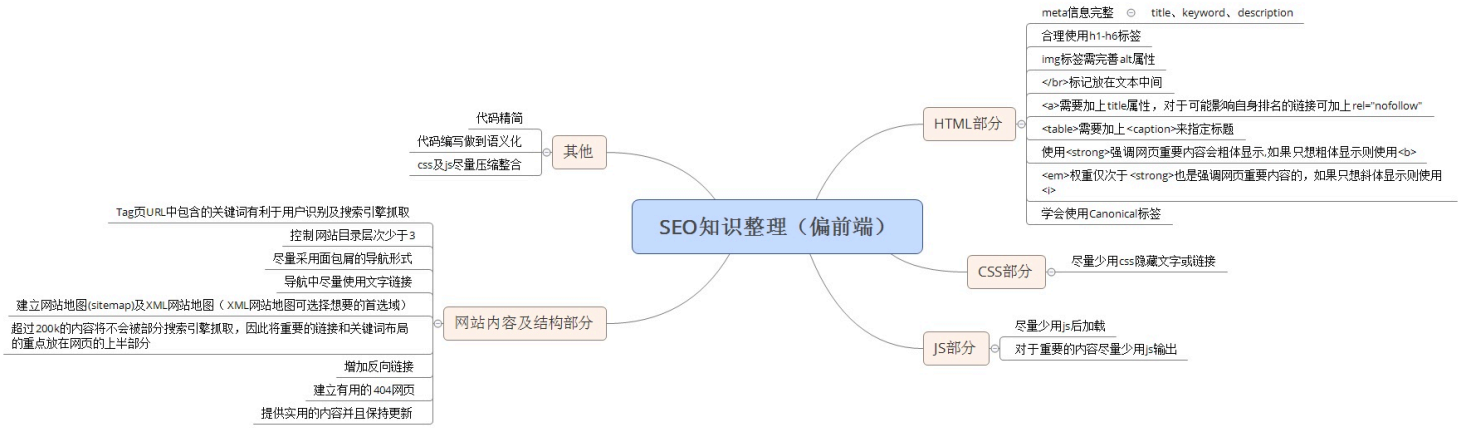
实习总结

一、SEO:

1.是什么?
`Search Engine Optimization`
搜索引擎优化。
对网站内部外部调整优化，改进网站在搜索引擎中的关键词的自然排名，获得更多流量，从而达成网站销售及品牌建设的目的。
2.怎么优化?



参考简书

二、页面在超大屏幕下也要居中

将body设置为`max-width`,`min-width`和`margin:0 auto;`;
参考：同人大赛

由于游戏用户一般是大屏幕，所以雷火的网页对大屏幕的要求比较高，页面最小宽度为1400;

三、如果实现点击壁纸后，直接下载?

图片

用`download`属性。
兼容性:
`var isSupportDownload = 'download' in document.createElement('a');`

文本:

Blob格式。

可以将文本或者JS字符串信息借助`Blob`转换成二进制，然后，作为`<a>`元素的`href`属性，配合`download`属性，实现下载。

```
var funDownload = function (content, filename) {
    // 创建隐藏的可下载链接
   var eleLink = document.createElement('a');
   eleLink.download = filename;
   eleLink.style.display = 'none';

    // 字符内容转变成blob地址
   var blob = new Blob([content]);  // blob的参数必须是数组
   eleLink.href = URL.createObjectURL(blob);

    // 触发点击
   document.body.appendChild(eleLink);
   eleLink.click();
    // 然后移除
   document.body.removeChild(element);
};
```

content指需要下载的文本或字符串内容，filename指下载到系统中的文件名称。

`URL.createObjectURL`会创建一个DOMString，其实包含一个表示参数中给出的对象（e.g.`blob`）的URL。
参考

四、跳转判断

```
window.addHandler(window,"onorientationchange",function(e){
    var horShow = document.getElementById("forhorview");
    if(horShow){
        horShow.style.display = 'block';
    }else{
        var horShow1 = document.createElement('div');
        horShow1.innerHTML = "<div id='forhorview'><p>推荐使用竖屏浏览哦~</p></div>";
        document.body.appendChild(horShow1);
    }
})
```

五、图片预加载

多图加载的思想：
把需要加载的图片放到一个数组里面，等图片全部加载完成了再显示页面。
`Loader`方法是怎么实现的不知道。猜测是使用Promise机制。

```
 trueload:function(){
    var img_list = ["/img/barrage-bg.png","/img/bg-m-1.png","/img/bg-m-2.png","/img/bottom-
bg","/img/bottom-bg.png","/img/btn-liuyan.png","/img/btn-
more.png","/img/cgyl.png","/img/collapse.png","/img/db-
bg.png","/img/dianliang.png","/img/expand.png","/img/hdjs.png","/img/head-
act.png","/img/head.png","/img/home.png","/img/hqmp.png","/img/jcqz.png","/img/jzhd.png","/img/nav
-circle.png","/img/test-bg.png","/img/test.jpg","/img/test22.png","/img/tm-bg1.png","/img/tm-
bg2.png","/img/xqq-circle-1.png","/img/xqq-circle-2.png","/img/xqq-circle-3.png","/img/xqq-
wz.png","/img/xqx.png","/img/xywj.png","/img/zb.png"];
    Loader.show({
        iFileData: img_list,
        bgColor: '#000',//loading背景色值，默认#000
        customAnimation: function(curPer){//加载进度回调函数，取值0~1
            $('.app').show();
        },
        completeCallback: function(){//完成预加载回调函数
        }
    })
}
```

六、移动端穿透问题

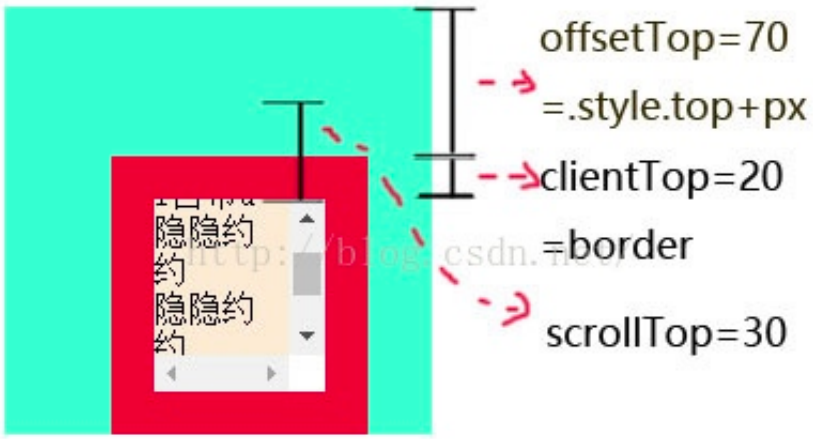
滑动弹框的时候，下层的div也会滚动。

如果垂直偏移已经到达最顶端且手指往下(`y = event.changedTouches[0].pageY`,`y-prevy<0`)，说明已经到最顶部;
如果垂直偏移等于div的高度而且手指往上滑(`y-prevy<0`)，说明已经到最底部;
这两种情况下，执行`preventDefault`;

```
smartScroll: function (contentWrapSelector) {
    var atTop = false,
        atBottom = false,
        prevY = 0,
        $contentWrap = $(contentWrapSelector);  // 在给jquery对象取名的时候在前面加$。 一看就知道是jquery
对象。

    $contentWrap.on('touchstart', function (e) {
        var event = e.originalEvent;
        prevY = event.changedTouches[0].pageY;  //changedTouches是发生变化的触摸点的列表。

        if ($contentWrap.scrollTop() === 0) {  // 滚动条的垂直偏移：scrollTop为0说明已经到达最顶端
            atTop = true;
        }
        if (Math.abs($contentWrap.scrollTop() - ($contentWrap[0].scrollHeight -
$contentWrap[0].clientHeight)) < 2) {
            atBottom = true;
        }
    });
    $contentWrap.on('touchend', function (e) {
        atTop = false;
        atBottom = false;
    });
    $contentWrap.on('touchmove', function (e) {
        var event = e.originalEvent;
        var y = event.changedTouches[0].pageY,
            deltaY = y - prevY;

        if ((atTop && deltaY > 0) || (atBottom && deltaY < 0) || (atTop && atBottom)) {
            e.preventDefault();
        }
        prevY = y;
    });
}

smartScroll(".divName");
```

补充:

```
lientHeight = 可视区高度 + padding

offsetHeight = content高度 + padding + border

当内部元素高度超出了外部元素高度时

    scrollHeight = 内部元素的实际高度(content+padding+border+margin) + 父元素的padding-top值

当内部元素高度小于外部元素高度是

    scrollHeight = 可视区高度 + padding
```




```
scrollHeight - scrollTop = clientHeight: //当这两个条件成立时，也就代表垂直滚动条走到底了

scrollWidth - scrollLeft = clientWidth: //当这两个条件成立时，也就代表水平滚动条走到底了
```

其实就是
`scrollTop = scrollHeight - clientHeight`

参考

七、移动端页面加载后自动播放背景音乐

困难:
移动端不支持背景音乐自动播放
解决思路:
用户第一下点击屏幕的时候开始播放背景音乐（ios无法自动播放音视频，除非是用户触发的动作）；
微信端需要特殊处理一下;
兼容ios和微信，需要用户交互事件触发之后再加载音频;

```
autoplayBgMusic: function () { // 自动播放背景音乐
    var self = this;
    var isFirstTouchScreen = false;
    // 为兼容iOS和微信，需用户交互事件触发后加载音频
    bgMusic.load(); // 需要主动触发下，不然不会加载  bgMusic是个audio标签
    self.playBgMusic();
    document.addEventListener("WeixinJSBridgeReady", function () {
        self.playBgMusic();
    }, false);
    document.addEventListener("touchstart", function () {
        if (isFirstTouchScreen) {
            return;
        }
        isFirstTouchScreen = true;
        self.playBgMusic();
    }, false);

    if (bgMusic.readyState == 4) { //ajax的状态码（完成）响应内容解析完成，可以在客户端调用了
        self.playBgMusic();
    } else {
        bgMusic.addEventListener("canplaythrough", function () {  // 不停地循环播放
            self.playBgMusic();
        }, false);
    }

    // 判断音频是否播放
    bgMusic.addEventListener('timeupdate', function (e) { //timeupdate 事件在音频/视频 (audio/video) 的播放位置发生改变时触发。
        self.setBgMusicPlaying();
    });
    bgMusic.addEventListener('play', function (e) {
        self.setBgMusicPlaying();
    }, false);

    bgMusic.addEventListener('pause', function (e) {
        self.setBgMusicPaused();
    }, false);
    bgMusic.addEventListener("playing", function (e) {
        self.setBgMusicPlaying();
    });
}
```

顺便补充一个知识点；
`addEventListener()`

参数:
`addEventListener(事件名，函数，布尔值)`；
第三个参数，`false`表示在冒泡阶段调用事件处理程序，`true`表示捕获阶段
大多数情况下，都是将事件处理程序添加到事件流的冒泡阶段，这样可以最大限度地兼容各种浏览器。

优点:
可以绑定多个事件；事件会按照添加它们的顺序执行。

注意点:
`addEventListener`绑定的处理事件只能通过`removeEventListener`来移除，参数必须相同。

所以，如果`addEventListener`中参数2是个匿名函数，会移除失败。

比如：

```
var btn = document.getElementById("myBtn");
btn.addEventListener("click",function(){
    alert(666);
},false)
```

移除无效：

```
btn.removeEventListener("click",
    function(){ //没有用！实际上，第二个参数与传入 addEventListener()中的那一个是完全不同的函数
     alert(this.id);
}, false)
```

移除成功：

```
var btn = document.getElementById("myBtn");
var handler = function(){
    alert(666);
}
btn.addEventListener("click",handler,false);

removeEventListener("click",handler,false);  //在 addEventListener()和 removeEventListener()中使 用了相同的函数。
```

《高级教程3》P352

八、跨域问题

下面遇到的这两种都属于***Ajax请求不同源的跨域跨域***。

GET请求

jsonp跨域

动态插入`script`标签，设置`src`属性，向服务器发出请求。发送请求的字符串中有一个callback回调函数。服务器收到请求后，会将数据放在回调函数的参数位置返回。

由于`script`元素请求的脚本，直接作为代码运行。只要浏览器定义了回调函数，作为参数的json数据就会被视为js对象。

原理:

```
所有具有src属性的HTML标签都是可以跨域的，包括img标签和script标签
```

优点:
简单适用，老式浏览器全部支持，服务器改造小。
不需要XMLHttpRequest或ActiveX的支持。

缺点:
只支持GET请求。

POST请求

```
// 允许跨域
crossDomain: true,
// 下面这句话允许跨域的cookie访问
xhrFields: {
    withCredentials: true // 带本地的一些cookie信息
}
```

`crossDomain`主要是CORS跨域ajax请求时使用的。支持post等请求，需要服务器端支持，返回的内容可以是标准的json。

九、cookie过期问题

项目中每次登录验证都向后台发送get请求。不用uri组件判断。。

十、去掉字符串开头和结尾的空白符

模拟字符串的trim()方法:

```
content.replace(/^\s+|\s+$/gm,'');
```