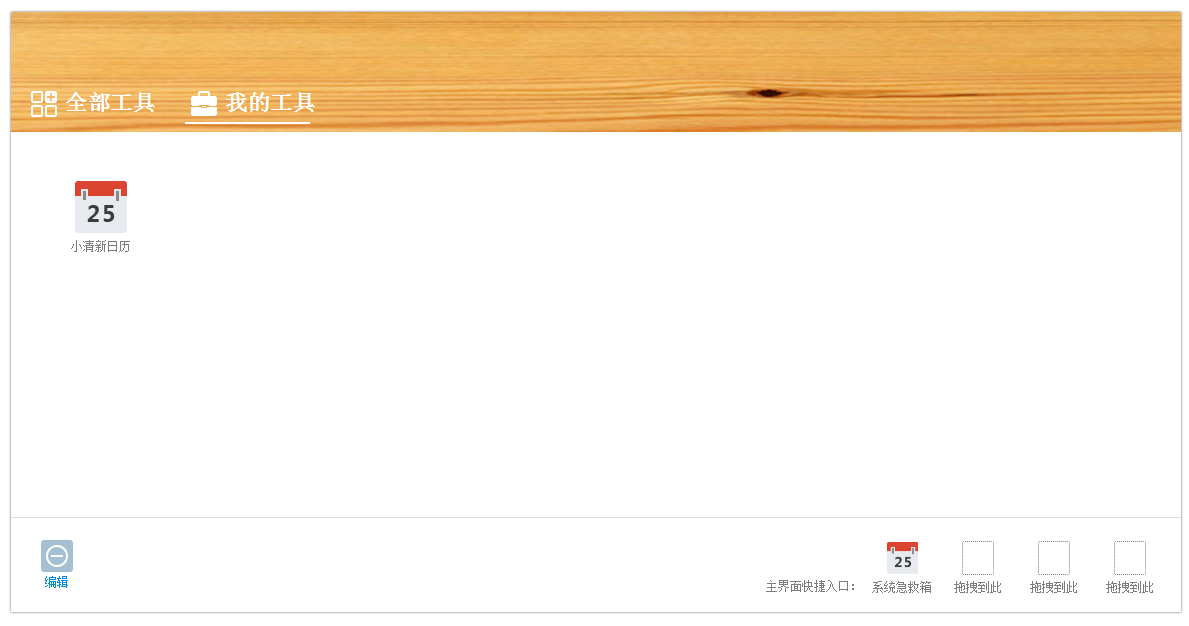
<http://jianwangsan.cn/img/bigImg.png>

<http://jianwangsan.cn/img/toolsImg.png>

放在img文件夹之内食用

目前效果应该如下：





图片之所以重复，是因为使用的是默认第一个位置的图片，在实际生成的时候，会进行修改。

demo链接：

<http://jianwangsan.cn/toolbox>

**（四）制作JSON，自动将图标填充进全部工具**

首先是JSON，因为工具很多，所以JSON内容很长。

具体而言，JSON是一个数组中的对象（只有这一个对象），他有两个属性：BigImg和CommonImg。

这两个属性都是数组类型；

BigImg里面，他用于存放最上面的三个大图标；

CommonImg里面，存放其他工具图标。

BigImg单个数组元素的结构如下：

{  
 **"title"**: **"微信清理"**,  
 **"description"**: **"定期清理微信，节省手机空间"**,  
 **"bigImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"0px"**,  
 **"y"**: **"0px"** }  
 },  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-100px"**,  
 **"y"**: **"0px"** }  
 }  
},

前两个属性看值就知道了；

bingImg和commonImg属性中的ImgPosition中的两个属性，主要是描述这个图标在图片中的位置；

CommonImg结构类似：

{  
 **"title"**: **"手游模拟器"**,  
 **"description"**: **"电脑玩手游，挂机辅助神器"**,  
 **"type"**: **"title"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-100px"**,  
 **"y"**: **"-100px"** }  
 }  
},

只不过少了一个bigImg属性（因为他不需要）；

但多了一个type属性，用于描述其将放置于哪个分类下面。

下面上JSON的全部内容：（共计661行）

[  
 {  
 **"BigImg"**: [  
 {  
 **"title"**: **"微信清理"**,  
 **"description"**: **"定期清理微信，节省手机空间"**,  
 **"bigImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"0px"**,  
 **"y"**: **"0px"** }  
 },  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-100px"**,  
 **"y"**: **"0px"** }  
 }  
 },  
 {  
 **"title"**: **"雷电OS"**,  
 **"description"**: **"雷电OS Editor 旧机变新机"**,  
 **"bigImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-350px"**,  
 **"y"**: **"0px"** }  
 },  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-600px"**,  
 **"y"**: **"-500px"** }  
 }  
 },  
 {  
 **"title"**: **"手机相册扩容"**,  
 **"description"**: **"无损处理图片，腾出50%空间"**,  
 **"bigImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-700px"**,  
 **"y"**: **"0px"** }  
 },  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-700px"**,  
 **"y"**: **"-500px"** }  
 }  
 }  
 ],  
 **"CommonImg"**: [  
 {  
 **"title"**: **"手游模拟器"**,  
 **"description"**: **"电脑玩手游，挂机辅助神器"**,  
 **"type"**: **"title"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-100px"**,  
 **"y"**: **"-100px"** }  
 }  
 },  
 {  
 **"title"**: **"360连回家"**,  
 **"description"**: **"随时随地，清理家中电脑"**,  
 **"type"**: **"title"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-200px"**,  
 **"y"**: **"-100px"** }  
 }  
 },  
 {  
 **"title"**: **"驱动大师"**,  
 **"description"**: **"驱动安装一键解决"**,  
 **"type"**: **"title"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-300px"**,  
 **"y"**: **"-100px"** }  
 }  
 },  
 {  
 **"title"**: **"安全桌面"**,  
 **"description"**: **"一键整理您的桌面"**,  
 **"type"**: **"safe"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-400px"**,  
 **"y"**: **"-100px"** }  
 }  
 },  
 {  
 **"title"**: **"隐私保镖"**,  
 **"description"**: **"五层隐私防护，清理隐私痕迹"**,  
 **"type"**: **"safe"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-500px"**,  
 **"y"**: **"-100px"** }  
 }  
 },  
 {  
 **"title"**: **"防黑加固"**,  
 **"description"**: **"修补可能会被黑客利用的"**,  
 **"type"**: **"safe"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-600px"**,  
 **"y"**: **"-100px"** }  
 }  
 },  
 {  
 **"title"**: **"软件管家"**,  
 **"description"**: **"安全下载，轻松管理您的软件"**,  
 **"type"**: **"safe"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-400px"**,  
 **"y"**: **"0px"** }  
 }  
 },  
 {  
 **"title"**: **"人工服务"**,  
 **"description"**: **"7\*24小时为您解决电脑问题"**,  
 **"type"**: **"safe"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-500px"**,  
 **"y"**: **"0px"** }  
 }  
 },  
 {  
 **"title"**: **"手机助手"**,  
 **"description"**: **"免费手机应用，资源下载平台"**,  
 **"type"**: **"safe"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-700px"**,  
 **"y"**: **"-100px"** }  
 }  
 },  
 {  
 **"title"**: **"弹窗拦截"**,  
 **"description"**: **"拦弹窗，去广告，就是给力"**,  
 **"type"**: **"safe"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-800px"**,  
 **"y"**: **"-100px"** }  
 }  
 },  
 {  
 **"title"**: **"隔离沙箱"**,  
 **"description"**: **"隔离系统真实环境运行软件"**,  
 **"type"**: **"safe"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-900px"**,  
 **"y"**: **"-100px"** }  
 }  
 },  
 {  
 **"title"**: **"软件小助手"**,  
 **"description"**: **"快速启动您常用软件"**,  
 **"type"**: **"safe"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"0px"**,  
 **"y"**: **"-200px"** }  
 }  
 },  
 {  
 **"title"**: **"主页防护"**,  
 **"description"**: **"防止恶意程序篡改浏览器主页"**,  
 **"type"**: **"safe"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-100px"**,  
 **"y"**: **"-200px"** }  
 }  
 },  
 {  
 **"title"**: **"主页修复"**,  
 **"description"**: **"一键解决浏览器主页相关问题"**,  
 **"type"**: **"safe"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-200px"**,  
 **"y"**: **"-200px"** }  
 }  
 },  
 {  
 **"title"**: **"文件解密"**,  
 **"description"**: **"免费还原被木马加密的文件"**,  
 **"type"**: **"safe"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-300px"**,  
 **"y"**: **"-200px"** }  
 }  
 },  
 {  
 **"title"**: **"网络优化"**,  
 **"description"**: **"全新家庭网络管理，秒踢蹭网"**,  
 **"type"**: **"network"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-400px"**,  
 **"y"**: **"-200px"** }  
 }  
 },  
 {  
 **"title"**: **"断网急救箱"**,  
 **"description"**: **"上不了网？就用断网急救箱"**,  
 **"type"**: **"network"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-600px"**,  
 **"y"**: **"0px"** }  
 }  
 },  
 {  
 **"title"**: **"免费WiFi"**,  
 **"description"**: **"电脑变热点，免费无线上网"**,  
 **"type"**: **"network"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-700px"**,  
 **"y"**: **"0px"** }  
 }  
 },  
 {  
 **"title"**: **"宽带测速器"**,  
 **"description"**: **"获取网络带宽和上网速度数值"**,  
 **"type"**: **"network"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-800px"**,  
 **"y"**: **"0px"** }  
 }  
 },  
 {  
 **"title"**: **"WiFi体检"**,  
 **"description"**: **"检测并修复路由器安全隐患"**,  
 **"type"**: **"network"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-500px"**,  
 **"y"**: **"-200px"** }  
 }  
 },  
 {  
 **"title"**: **"流量防火墙"**,  
 **"description"**: **"发现并阻止偷偷占流量的程序"**,  
 **"type"**: **"network"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"0px"**,  
 **"y"**: **"-100px"** }  
 }  
 },  
 {  
 **"title"**: **"LSP修复"**,  
 **"description"**: **"修复网络异常和不能上网"**,  
 **"type"**: **"network"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-600px"**,  
 **"y"**: **"-200px"** }  
 }  
 },  
 {  
 **"title"**: **"DNS优选"**,  
 **"description"**: **"杜绝网络差，启用更优DNS"**,  
 **"type"**: **"network"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-700px"**,  
 **"y"**: **"-200px"** }  
 }  
 },  
 {  
 **"title"**: **"寝室必备"**,  
 **"description"**: **"看看哪个室友在占网速"**,  
 **"type"**: **"network"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-800px"**,  
 **"y"**: **"-200px"** }  
 }  
 },  
 {  
 **"title"**: **"360壁纸"**,  
 **"description"**: **"海量高清壁纸，美化电脑桌面"**,  
 **"type"**: **"system"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-900px"**,  
 **"y"**: **"-200px"** }  
 }  
 },  
 {  
 **"title"**: **"急救盘"**,  
 **"description"**: **"一盘在手，系统无忧"**,  
 **"type"**: **"system"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"0px"**,  
 **"y"**: **"-300px"** }  
 }  
 },  
 {  
 **"title"**: **"任务管理器"**,  
 **"description"**: **"找出当前占用资源的程序"**,  
 **"type"**: **"system"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-300px"**,  
 **"y"**: **"0px"** }  
 }  
 },  
 {  
 **"title"**: **"鲁大师"**,  
 **"description"**: **"辨别硬件真伪，实时监测温度"**,  
 **"type"**: **"system"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-100px"**,  
 **"y"**: **"-300px"** }  
 }  
 },  
 {  
 **"title"**: **"默认软件"**,  
 **"description"**: **"帮您设置常用的默认软件"**,  
 **"type"**: **"system"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-200px"**,  
 **"y"**: **"-300px"** }  
 }  
 },  
 {  
 **"title"**: **"查找大文件"**,  
 **"description"**: **"找出占用磁盘空间的大文件"**,  
 **"type"**: **"system"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-300px"**,  
 **"y"**: **"-300px"** }  
 }  
 },  
 {  
 **"title"**: **"注册表瘦身"**,  
 **"description"**: **"清理无效、错误的注册表键值"**,  
 **"type"**: **"system"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-400px"**,  
 **"y"**: **"-300px"** }  
 }  
 },  
 {  
 **"title"**: **"系统盘瘦身"**,  
 **"description"**: **"通过瘦身解决系统盘空间不足"**,  
 **"type"**: **"system"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-500px"**,  
 **"y"**: **"-300px"** }  
 }  
 },  
 {  
 **"title"**: **"文件恢复"**,  
 **"description"**: **"快速帮您恢复被误删的文件"**,  
 **"type"**: **"system"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-600px"**,  
 **"y"**: **"-300px"** }  
 }  
 },  
 {  
 **"title"**: **"系统急救箱"**,  
 **"description"**: **"查杀顽固木马，修复系统异常"**,  
 **"type"**: **"system"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-600px"**,  
 **"y"**: **"0px"** }  
 }  
 },  
 {  
 **"title"**: **"磁盘擦除"**,  
 **"description"**: **"彻底清除磁盘数据，保护隐私"**,  
 **"type"**: **"system"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-700px"**,  
 **"y"**: **"-300px"** }  
 }  
 },  
 {  
 **"title"**: **"一键装机"**,  
 **"description"**: **"装机必备软件一键搞定"**,  
 **"type"**: **"system"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-800px"**,  
 **"y"**: **"-300px"** }  
 }  
 },  
 {  
 **"title"**: **"右键管理"**,  
 **"description"**: **"管理鼠标的右键菜单"**,  
 **"type"**: **"system"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-900px"**,  
 **"y"**: **"-300px"** }  
 }  
 },  
 {  
 **"title"**: **"系统重装"**,  
 **"description"**: **"无需光盘，恢复系统初始状态"**,  
 **"type"**: **"system"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"0px"**,  
 **"y"**: **"-400px"** }  
 }  
 },  
 {  
 **"title"**: **"网游加速器"**,  
 **"description"**: **"解决游戏卡机掉线，永久免费"**,  
 **"type"**: **"game"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-100px"**,  
 **"y"**: **"-400px"** }  
 }  
 },  
 {  
 **"title"**: **"游戏大厅"**,  
 **"description"**: **"小号多开不串号，键鼠回放爽"**,  
 **"type"**: **"game"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-200px"**,  
 **"y"**: **"-400px"** }  
 }  
 },  
 {  
 **"title"**: **"游戏保险箱"**,  
 **"description"**: **"保护游戏网银账号安全"**,  
 **"type"**: **"game"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-300px"**,  
 **"y"**: **"-400px"** }  
 }  
 },  
 {  
 **"title"**: **"游戏优化器"**,  
 **"description"**: **"一键解决玩游戏卡、慢问题"**,  
 **"type"**: **"game"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-400px"**,  
 **"y"**: **"-400px"** }  
 }  
 },  
 {  
 **"title"**: **"360理财"**,  
 **"description"**: **"360互联网金融服务平台"**,  
 **"type"**: **"smalltools"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-500px"**,  
 **"y"**: **"-400px"** }  
 }  
 },  
 {  
 **"title"**: **"360看图"**,  
 **"description"**: **"防范图片木马，安全查看照片"**,  
 **"type"**: **"smalltools"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-600px"**,  
 **"y"**: **"-400px"** }  
 }  
 },  
 {  
 **"title"**: **"魔法摄像头"**,  
 **"description"**: **"让视频聊天既安全又有趣"**,  
 **"type"**: **"smalltools"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-700px"**,  
 **"y"**: **"-400px"** }  
 }  
 },  
 {  
 **"title"**: **"360云盘"**,  
 **"description"**: **"安全免费，超大空间的云盘"**,  
 **"type"**: **"smalltools"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-200px"**,  
 **"y"**: **"0px"** }  
 }  
 },  
 {  
 **"title"**: **"C盘搬家"**,  
 **"description"**: **"转移系统盘重要资料和文件"**,  
 **"type"**: **"smalltools"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-800px"**,  
 **"y"**: **"-400px"** }  
 }  
 },  
 {  
 **"title"**: **"360问答"**,  
 **"description"**: **"提出你的问题，分分钟有答案"**,  
 **"type"**: **"smalltools"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-900px"**,  
 **"y"**: **"-400px"** }  
 }  
 },  
 {  
 **"title"**: **"苹果设备清理"**,  
 **"description"**: **"清理App垃圾，节省手机空间"**,  
 **"type"**: **"smalltools"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"0px"**,  
 **"y"**: **"-500px"** }  
 }  
 },  
 {  
 **"title"**: **"360压缩"**,  
 **"description"**: **"新一代的压缩软件，永久免费"**,  
 **"type"**: **"smalltools"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-100px"**,  
 **"y"**: **"-500px"** }  
 }  
 },  
 {  
 **"title"**: **"健康精灵"**,  
 **"description"**: **"可爱精灵，助您健康使用电脑"**,  
 **"type"**: **"smalltools"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-200px"**,  
 **"y"**: **"-500px"** }  
 }  
 },  
 {  
 **"title"**: **"小清新日历"**,  
 **"description"**: **"查询天气、农历和节假日"**,  
 **"type"**: **"smalltools"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"0px"**,  
 **"y"**: **"0px"** }  
 }  
 },  
 {  
 **"title"**: **"安全浏览器"**,  
 **"description"**: **"恶意网站拦截，下载保护"**,  
 **"type"**: **"smalltools"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-300px"**,  
 **"y"**: **"-500px"** }  
 }  
 },  
 {  
 **"title"**: **"文件粉碎机"**,  
 **"description"**: **"彻底粉碎无法删除的文件"**,  
 **"type"**: **"smalltools"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-400px"**,  
 **"y"**: **"-500px"** }  
 }  
 },  
 {  
 **"title"**: **"U盘鉴定器"**,  
 **"description"**: **"鉴定U盘真实容量"**,  
 **"type"**: **"smalltools"**,  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-500px"**,  
 **"y"**: **"-500px"** }  
 }  
 }  
 ]  
 }  
]

为了适应type，接下来我们需要改造html模板：

将div.toolbox-all的dom结构改装成如下：

**div**.*toolbox-all  
 //这个是最上面的大图标那一行* **div**.*firstRow  
 //以下是单个按钮  
 //横线那一行，如果是多行app，应考虑使用另外一个* **div**.*dotted* **div**.*commonRow*.*title* **div**.*titleRow* **span**.*titleRow-left* **span**.*titleRow-text* **电脑安全  
 div**.*commonRow*.*safe* **div**.*titleRow* **span**.*titleRow-left* **span**.*titleRow-text* **网络优化  
 div**.*commonRow*.*network* **div**.*titleRow* **span**.*titleRow-left* **span**.*titleRow-text* **系统工具  
 div**.*commonRow*.*system* **div**.*titleRow* **span**.*titleRow-left* **span**.*titleRow-text* **游戏优化  
 div**.*commonRow*.*game* **div**.*titleRow* **span**.*titleRow-left* **span**.*titleRow-text* **实用小工具  
 div**.*commonRow*.*smalltools*

这样通过type来定位该图标被添加的dom位置即可。

**顺便，以上订正了一个之前把commonRow打成commanRow的问题。记得修改样式表（尴尬）**

再顺便订正三个样式需要被调整的地方：

.back .contentbox .commonRow .normalTool .text .title {  
 line-height: 25px;  
 font-size: 16px;  
}  
  
.back .contentbox .commonRow .normalTool .text .description {  
 line-height: 30px;  
 font-size: 12px;  
 color: #aaa;  
}

.back .contentbox .commonRow .normalTool .addButton {  
 display: none;  
 position: absolute;  
 top: 7px;  
 right: 15px;  
 width: 60px;  
 height: 22px;  
 background-image: linear-gradient(rgb(98, 227, 25) 0%, rgb(68, 208, 27) 100%);  
 font-size: 12px;  
 color: white;  
 text-align: center;  
 line-height: 20px;  
 border: 1px solid rgb(65, 199, 36);  
 -webkit-border-radius: 1px;  
 -moz-border-radius: 1px;  
 border-radius: 1px;  
}

下来呢，我们需要读取JSON，然后将其添加入页面之中；

首先，创建一个Tool类，他表示一个图标；

//单个工具  
var Tool = function (obj) {  
 this.**obj** = obj;  
  
 // 0表示未加载到我的工具，1表示加载到我的工具，2表示加载到我的工具的右下小窗处  
 // 为了方便测试，这里先默认设置为1  
 this.**state** = 1;  
  
 //用于在全部工具页面  
 this.createBigImgDom = function (callback) {  
 var self = this;  
 var obj = this.**obj** var str = '<div class="BigTool">' +  
 '<span class="img" style="background-position: ' + obj.bigImg.ImgPosition.**x** + ' ' + obj.bigImg.ImgPosition.**y** + '"></span>' +  
 '<span class="mask"></span>' +  
 '<div class="text">' +  
 '<div class="title">' + obj.**title** + '</div>' +  
 '<div class="description">' + obj.**description** + '</div>' +  
 '</div>' +  
 '<div class="addButton">添加</div>' +  
 '</div>';  
 var node = **$**(str);  
 node.click(function () {  
 if (self.**state**) {  
 callback();  
 }  
 })  
 return node;  
 };  
 this.createNormalTool = function (callback) {  
 var self = this;  
 var obj = this.**obj** var str = '<div class="normalTool">' +  
 '<div class="img" style="background-position: ' + obj.commonImg.ImgPosition.**x** + ' ' + obj.commonImg.ImgPosition.**y** + '"></div>' +  
 '<div class="text">' +  
 '<div class="title">' + obj.**title** + '</div>' +  
 '<div class="description">' + obj.**description** + '</div>' +  
 '</div>' +  
 '<div class="addButton">添加</div>' +  
 '</div>';  
 var node = **$**(str);  
 node.click(function () {  
 if (self.**state**) {  
 callback();  
 }  
 })  
 return node;  
 };  
 this.createSmallTool = function (callback) {  
 var obj = this.**obj** var position\_x = parseInt(obj.commonImg.ImgPosition.**x**) \* 0.615 + "px";  
 var position\_y = parseInt(obj.commonImg.ImgPosition.**y**) \* 0.615 + "px";  
 var str = '<div class="tool-foot">' +  
 '<div class="img" style="background-position: ' + position\_x + ' ' + position\_y + '"></div>' +  
 '<div class="text"></div>' +  
 '</div>';  
 var node = **$**(str);  
 node.click(function () {  
 if (**self**.state) {  
 callback();  
 }  
 })  
 return node;  
 };  
}

他有三个方法，两个属性；

①obj属性是在创建的时候赋值给他的，方便读取创建实例时的初始值。这个初始值就是上面那个JSON中的一个元素（BigImg或CommonImg中的一个元素）；

②state属性表示该按钮状态，具体看注释

三个方法的作用依次为：

①返回一个用于全部工具最顶端的大图标的dom；

②返回一个用于放置在全部工具、我的工具普通位置的dom；

③返回一个用于放在我的工具右下角小位置的dom；

④他们都有一个点击事件，会判断当前状态来进行。为了方便测试，我这里并没有针对性的设置。在之后会进行修改。

我们还缺一些其他的方法，例如将移动用的函数，点击后触发事件的函数等等；

还缺一些属性，例如，设置其目前是否可以移动，目前处于什么位置的东西等等；

等等我们再补全这个Tool类。

然后，我们需要创建一个加载JSON，处理数据的类。

在创建这个类之前，我们建立一个data文件夹，和img、javascripts、stylesheets文件夹平级；

将JSON命名为tools.json，并放于data文件夹中；

下面是处理这个JSON的JS代码类：

var ToolsConfigJsonLoad = function (url) {  
 this.**url** = url ? url : "data/tools.json";  
 this.load = function () {  
 var self = this;  
 **$**.ajax({  
 **url**: self.**url**,  
 **dataType**: "json",  
 **type**: "GET",  
 success: function (data) {  
 self.addToolsInToolbox\_all(data);  
  
 }  
 })  
 };  
  
 //将内容添加到全部工具页面中  
 this.addToolsInToolbox\_all = function (data) {  
 var type = [];  
 data[0].BigImg.forEach(function (obj) {  
 var tool = new Tool(obj);  
 var mixin = new MixinTool(tool);  
 var callback = mixin.mixin()  
 **$**(".firstRow").append(tool.createBigImgDom(callback));  
 })  
 data[0].CommonImg.forEach(function (obj) {  
 if (type.indexOf(obj.**type**) < 0) {  
 type.push(obj.**type**);  
 }  
 var tool = new Tool(obj);  
 var mixin = new MixinTool(tool);  
 var callback = mixin.mixin()  
  
 **$**(".commonRow." + obj.**type**).append(tool.createNormalTool(callback));  
 })  
   
 this.addPlaceHolderWhenOnlyTwoToolsInToolbox\_All(type);  
 this.addDottedLineInToolbox\_All();  
  
 };  
 // 这个目的是当某一行只有两个图标时，创造一个占位的图标  
 this.addPlaceHolderWhenOnlyTwoToolsInToolbox\_All = function (type) {  
 type.forEach(function (obj) {  
 var length = **$**(".commonRow." + obj + " > \*").**length**;  
 if (length % 3 == 2) {  
 **$**(".commonRow." + obj).append(**$**('<div class="normalToolHolder"></div>'));  
 }  
 })  
 };  
  
 // 这个目的是为了给全部工具中的多行工具之间添加分割线  
 this.addDottedLineInToolbox\_All = function () {  
 **$**(".commonRow .normalTool:nth-child(3n+4)").before('<div class="dotted"></div>');  
 }  
}

这部分代码最重要的是load函数；

①他会发起一个ajax请求，来读取这个json；

②ajax请求可以使用用户自己给的url（如果有的话），或者默认url；

③在请求成功后，会对数据进行处理，简单来说，分别遍历BigImg这个属性的每个元素以及CommonImg这个属性的每个元素；

④利用这些元素的数据，生成一个Tool实例，然后又对她做了一些其他事情（具体之后再说），然后生成一个dom对象，插入到指定位置。

在生成工具的时候，显然每个工具的用途是不同的，因此我们希望这个工具在点击的时候，执行不同的处理方法，具体做法有以下几种。

①生成Tool实例的时候，手动给起赋予一个处理函数（缺点：生成代码虎非常长，而且堆积在一起会很乱）；

②将处理函数放在Tool类里，然后生成dom的时候，点击事件触发Tool了的不同方法（缺点：Tool类会非常长，并且可以通过Tool类的实例来调用本来不希望他调用的方法）；

③将所有的处理方法集中在一个类之中，我们需要编辑的时候只需要编辑这个类即可，然后将对应的方法，作为回调函数传递给这个dom的创建函数，在创建函数里，调用这个回调函数（我的选择）。

我这里选择的是第三种方法，因此需要生成一个MixinTool类，他具备将对应的方法返回给对应的Tool类实例的功能。

为了区别不同按钮，我在JSON里每个元素里新加了一个属性ID，具体修改后内容如下；

{  
 **"title"**: **"微信清理"**,  
 **"ID"**:**"No0"**,  
 **"description"**: **"定期清理微信，节省手机空间"**,  
 **"bigImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"0px"**,  
 **"y"**: **"0px"** }  
 },  
 **"commonImg"**: {  
 **"ImgPosition"**: {  
 **"x"**: **"-100px"**,  
 **"y"**: **"0px"** }  
 }  
},

这里的ID的值，就是利用该元素生成Tool实例时，我们写在MixinTool类里，该元素预期拿取的点击事件处理函数。

MixinTool类的代码如下：

var MixinTool = function (tool) {  
 this.mixin = function () {  
 var self = this;  
 if ("ID" in tool.**obj** & tool.**obj**.**ID** in this) {  
 //console.log(self[tool.obj.ID])  
 return self[tool.**obj**.**ID**];  
 } else {  
 return self.default;  
 }  
 };  
  
 this.default = function () {  
 **console**.log("No thing will happen");  
 }  
  
 this.No0 = function () {  
 **console**.log("No 0 you click it");  
 };  
  
 this.No1 = function () {  
 **console**.log("No 1 you click it");  
 };  
  
 this.No2 = function () {  
 **console**.log("No 2 you click it");  
 };  
  
 this.No3 = function () {  
 **console**.log("No 3 you click it");  
 };  
  
 this.No4 = function () {  
 **console**.log("No 4 you click it");  
 };  
  
 this.No5 = function () {  
 **console**.log("No 5 you click it");  
 };  
}

他有一些方法，假如某个Tool实例没有对应的方法，他会执行default这个函数作为点击的响应事件；否则执行对应的。

mixin函数需要显示调用，作为Tool类创建dom结点时的参数使用。

具体如何使用参照上面的例子。

由于我们已经抽象出来多个类了，因此不如将之前页面切换的逻辑也抽象成一个类，具体代码如下：

//注意，这些其实都是全局变量  
var Tab = function () {  
 //以下代码大量考虑到扩展性，例如，可以新增一个tab和content页面  
 this.tabClick = function () {  
 **$**(".tool").click(function () {  
 //这里是上面的图标的逻辑变换  
 if (!(**$**(this.children[0]).hasClass("select"))) {  
 **$**(".select").removeClass("select");  
 **$**(this.children[0]).addClass("select");  
  
 //这里是hover的横线的位置变化  
 var node = **$**(".tool .hover");  
 node.remove();  
 //当动画需要停止的时候，让他停止  
 if ('stop' in node) {  
 node.stop();  
 }  
 node.**css**("left", "0px");  
 **$**(this).append(node);  
  
 //以下应该是切换页面的逻辑  
 //获取切换到哪一个页面，  
 var index = null;  
 for (var i = 0; i < this.parentNode.children.**length**; i++) {  
 if (this == this.parentNode.children[i]) {  
 index = i;  
 }  
 }  
 **$**(".contentbox > div").addClass("displayNONE");  
 **$**(".contentbox > div:nth-child(" + (index + 1) + ")").removeClass("displayNONE");  
  
 }  
 })  
 };  
  
 this.tabMouseEnter = function () {  
 **$**(".tool").mouseenter(function (evt) {  
 //只有当鼠标移动到非当前选中的tab上时，才会移动  
 if (!(**$**(this.children[0]).hasClass("select"))) {  
  
 var self = this;  
 var node = **$**(".tool .hover");  
 var start = null;  
 var end = null;  
  
 var tools = **$**(".toolTab")[0].children  
 for (var i = 0; i < tools.**length**; i++) {  
 if (self == tools[i]) {  
 end = i;  
 } else if (**$**(".select")[0].parentNode == tools[i]) {  
 start = i;  
 }  
 }  
  
 //停止之前的动画  
 if ('stop' in node) {  
 node.stop();  
 }  
 //现在开始动画效果  
 node.animate({**"left"**: (end - start) \* 160 + "px"})  
 }  
 })  
 };  
 this.tabMouseLeave = function () {  
 **$**(".tool").mouseleave(function () {  
 //只有当鼠标移动到非当前选中的tab上时，才会移动  
 if (!(**$**(this.children[0]).hasClass("select"))) {  
 var node = **$**(".tool .hover");  
 //停止之前的动画  
 if ('stop' in node) {  
 node.stop();  
 }  
 node.animate({**"left"**: "0px"})  
 }  
 })  
 }  
}

而调用到目前为止的类和函数，十分简单，如代码：

**$**(**document**).ready(function () {  
 //这里是点击切换显示页面  
 var toolboxTab = new Tab();  
 toolboxTab.tabClick();  
 toolboxTab.tabMouseEnter();  
 toolboxTab.tabMouseLeave();  
  
 var jsonLoad = new ToolsConfigJsonLoad();  
 jsonLoad.load();  
})

目前进度：

①自动生成全部工具里的全部工具；

②给工具添加点击响应事件；

目前还欠缺的内容：

①将全部工具里的工具，添加进我的工具；

②我的工具页面的各种逻辑；

③视情况，让工具可以被添加、或不能被添加（添加按钮在已添加后禁止显示）。

**（五）添加、点击和移动的逻辑**

我反思了一下，在（四）中我写的并不好，事实上，无论是大按钮，还是被添加到我的工具，或者是添加到常用工具栏，他都是一个按钮，因此，应该共享状态，即他们属于同一个tool实例，并能互相影响。

**需求分析：**

在重写Tool类之前，需要明确分析按钮的逻辑。

**在全部工具页面：**

①当按钮未被添加时，鼠标移动上去会有添加按钮显示；

②当按钮未被添加时，鼠标无论点击按钮本身还是点击添加按钮，都执行添加逻辑，将添加按钮显示为取消，执行一次添加动画，添加完成后，按钮隐藏；

③当按钮已被添加时，鼠标点击会启用按钮本身的逻辑（比如打开软件）；

**在我的工具页面：**

①当按钮未被添加时，不显示；

②当按钮被添加时，显示按钮；

③点击按钮时，点击会启用按钮本身的逻辑（比如打开软件）；

④长按按钮，可以拖动按钮，原有按钮位置不被占用，显示为空白；

⑤按钮拖动中时，若经过某个已有按钮的位置，原按钮位置不再被占用，经过的位置被空白占用，相当于把空白占位符从DOM中的原位置挪到了DOM树中的新位置；

⑥按钮拖动时，可以离开工具箱的页面显示范围，但不会显示出来（类似overflow:hidden）的效果；

⑦当按钮拖动到主界面快捷入口的四个图标范围时，若原位置有图标，再该位置图标及右侧所有图标依次向右移动一位；当离开这个位置时，原有图标立刻返回原位置；

⑧主界面快捷入口最多有四个图标，假如新插入一个，那么原本最右边的将被移除；

⑨点击编辑按钮时，所有图标右上角都会出现一个红叉符号；

（10）当编辑状态时，点击主界面快捷入口的四个图标的红叉时，将移除被点击的那个图标；

（11）当编辑状态时，点击我的工具的图标的红叉，将删除该图标（全部工具里的添加按钮恢复）；若该图标在快捷入口也存在时，快捷入口的该图标也被删除。

因此，这个Tool类也能满足以上功能；

**更新模板和样式；**

①要有四种状态的图标，依次为全部工具页面的大图标、普通图标，我的工具页面的普通图标、快捷入口图标；

②并且，要能创建这些图标，并且，由于之前没有设计全部工具页面的取消按钮、我的工具页面里的编辑按钮，因此要添上；

我的工具里图标的html模板：，编辑按钮默认为不显示：

**div**.*tool-my* **div**.*img* **div**.*text* **小清新日历  
 div**.*edit-img*.*displayNONE*

快捷入口的按钮的html模板：

**div**.*tool-foot* **div**.*img* **div**.*text* **系统急救箱  
 div**.*edit-img*.*displayNONE*

取消按钮添加后的大图标：

**div**.*BigTool* **span**.*img* **span**.*mask* **div**.*text* **div**.*title* **div**.*description* **div**.*Button*.*add* **添　加  
 div**.*Button*.*cancel*.*displayNONE* **取　消**

取消按钮添加后的html模板：

**div**.*normalTool* **div**.*img* **div**.*text* **div**.*title* **div**.*description* **div**.*Button*.*add* **添　加  
 div**.*Button*.*cancel*.*displayNONE* **取　消**

另附2个新增的CSS样式：

.back .contentbox .toolbox-my .toolbox-content .tool-my .edit-img {  
 position: absolute;  
 right: 14px;  
 top: 13px;  
 width: 18px;  
 height: 17px;  
 background-image: url(../img/toolbox.png);  
 background-position: -80px -50px;  
}

.back .contentbox .toolbox-my .toolbox-foot .edit-img {  
 position: absolute;  
 right: 14px;  
 top: 10px;  
 width: 18px;  
 height: 17px;  
 background-image: url(../img/toolbox.png);  
 background-position: -80px -50px;  
}

然后是样式修改：

显示大图标的：

把原来的addButton拆分成Button、add和cancel

.back .contentbox .toolbox-all .firstRow .BigTool .Button {  
 display: none;  
 position: absolute;  
 bottom: 10px;  
 right: 12px;  
 width: 60px;  
 height: 22px;  
 font-size: 12px;  
 text-align: center;  
 line-height: 20px;  
 -webkit-border-radius: 1px;  
 -moz-border-radius: 1px;  
 border-radius: 1px;  
}  
  
.back .contentbox .toolbox-all .firstRow .BigTool .Button.add {  
 background-image: linear-gradient(rgb(98, 227, 25) 0%, rgb(68, 208, 27) 100%);  
 color: white;  
 border: 1px solid rgb(65, 199, 36);  
}  
  
.back .contentbox .toolbox-all .firstRow .BigTool .Button.cancel {  
 background-image: linear-gradient(#f3f3f3 0%, #dfdfdf 100%);  
 color: black;  
 border: 1px solid #b6b6b6;  
 display: block;  
}  
  
.back .contentbox .toolbox-all .firstRow .BigTool:hover .add {  
 display: block;  
}

另外一个类似

.back .contentbox .commonRow .normalTool .Button {  
 display: none;  
 position: absolute;  
 top: 7px;  
 right: 15px;  
 width: 60px;  
 height: 22px;  
 font-size: 12px;  
 text-align: center;  
 line-height: 20px;  
 -webkit-border-radius: 1px;  
 -moz-border-radius: 1px;  
 border-radius: 1px;  
}  
  
.back .contentbox .commonRow .normalTool .add {  
 background-image: linear-gradient(rgb(98, 227, 25) 0%, rgb(68, 208, 27) 100%);  
 border: 1px solid rgb(65, 199, 36);  
 color: white;  
}  
  
.back .contentbox .commonRow .normalTool .cancel {  
 background-image: linear-gradient(#f3f3f3 0%, #dfdfdf 100%);  
 color: black;  
 border: 1px solid #b6b6b6;  
 display: block;

}  
  
.back .contentbox .commonRow .normalTool:hover .add {  
 display: block;  
}

为了符合实际需求，我们需要做以下工作：

①重构Tool类（之前实在太简陋了）；

②需要一个函数，专用进行操作；

③需要一个runAfter函数，他的效果是，监听某个对象的某个方法，在该方法执行完后执行runAfter的回调函数；

④然后利用这个runAfter监听一些函数，并在该函数触发时做一些事情。

**重构Tool类：**

他分为以下几个部分：

①变量声明（私有变量）：

var Tool = function (obj, myToolDom) {  
 var self = this;  
 var obj = obj;  
  
 // 0表示未加载到我的工具，1表示加载到我的工具，-1表示添加中（未使用）  
 var state = 0;  
 var BigImgDom = null; //全部工具大图标  
 var NormalToolDom = null; //全部工具小图标  
 var addButton = null; //确认按钮  
 var cancelButton = null; //取消按钮（有，但是实际未使用）  
 var InMyToolDom = null; //我的工具大图标  
 var editButtonInMyToolDom = null; //我的工具页面的编辑红叉按钮  
 var SmallToolDom = null; //快捷入口图标  
 var editButtonInSmallToolDom = null; //快捷入口图标的编辑按钮  
  
 var DomClickEvent = null; //这个是该dom点击事件的回调函数  
 var editing = false; //编辑状态  
 var move = false; //我的工具大图标移动状态  
 var SmallIconMove = false; //我的工具小图标移动状态  
  
 //如果有传参则用传参，如果没有则用默认值  
 var MyToolDom = myToolDom ? myToolDom : **$**(".toolbox-my .toolbox-content");

注意，以上四种图标，最多只会存在三种（所有工具的大图标和小图标只能存在一种状态）

②dom获取

根据实际需求，我们有时候需要获取dom，其中一个是私有的（只在tool类里调用），一个是公有的（会在外面调用），所以声明方式不同：

//获取DOM  
var getDomInAllTools = function () {  
 return BigImgDom ? BigImgDom : NormalToolDom; //因为只有一个存在  
};  
this.getDomInMyTools = function () {  
 return InMyToolDom;  
};

③创建图标

我们自然需要创建图标，四种图标都需要一个函数（因为他们的dom结构不同）；

//全部工具页面  
// 大按钮  
this.createBigImgDom = function (callback) {  
 var str = '<div class="BigTool">' +  
 '<span class="img" style="background-position: ' + obj.bigImg.ImgPosition.**x** + ' ' + obj.bigImg.ImgPosition.**y** + '"></span>' +  
 '<span class="mask"></span>' +  
 '<div class="text">' +  
 '<div class="title">' + obj.**title** + '</div>' +  
 '<div class="description">' + obj.**description** + '</div>' +  
 '</div>' +  
 '<div class="Button add">添　加</div>' +  
 '<div class="Button cancel displayNONE">取　消</div>' +  
 '</div>';  
 BigImgDom = **$**(str);  
 addButton = BigImgDom.**find**(".add");  
 cancelButton = BigImgDom.**find**(".cancel");  
 DomClickEvent = callback;  
 setClickEvent();  
 return getDomInAllTools();  
};  
  
// 普通按钮  
this.createNormalToolDom = function (callback) {  
 var str = '<div class="normalTool">' +  
 '<div class="img" style="background-position: ' + obj.commonImg.ImgPosition.**x** + ' ' + obj.commonImg.ImgPosition.**y** + '"></div>' +  
 '<div class="text">' +  
 '<div class="title">' + obj.**title** + '</div>' +  
 '<div class="description">' + obj.**description** + '</div>' +  
 '</div>' +  
 '<div class="Button add">添　加</div>' +  
 '<div class="Button cancel displayNONE">取　消</div>' +  
 '</div>';  
 NormalToolDom = **$**(str);  
 addButton = NormalToolDom.**find**(".add");  
 cancelButton = NormalToolDom.**find**(".cancel");  
 DomClickEvent = callback;  
 setClickEvent();  
 return getDomInAllTools();  
};

//我的工具页面  
// 创建普通的dom  
var createInMyToolDom = function () {  
 var str = '<div class="tool-my">' +  
 '<div class="img" style="background-position: ' + obj.commonImg.ImgPosition.**x** + ' ' + obj.commonImg.ImgPosition.**y** + '"></div>' +  
 '<div class="text">' + obj.**title** + '</div>' +  
 '<div class="edit-img displayNONE"></div>' +  
 '</div>'  
 var node = **$**(str);  
 return node;  
}  
  
// 创建小的dom  
var createSmallTool = function () {  
 var position\_x = parseInt(obj.commonImg.ImgPosition.**x**) \* 0.615 + "px";  
 var position\_y = parseInt(obj.commonImg.ImgPosition.**y**) \* 0.615 + "px";  
 var str = '<div class="tool-foot">' +  
 '<div class="img" style="background-position: ' + position\_x + ' ' + position\_y + '"></div>' +  
 '<div class="text">' + obj.**title** + '</div>' +  
 '<div class="edit-img displayNONE"></div>' +  
 '</div>';  
 var node = **$**(str);  
 return node;  
};

注：

（1）以上四个创建图标，在全部工具里的两种还额外获取了编辑按钮；

（2）创建全部工具的图标时，顺便获取了响应函数

④响应逻辑：

图标创建了必然需要对她设置事件，

（1）比如全部工具里图标的点击事件：

// 设置全部工具里的点击事件  
var setClickEvent = function () {  
 var node = BigImgDom ? BigImgDom : NormalToolDom;  
 node.click(function (event) {  
 if (state) {  
 DomClickEvent();  
 } else if (state === 0) {  
 self.addDomToMyTools();  
 }  
 });  
};

（2）以上点击事件又分为两种，已添加时，触发正常的响应逻辑（在创建图标时作为参数传进来的回调函数）；

未添加时，用于添加到我的工具里的函数：

//将dom添加到我的工具里  
this.addDomToMyTools = function () {  
 addButton.addClass("displayNONE");  
 state = 1; //设置其状态为已添加  
 //现在是在mytools里添加  
 if (!InMyToolDom) {  
 InMyToolDom = createInMyToolDom();  
 editButtonInMyToolDom = InMyToolDom.**find**(".edit-img");  
 //setMyToolEditEvent(editButtonInMyToolDom); //废弃，同下被整合

//setMyToolClickEvent(InMyToolDom); //废弃，被整合进移动按钮的逻辑中

setMyToolsDomMoveEvent();  
 }  
 MyToolDom.append(InMyToolDom);  
};

（3）这个添加事件又分为：

设置添加按钮为隐藏，设置添加状态为已添加；

没有dom的时候，创建dom并为其绑定事件，绑定的事件有：

【1】点击事件：

**（但事实上这个已废弃，因为后面涉及到移动按钮的函数，被整合在那里了）**

// 设置我的工具页面的两种按钮的点击事件  
var setMyToolClickEvent = function (node) {  
 node.click(function (event) {  
 if (!editing) {  
 DomClickEvent();  
 }  
 });  
};

【2】编辑按钮点击事件：

**（同样已废弃，因为后面涉及到移动按钮的函数，被整合在那里了）**

【3】设置按钮的移动函数，包括以上两个被整合的事件：

//我的工具大图标的移动函数  
var setMyToolsDomMoveEvent = function () {  
 var mouseX = null;  
 var mouseY = null;  
 var startLeft = null;  
 var startTop = null;  
 var placeHholderDom = **$**('<div class="tool-my .ItsplaceHolder"></div>');  
 InMyToolDom.mousedown(function (evt) {  
 if (evt.button != 0) {  
 return;  
 }  
 if (editing & **$**(evt.target)[0] == editButtonInMyToolDom[0]) {  
 self.setStateToUnadd();  
 InMyToolDom.detach();  
 if (**$**(".shortcut .tool-foot").filter(placeHholderDom[0])) {  
 SmallToolDom.detach();  
 **$**(".shortcut").append('<div class="tool-foot placeholder">' +  
 '<div class="placeholder-img"></div>' +  
 '<div class="text">拖拽到此</div>' +  
 '</div>')  
 }  
 **event**.stopPropagation();  
 return;  
 }  
 mouseX = evt.**clientX**; //这里的值是鼠标坐标  
 mouseY = evt.**clientY**;  
 var position = InMyToolDom.**position**();  
 startLeft = position.**left**; //没有px  
 startTop = position.**top**;  
 InMyToolDom.**css**("position", "absolute");  
 InMyToolDom.**css**("left", startLeft + "px");  
 InMyToolDom.**css**("top", startTop + "px");  
 InMyToolDom.after(placeHholderDom);  
 move = true;  
 });  
 **$**(".toolbox-my").mousemove(function (evt) {  
 if (!evt.**buttons** & move) { //只有不在按，且当前是true的时候，才触发  
 move = false;  
 placeHholderDom.after(InMyToolDom);  
 placeHholderDom.remove();  
 InMyToolDom.**css**("position", "relative");  
 InMyToolDom.**css**("left", "0");  
 InMyToolDom.**css**("top", "0");  
 self.InMyToolDomEndMoving();  
 if (mouseX == evt.**clientX** & mouseY == evt.**clientY**) {  
 if (!editing) {  
 DomClickEvent();  
 }  
 }  
 }  
 if (move) {  
 self.InMyToolDomMoving([placeHholderDom, evt]);  
 var offsetX = evt.**clientX** - mouseX;  
 var offsetY = evt.**clientY** - mouseY;  
 InMyToolDom.**css**("left", offsetX + startLeft + "px");  
 InMyToolDom.**css**("top", offsetY + startTop + "px");  
 }  
 });  
 InMyToolDom.mouseup(function (evt) {  
 if (move) {  
 move = false;  
 placeHholderDom.after(InMyToolDom);  
 placeHholderDom.remove();  
 InMyToolDom.**css**("position", "relative");  
 InMyToolDom.**css**("left", "0");  
 InMyToolDom.**css**("top", "0");  
 self.InMyToolDomEndMoving();  
 if (mouseX == evt.**clientX** & mouseY == evt.**clientY**) {  
 if (!editing) {  
 DomClickEvent();  
 }  
 }  
 }  
 })  
};

解释：

以上这个函数干了以下事：

《1》只有鼠标左键按下才有效；

《2》编辑状态时，按编辑按钮是有效的；（因此我们需要一个设置编辑状态的函数）

《3》编辑状态时点击编辑按钮，会删除当前按钮，并用一个空白的按钮替代；

《4》这个dom移除，为了不移除他的事件，因此用的是jquery的detach()方法，并阻止该按钮事件的冒泡；

《5》非编辑状态时，可以移动按钮，原理是设置该按钮的定位为绝对定位，并用一个空白按钮放在该按钮的DOM后起到占位作用，并设置该按钮可移动；

《6》由于代码本身写的并不是很好，因此移动状态被取消时，计算移动距离，如果没有位移偏差，且非编辑状态，触发该按钮的点击事件；（如果要写的好的话，该按钮的移动状态设置，应该在移动位置有偏差之后才设置position为绝对定位）；

《7》移动时触发一个InMyToolDomMoving方法，这个要被runAfter方法所监视，返回值是空白dom和鼠标移动时触发的event事件。该事件会决定该按钮和其他按钮的互动；

《8》移动结束后，会触发一个InMyToolDomEndMoving方法，该方法同样被监视。并且设置移动结束后的鼠标弹起事件（同移动中的那个）。

《9》移动结束后，把图标放在占位图标之前（注意，该占位图标的位置可能移动），并解除移动状态，取消绝对定位，移除占位图标；

【4】然后是我的工具的大图标的移动中事件：

this.InMyToolDomEndMoving = function () {  
 if (!SmallToolDom) {  
 SmallToolDom = createSmallTool();  
 //setMyToolClickEvent(SmallToolDom);  
 editButtonInSmallToolDom = SmallToolDom.**find**(".edit-img");  
 setSmallEditEvent(editButtonInSmallToolDom);  
 setSmallToolDomMoveEvent();  
 }  
 return SmallToolDom;  
};

他会返回小图标，如果没有小图标则创建，并为他绑定事件。

绑定事件有：

《1》点击事件（整合进移动相关函数）；

《2》编辑按钮点击事件：

// 设置小的dom的编辑按钮的点击事件  
var setSmallEditEvent = function (editNode) {  
 editNode.click(function (event) {  
 self.removeSmallTool();  
 editButtonInSmallToolDom.addClass("displayNONE");  
 SmallToolDom.detach();  
 event.stopPropagation();  
 })  
};

《3》移动事件；

var setSmallToolDomMoveEvent = function () {  
 var mouseX = null;  
 var mouseY = null;  
 var startLeft = null;  
 var startTop = null;  
 var placeHholderDom = **$**('<div class="tool-foot placeholder SmallToolHolder">' +  
 '<div class="placeholder-img"></div>' +  
 '<div class="text">拖拽到此</div>' +  
 '</div>');  
 SmallToolDom.mousedown(function (evt) {  
 if (evt.button != 0) {  
 return;  
 }  
 if (editing & **$**(evt.target)[0] == editButtonInSmallToolDom[0]) {  
 **event**.stopPropagation();  
 return;  
 }  
 mouseX = evt.**clientX**; //这里的值是鼠标坐标  
 mouseY = evt.**clientY**;  
 var position = SmallToolDom.**position**();  
 startLeft = position.**left**; //没有px  
 startTop = position.**top**;  
 SmallToolDom.**css**("position", "absolute");  
 SmallToolDom.**css**("left", startLeft + "px");  
 SmallToolDom.**css**("top", startTop + "px");  
 SmallToolDom.after(placeHholderDom);  
 SmallIconMove = true;  
 if (**$**(".SmallToolHolder").**length** > 1) {  
 **$**(".SmallToolHolder")[1].remove();  
 }  
 });  
 **$**(".toolbox-my").mousemove(function (evt) {  
 if (!evt.**buttons** & SmallIconMove) { //只有不在按，且当前是true的时候，才触发  
 SmallIconMove = false;  
 placeHholderDom.before(SmallToolDom);  
 placeHholderDom.remove();  
 SmallToolDom.**css**("position", "relative");  
 SmallToolDom.**css**("left", "0");  
 SmallToolDom.**css**("top", "0");  
 **$**(".SmallToolHolder").remove();  
 if (mouseX == evt.**clientX** & mouseY == evt.**clientY**) {  
 if (!editing) {  
 DomClickEvent();  
 }  
 }  
 }  
 if (SmallIconMove) {  
 self.SmallToolDomMoving([placeHholderDom, evt]);  
 var offsetX = evt.**clientX** - mouseX;  
 var offsetY = evt.**clientY** - mouseY;  
 SmallToolDom.**css**("left", offsetX + startLeft + "px");  
 SmallToolDom.**css**("top", offsetY + startTop + "px");  
 if (**$**(".SmallToolHolder").**length** > 1) {  
 **$**(".SmallToolHolder")[1].remove();  
 }  
 }  
 });  
 SmallToolDom.mouseup(function (evt) {  
 if (SmallIconMove) {  
 SmallIconMove = false;  
 placeHholderDom.before(SmallToolDom);  
 placeHholderDom.remove();  
 SmallToolDom.**css**("position", "relative");  
 SmallToolDom.**css**("left", "0");  
 SmallToolDom.**css**("top", "0");  
 **$**(".SmallToolHolder").remove();  
 if (mouseX == evt.**clientX** & mouseY == evt.**clientY**) {  
 if (!editing) {  
 DomClickEvent();  
 }  
 }  
 }  
 })  
}

由于和大图标的移动按钮事件类似，就不细说了

总之，他在移动的时候，会触发一个SmallToolDomMoving方法，runAfter会监视他。

【5】下来是两个按钮移动时触发的事件和移动结束后触发的事件：

//移动时会触发这个方法  
this.InMyToolDomMoving = function (arr) {  
 return arr;  
};  
  
this.InMyToolDomEndMoving = function () {  
 if (!SmallToolDom) {  
 SmallToolDom = createSmallTool();  
 //setMyToolClickEvent(SmallToolDom);  
 editButtonInSmallToolDom = SmallToolDom.**find**(".edit-img");  
 setSmallEditEvent(editButtonInSmallToolDom);  
 setSmallToolDomMoveEvent();  
 }  
 return SmallToolDom;  
};  
  
this.SmallToolDomMoving = function (arr) {  
 return arr;  
};

⑤其他事件：

//设置编辑或者结束编辑  
this.setEditing = function () {  
 editing = true;  
 editButtonInMyToolDom ? editButtonInMyToolDom.removeClass("displayNONE") : "";  
 editButtonInSmallToolDom ? editButtonInSmallToolDom.removeClass("displayNONE") : "";  
};  
  
this.cancelEditing = function () {  
 editing = false;  
 editButtonInMyToolDom.addClass("displayNONE");  
 editButtonInSmallToolDom ? editButtonInSmallToolDom.addClass("displayNONE") : "";  
}

//设置dom为未添加状态  
this.setStateToUnadd = function () {  
 addButton.removeClass("displayNONE");  
 editButtonInMyToolDom.addClass("displayNONE");  
 if (editButtonInSmallToolDom) {  
 editButtonInSmallToolDom.addClass("displayNONE");  
 }  
 state = 0; //设置其状态为未添加  
}

编辑状态的两个方法用于设置可编辑或者不可编辑；

另一个用于在删除按钮时调用；

——————————————————————————

**重写ToolsConfigJsonLoad函数：**

这个函数做了这些事情：

①读取json；

②把json转化为Tool类的数据来源；

③创建Tool类实例，并监听其事件；

④在创建图标的时候，添加分割线、或者创建占位图标等；

声明这个函数：

var ToolsConfigJsonLoad = function (InMyToolArray, url) {  
 this.**url** = url ? url : "data/tools.json";

这个函数只有一个公有方法，那就是读取json：

this.load = function () {  
 var self = this;  
 **$**.ajax({  
 **url**: self.**url**,  
 **dataType**: "json",  
 **type**: "GET",  
 success: function (data) {  
 addToolsInToolbox\_all(data);  
  
 }  
 })  
};

读取json成功之后调用的函数：

//将内容添加到全部工具页面中  
var addToolsInToolbox\_all = function (data) {  
 var type = [];  
 data[0].BigImg.forEach(function (obj) {  
 var tool = new Tool(obj);  
 var mixin = new MixinTool(obj);  
 var callback = mixin.mixin()  
 listenToolEvent(tool);  
 **$**(".firstRow").append(tool.createBigImgDom(callback));  
 })  
 data[0].CommonImg.forEach(function (obj) {  
 if (type.indexOf(obj.**type**) < 0) {  
 type.push(obj.**type**);  
 }  
 var tool = new Tool(obj);  
 var mixin = new MixinTool(obj);  
 var callback = mixin.mixin()  
 listenToolEvent(tool);  
 **$**(".commonRow." + obj.**type**).append(tool.createNormalToolDom(callback));  
 })  
 addPlaceHolderWhenOnlyTwoToolsInToolbox\_All(type);  
 addDottedLineInToolbox\_All();  
};

监听事件先略过；

添加分割线：

// 这个目的是为了给全部工具中的多行工具之间添加分割线  
var addDottedLineInToolbox\_All = function () {  
 **$**(".commonRow .normalTool:nth-child(3n+4)").before('<div class="dotted"></div>');  
};

为保证美观，添加占位图标：

// 这个目的是当某一行只有两个图标时，创造一个占位的图标  
var addPlaceHolderWhenOnlyTwoToolsInToolbox\_All = function (type) {  
 type.forEach(function (obj) {  
 var length = **$**(".commonRow." + obj + " > \*").**length**;  
 if (length % 3 == 2) {  
 **$**(".commonRow." + obj).append(**$**('<div class="normalToolHolder" style="cursor:default"></div>'));  
 }  
 })  
};

下来事件监听函数的原型：

//参数1是对象，参数2是方法名（字符串），参数三是该方法执行后执行的函数  
var runAfter = function (obj, runEvent, AfterEvent) {  
 var temp = obj[runEvent];  
 obj[runEvent] = function (arguments) {  
 var result = temp(arguments);  
 AfterEvent(obj, result);  
 }  
}

类似dojo的aspect.after方法

最后是利用runAfter函数进行事件监听，调用它时只需要传递Tool类的实例。

//监听事件  
var listenToolEvent = function (tool) {

他包含以下方法：

①当图标添加进我的工具时，将实例添加到一个数组中：

runAfter(tool, "addDomToMyTools", function () {  
 InMyToolArray.push(tool);  
});

②或者是删除我的工具里的按钮时，移除这个实例：

runAfter(tool, "setStateToUnadd", function () {  
 var MyToolIndex = InMyToolArray.indexOf(tool);  
 InMyToolArray.splice(MyToolIndex, 1);  
});

③还需要一个占位图标，用于图标在移动时占位：

var placeHolderInSmall = **$**('<div class="tool-foot placeholder">' +  
 '<div class="placeholder-img"></div>' +  
 '<div class="text">拖拽到此</div>' +  
 '</div>')

④监听图标移动时的方法，他涉及到我的工具页面里大图标的互动，和快捷入口小图标的互动：

runAfter(tool, "InMyToolDomMoving", function (obj, result) {  
 var placeHolder = result[0];  
 var event = result[1];  
 InMyToolArray.forEach(function (tool) {  
 var node = tool.getDomInMyTools();  
 if (node.**css**("position") !== "absolute") {  
 var position = node.**offset**(); //获取相对于文档的位置  
 if (event.**clientY** > position.**top** & event.**clientY** < position.**top** + 100 & event.**clientX** > position.**left** & event.**clientX** < position.**left** + 100) {  
 if (node.**index**() < placeHolder.**index**()) { //根据索引决定放在前还是后面  
 node.before(placeHolder);  
 } else {  
 node.after(placeHolder);  
 }  
 }  
 }  
 });  
  
 var theNodeInSmallTools = false; //是否重合  
 **$**(".tool-foot").toArray().forEach(function (node, index) {  
 var position = **$**(node).**offset**(); //获取相对于文档的位置  
 if (event.**clientY** > position.**top** & event.**clientY** < position.**top** + 70 & event.**clientX** > position.**left** & event.**clientX** < position.**left** + 76) {  
 theNodeInSmallTools = true;  
 if (**$**(node) != placeHolderInSmall) {  
 **$**(node).before(placeHolderInSmall);  
 }  
 }  
 });  
 //如果重合  
 if (theNodeInSmallTools) {  
 **$**(".shortcut .tool-foot:last-child").addClass("displayNONE");  
 } //如果不重合  
 else {  
 placeHolderInSmall.remove();  
 **$**(".shortcut .tool-foot.displayNONE").removeClass("displayNONE");  
 }  
});

⑤当图标停止移动时，需要决定他是否被移动到一个新位置，或者是添加到快捷入口那里；

runAfter(tool, "InMyToolDomEndMoving", function (obj, node) {  
 var sign = false;  
 **$**(".tool-foot").toArray().forEach(function (node, index) {  
 if (**$**(node)[0] == placeHolderInSmall[0]) {  
 sign = true;  
 }  
 })  
 if (sign) {  
 if (node.hasClass("displayNONE")) {  
 node.removeClass("displayNONE");  
 }  
 placeHolderInSmall.before(node);  
 placeHolderInSmall.remove();  
 }  
 if (**$**(".tool-foot").**length** < 4) {  
 var temp = '<div class="tool-foot placeholder">' +  
 '<div class="placeholder-img"></div>' +  
 '<div class="text">拖拽到此</div>' +  
 '</div>';  
 **$**(".shortcut").append(temp);  
 } else if (**$**(".tool-foot").**length** > 4) {  
 **$**(".tool-foot")[4].remove(); //移除第五个  
 } else {  
 **$**(".tool-foot.displayNONE").removeClass("displayNONE");  
 }  
 **$**(".shortcut").append(**$**(".tool-foot.placeholder"));  
})

⑥还有小图标移动时，需要和小图标互动，例如可能需要交换位置：

runAfter(tool, "SmallToolDomMoving", function (obj, result) {  
 var placeHolder = result[0];  
 var event = result[1];  
 **$**(".tool-foot").toArray().forEach(function (node) {  
 var position = **$**(node).**offset**(); //获取相对于文档的位置  
 if (**$**(node).**css**("position") !== "absolute") {  
 if (event.**clientY** > position.**top** & event.**clientY** < position.**top** + 70 & event.**clientX** > position.**left** & event.**clientX** < position.**left** + 76) {  
 if (**$**(node).**index**() < placeHolder.**index**()) { //根据索引决定放在前还是后面  
 **$**(node).before(placeHolder);  
 } else {  
 **$**(node).after(placeHolder);  
 }  
 }  
 }  
 });  
});

——————————————————————

**让工具箱运行起来：**

以上代码，只涉及到工具箱页面各个图标的交互逻辑，但是并没有真正运行起来（例如，没有显式调用load()方法），并且也没有编辑图标的逻辑。

因此，我们需要一个函数补全剩下的部分：

var ToolBoxEvent = function () {  
 var InMyToolArray = [];  
 var jsonLoad = new ToolsConfigJsonLoad(InMyToolArray);  
 jsonLoad.load();  
  
 **$**("#edit").click(function () {  
 //编辑中  
 if (**$**(this).hasClass("editing")) {  
 InMyToolArray.forEach(function (item) {  
 item.cancelEditing();  
 })  
 //设置编辑按钮的样式变更  
 **$**(this).removeClass("editing");  
 var text = **$**(this).**find**(".text");  
 text.text("编辑");  
 text.**css**("width", "32px");  
 text.**css**("margin-left", "0px");  
 } else {  
 InMyToolArray.forEach(function (item) {  
 item.setEditing();  
 })  
 //设置编辑按钮的样式变更  
 **$**(this).addClass("editing");  
 var text = **$**(this).**find**(".text");  
 text.text("退出编辑");  
 text.**css**("width", "52px");  
 text.**css**("margin-left", "-10px");  
 }  
 });  
}

这个函数干了两件事：

①声明一个ToolsConfigJsonLoad类的实例，并调用它的load方法（加载工具箱）；

②设置编辑按钮的事件。

这样的话，工具箱就可以正常跑起来了。

当然，还有一些小BUG，需要被修复，不过这并不影响整体功能的运转。作为一个DEMO，我表示没力气继续调bug了

博客网址：

<http://blog.csdn.net/qq20004604/article/details/52290580>