

浏览器兼容性专题

桌面浏览器

IE 浏览器



IE 图标

Internet Explorer，微软公司 Windows 系统的内置浏览器。IE 浏览器在早期因为与 Windows 系统一起捆绑发行，因此战胜了当时最有影响力的由网景公司发布的浏览器 Netscape Navigator（简称 NN）。NN 在浏览器的历史上不应该被遗忘，因为它发明了 Cookie 和 JavaScript。



NN 图标，Navigator 是航海家的意思，浏览网页就像在大海上遨游

IE6 是 IE 最著名的版本，因为它是 Windows XP 时代的产物。因为 XP 死而不僵，IE6 虽然已经被大家深恶痛绝，但依然存在。

IE8 也是一个比较重要的分水岭，它是 Windows 7 时代的产物。

因为捆绑 Windows 销售 IE 是一种不正当竞争行为，微软终于在欧洲被仇人告上法庭，并最终被判败诉。

从 IE9（Windows 7 SP1 后发布，不再支持 Windows XP）开始，微软加速了浏览器的研发速度，相继发布了 IE10、IE11，但这时 IE 已经跟不上时代的潮流。谷歌浏览器（Chrome）、火狐（Firefox）等现代浏览器的兴起最终迫使微软放弃了有几十年历史的 IE，

内部资料

它已经不堪重负，为了保持兼容性，它无法很好地支持现代 Web 标准。



Edge 图标

之后微软又推出了 Edge 浏览器，这时已经到了 Windows 10 的时代，世界已经变了，大家对 IE 已经无感，甚至已经觉得微软和 Windows 不酷了，因为这家公司刚刚经历了移动互联网迷失的 10 年（2007~2016）。Edge 浏览器早已被谷歌的 Chrome 远远甩在了身后。

IE 的内核叫 Trident，即三叉戟（三齿鱼叉）的意思，在中国那是小妖使用的武器，美国军队却很喜欢用这个名字命名飞机、导航甚至核潜艇。所以 Trident 这个名字在美国还是很有名的。



Trident 三叉戟内核的历史太古老了，而微软又不擅长维护老代码，所以 IE 浏览器的问题和 Bug 都很多，也给前端开发带来了无尽的恶梦！

从 IE8 开始，为了兼容 IE6/7，IE 中增加了浏览器模式和文档（本）模式。





看看这些截图都快晕了吧？我来给你解释一下吧：

改变**浏览器模式**会让 IE 重新向服务器发起请求，并且会改变请求头中的 UserAgent 头中关于浏览器版本的信息。这样在 IE8 中，如果你选择了 IE7 浏览器模式，则 UserAgent 头中的浏览器版本就会变成 IE7，从而起到欺骗服务器的效果。

改变**文档（本）模式**则不会重新发起请求，而只会导致浏览器重新按照不同的 Trident 版本重新渲染页面。假设浏览器模式是 IE8，则从服务器请求回来的页面可能是针对是 IE8 生成的，此时你改变文档模式为 IE7，则 IE 浏览器会重新以 IE 的 Trident 内核重新渲染页面。这时你会看到页面刷新一下，但实际上并没有请求服务器。这样就可以实现强制以 IE7 版本的内核来显示页面。

这还不是最复杂的！因为切换浏览器模式、文档模式这些都是给开发者使用的功能，用户不可能学会这些！所以在开发页面时，就需要以写代码的方式让页面在 IE6、7、8、9、10、11 上都能正常显示，这才是最让人头痛的。

怎样才能实现这一点呢？那就需要使用各种技巧，这些技巧以 CSS 为主，一般统称为 CSS Hack。Hack 即骇客、黑客；作为动词的原义是砍、劈、踢、非法侵入，有没有联想到电影里面打打杀杀、破门而入的镜头？

这些技巧不是正常写代码的方式，而是一种缺陷、Bug、漏洞或者后门。在面对兼容性这种残酷的、迫不得已的情况下，竟然成了一种“正常”的代码！

如：正常的样式是这样的：

```
padding: 20px;
```

但 IE6 居然能够识别出这样的代码：

```
_padding: 20px;  
-padding: 20px;
```

IE7 竟然又换了一种：

```
*padding: 20px;
```

IE8、IE9、IE10 也不甘示弱：

```
padding: 20px\0;
```

IE9、IE10 还有这样的：

```
padding: 20px\9\0;
```

偶的个神啊，还让不让人活了？

除此之外，IE6 不支持 !important 这一点也可以拿来利用。这尼玛是个缺陷啊~~

```
padding: 20px!important; IE6 就不认识
```

除了上面样式名前后的各种 Hack，筛选器也表示不服。正常的筛选器是这样的：

```
#eleID {样式} // ID 筛选器
```

IE6 可以识别这样的筛选器：

```
*html #eleID { 样式 }
```

IE7 可以识别：

```
*+html #eleID { 样式 }
```

借助@media 就更神奇了：

```
@media screen\9{...}
```

只对 IE6/7 生效

```
@media \0screen {body { background: red; }}
```

只对 IE8 有效

```
@media \0screen\,screen\9{body { background: blue; }}
```

只对 IE6/7/8 有效

```
@media screen\0 {body { background: green; }}
```

只对 IE8/9/10 有效

```
@media screen and (min-width:\0\0) {body { background: gray; }}
```

只对 IE9/10 有效

```
@media screen and (-ms-high-contrast: active), (-ms-high-contrast: none) {body { background: orange; }}
```

只对 IE10 有效

等等，还没完，IE 的条件注释也没闲着：

只在 IE 下生效

```
<!--[if IE]>
```

这段文字只在 IE 浏览器显示

```
<![endif]-->
```

只在 IE6 下生效

```
<!--[if IE 6]>
```

这段文字只在 IE6 浏览器显示

```
<![endif]-->
```

只在 IE6 以上版本生效

```
<!--[if gte IE 6]>
```

这段文字只在 IE6 以上(包括)版本 IE 浏览器显示

```
<![endif]-->
```

只在 IE8 上不生效

```
<!--[if ! IE 8]>
```

这段文字在非 IE8 浏览器显示

```
<![endif]-->
```

非 IE 浏览器生效

```
<!--[if !IE]>
```

这段文字只在非 IE 浏览器显示

```
<![endif]-->
```

如果你看到这还能扛得住，好，果然有定力！

让我们来总结一下。IE 的 Hack 可以分成以下 4 类：

第 1 类：样式前后缀，如 `_padding: 20px;`

第 2 类：筛选器前缀，如 `*+html #eleID {}`

第 3 类：媒体查询，如 `@media screen\9{ {} }`

第 4 类：条件注释，如 `<!--[if IE 6]>`

最终，你的样式表写成这个鬼样子都是小儿科了：

```
.hack{  
  
    /*注意顺序，否则 IE6/7 下可能无法正确显示，导致结果显示为白色背景*/  
    background-color:red; /* All browsers */  
    background-color:blue !important; /* All browsers but IE6 */  
    *background-color:black; /* IE6, IE7 */  
    +background-color:yellow; /* IE6, IE7*/
```

```
background-color:gray\9; /* IE6, IE7, IE8, IE9, IE10 */
background-color:purple\0; /* IE8, IE9, IE10 */
background-color:orange\9\0;/*IE9, IE10*/
_background-color:green; /* Only works in IE6 */
*+background-color:pink; /* WARNING: Only works in IE7 ? */
}
```

如果到这儿你还没有被 IE 深厚的内力爆掉，鄙人真是佩服佩服，英雄请往这边看：

<http://blog.csdn.net/freshlover/article/details/12132801>

即使你有能力使用这些 Hack 手段，那测试这么多版本的 IE 浏览器也是件令人抓狂的事，还好有个 IETester 可以给我们一点帮助，IETester 将 IE5.5、6、7、8、9、10、11 集成到一个软件中，它是专门为 IE 兼容性测试而开发的一个工具。不爽的是这个工具中的高级功能(查看工具是需要有注册码才能使用的，在没有注册码的情况下可以使用背景颜色和边框线标识标签元素，以帮助确定样式效果)。



IE Tester 图标



使用 IE Tester 可以创建各种版本 IE 浏览器窗口。

Firefox 浏览器



Firefox 图标

这个就是我们熟知的火狐浏览器了。火狐浏览器是由一些支持 Web 标准化的技术大牛开发的开源浏览器,这些人不愿意也无法忍受 IE 的残暴统治(因为这些人来自于网景公司,视微软为仇敌;当然传说有些人还在微软干过,因为受不了微软,所以反叛了),他们决心开发一款最优秀、最标准化的浏览器以奋起反抗,将 IE 拉下马。这些人最终都汇集在了 Mozilla 基金会的旗下(没错,我们经常拜读的 MDN 就是 Mozilla 基金会掌管的)。Mozilla 这个名字来自于 Mosaic,它是世界上第一款浏览器的名字,当然比网景的 NN 还早,因为它是万维网(www)之父蒂姆·伯纳斯·李(Tim Berners-Lee)在 1990 年开发的。那时你爸爸妈妈还在谈恋爱呢!哈哈~~

说起火狐,不得不说说它的名字:最初被命名为 Phoenix,意味火中重生,但和凤凰科技重名了;然后又改为 Firebird(火中的鸟),不幸的是又和一款数据库软件重名了。看来用鸟名就是不靠谱,于是又改成 Firefox。真可谓是一波三折!

Firefox 早期就以支持和发展 Web 标准为己任,所以是支持标准最好的、跨平台的、支持语言最多浏览器,也是对学习 Web 技术的初学者最友好的浏览器。微软中做 IE 浏览器的那伙人最初的时候还不断嘲讽 Firefox 团队,每次 Firefox 发布新版本的时候,都会千里迢迢地给远在欧洲的 Firefox 团队订一个大蛋糕。但后来 Firefox 越做越好,发布新版本的速度也越来越快,使用 Firefox 的人也越来越多,而 IE 则日益衰败,最后微软 IE 那伙人连送蛋糕都觉得不好意思了。

现在 Firefox 以开源方式组织开发和更新,可以保持每 6 周更新一次的速度。Firefox

不仅加速了 H5 的普及，还联合世界上最酷的公司开始将 VR 技术搬到浏览器中（因为 3D 方面 WebGL 已经完成了）。

Firefox 浏览器的内核叫 Gecko，这个词是壁虎的意思。壁虎爬在墙壁上巡游捕食，而人则在爬在网页上浏览信息。



Hello www.....

使用 Firefox 浏览器的用户一般都会使用比较新的版本，因为即然会安装一个系统中没有的浏览器，就一定会更新它。最笨的办法无非就是重新安装。况且 Firefox 浏览器还会自动提示更新。因此 Firefox 就不用考虑那么多 Hack 来解决兼容性问题了。

谷歌浏览器

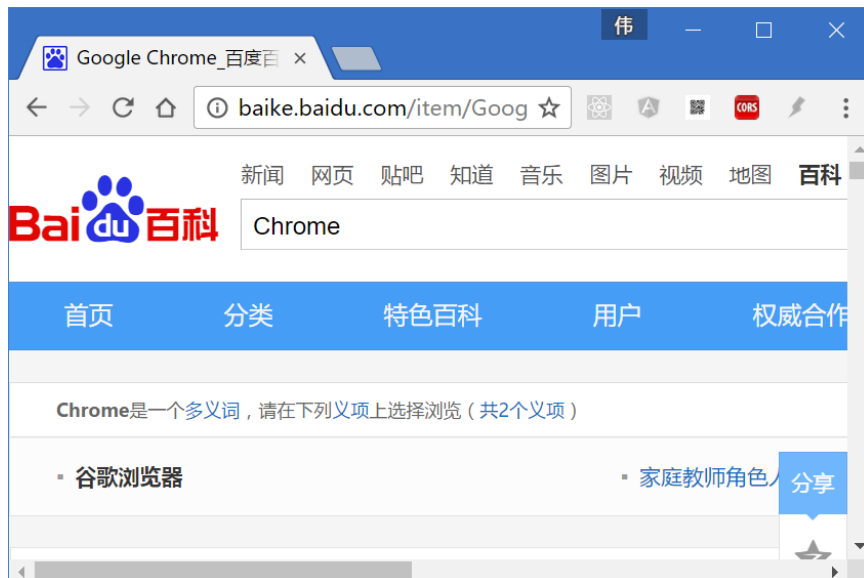


Chrome 图标

谷歌浏览器自然是谷歌公司开发的，它的历史要晚得多，2008 年北京奥运会之后才发布。但因为谷歌在互联网世界堪称老大，其影响力和技术能力自然是首屈一指，而且谷歌向来以简洁的用户界面著称，因此谷歌浏览器一发布就受到了用户的热捧，尤其是不喜欢 IE 的用户。

谷歌浏览器的本名叫 Chrome，Chrome 的本意是铬，铬是一种金属，非常坚硬，是制造高端武器不可或缺的元素。哪怕在其它金属表面薄薄地镀上一层铬就会使其硬度大增，而

且可以变得非常坚硬、锋利。据说中国秦代的宝剑表面就渗入了铬才在保存了 2000 多年后依然锋利无比。因为 Chrome 经常被镀在金属表面，因此计算机技术中也常常用 Chrome 表示窗口界面的“装饰”。



比如上面窗口周围的一圈蓝色区域也被称为 Chrome。而谷歌的浏览器显然要以网页的边框自居，可以理解为 Chrome 就是网页的容器！嗯，很专业！

因为谷歌是一家生长在互联网上的公司，它很擅长从互联网赚钱，而且也不惜重金培育互联网技术。谷歌一直一来都有一个宏愿，就是使用 Web 技术来取代其它的所有的计算机技术。比如谷歌就第一个使用 Web 技术搞出了一套完整的网页版 Office，试图瓦解微软的 Office 系列软件。谷歌还进一步利用 Chrome 浏览器搞出了一个操作系统，Chrome OS，Chrome OS 中一切应用都是 Web 技术做出来的页面，这一招直接从系统上革了微软的命。这直接导致了上网本概念的出现。如果不是后来乔布斯将世界的潮流引向移动互联网，引向触控，Chrome 也许和 Windows 已经厮杀出另一个新世界了。

当然，随着 Chrome 一同而来的，还有 Node.js。为了开发 Chrome，谷歌彻底改造了 JavaScript 的执行方式（即 V8 执行引擎），大大地提升了 JS 的性能，并且还将这项技术开源，于是 Node.js 这项能够让 JS 进行服务端开发的技术就应运而生了。

Chrome 浏览器的内核叫 Webkit，Webkit 本意是 Web 工具包。

Chrome 浏览器还促进了国产浏览器的大发展，如 360 浏览器，QQ 浏览器等。所谓国产浏览器无非是国内软件产商将 IE 内核和 Chrome 内核组合而成的混合多核浏览器。从技术上来说，就是希望用一款软件满足兼容老网页和享受新技术两方面的需要。这些浏览器都有两种模式：兼容模式（IE 的 Trident 内核）、极速模式（Chrome 的 Webkit 内核），之所以选择 Webkit 是因为 Webkit 的代码写得好，容易看懂和维护，嘿嘿。



Safari 图标，Safari 是游猎、观赏的意思；嗯，很有诗义，很苹果

苹果的 Safari 浏览器现在也采用 Webkit 内核（最初使用 Trident 内核）。看来 Webkit 代码写得真是很不错，连挑剔的苹果都相中了。而 Trident 早已经众叛亲离。

下面这个表格总结了一些常见的浏览器的开发公司、内核等信息：

名称	开发公司	内核	最新版
Google Chrome	谷歌	Webkit	55.0
360 极速浏览器	奇虎 360	Trident + Webkit	8.7
百度浏览器	百度	Trident + Webkit	8.7
猎豹浏览器	猎豹移动	Trident + Webkit	6.0.1
傲云浏览器	傲游天下科技	Trident + Webkit	5.0.1
Firefox 火狐浏览器	Mozilla	Gecko	51.0
Internet Explorer	微软	Trident	11
safari	苹果公司	AppleWebKit	5.34
QQ 浏览器	腾讯	Chromium+Trident	9
搜狗浏览器	搜狗公司	Chromium+Trident	7.0.6
世界之窗浏览器	凤凰工作室	Chromium+Trident	7.0
Opera 浏览器	Opera Software ASA	Presto	41.0.2
UC 浏览器	UC Mobile Limited	Blink + Trident	6.1

注：Presto 内核目前只有 Opera 浏览器采用，Chromium 相当于 Chrome 的工程版或称实验版，Blink 也是由 Webkit 衍生而来。

更多的浏览器：

浏览器名	内核
火狐浏览器	Gecko
chrome 和 safari、Comodo Drangon(科摩多龙)，苹果，搜狗高速浏览器 3，快快浏览器，枫树浏览器，云游浏览器，360 极速浏览器，世界之窗极速版，SRWare Iron，猎豹浏览器，RockMelt，QQ 浏览器	WebKit
Opera(欧朋)，NDSBrowser	Presto
IE，360 极速浏览器，猎豹安全浏览器，傲游浏览器，百度浏览器，世界之窗浏览器，2345 浏览器，腾讯 TT，淘宝浏览器，采编读浏览器，搜狗高速浏览器，阿云浏览器，瑞星安全浏览器，Slim Browser，GreenBrowser、爱帆浏览器，115 浏览器，155 浏览器，114 浏览器，N 氧化碳浏览器，糖果浏览器，彩虹浏览器，瑞影浏览器，勇者无疆浏览器，闪游浏览器，蚂蚁浏览器，飞腾浏览器，速达浏览器，佐罗浏览器，海豚浏览器（iPhone/iPad/Android），UC 浏览器	Trident
苹果浏览器之前的版本，后改为 WebKit 内核	Trident

看了这些之后你是不是也有了一种没事做款浏览器玩玩的冲动？

这么多产商都在做浏览器，肯定是有利益在里面的。一方面是想在浏览器上加入网址导航和广告来赚钱，一方面还想收集用户的上网数据。你上网都是通过浏览器来收发数据的，所有的数据浏览器产商都能拿到（HTTPS 加密的数据也不例外）。凡是你能在网页上看到的，浏览器产商就能获取到。比如你的银行帐户信息、你的邮件内容等等。开发一款浏览器软件就能获取全国人民（包括你）的所有的隐私，这个诱惑简直是不能拒绝的！

浏览器很多，内核也很多，各个内核也提供了一些前缀用来区别各自的特殊样式，这些前缀最开始的目的是为了试验新特性，但后来就被用来解决兼容性问题了。如下：

内核	CSS 前缀
Trident（三叉戟，IE）	-ms-
Gecko（壁虎，火狐）	-moz-

WebKit（谷歌 Chrome）	-webkit-
Presto（快，Opera）	-o-

手机浏览器

现在手机阵营可以简化为两个：苹果和安卓。微软、黑莓、塞班都不用考虑了，它们已经被淘汰出局了。

苹果阵营中官方的浏览器 Safari 自然是主流，因为果粉一般都比较崇拜苹果。而且最厉害的一招是，所有第三方浏览器都得经过苹果的审核才能发布到 App Store 上，所以苹果上的浏览器可以比较放心，兼容性问题很少，也比较容易解决。考虑较多的也就是少数顽固不肯更新最新版 iOS 系统的老手机（占比很少，大气一点可以直接忽略）。

而安卓系统也基本是 Webkit 的天下，但是情况比较复杂，老旧版本就更差了，而且老旧版本也不算少。

安卓系统都有系统自带的浏览器，它们是安卓系统自带的，但手机产商通常都会对它们进行定制，增加网址导航等功能。除此之外，还有很多第三方浏览器，看表：

名称	最新版
Chrome	54.0.28
UC 浏览器	11.2.5
搜狗浏览器	5.4.1
QQ 浏览器	7.1.2
百度手机浏览器	7.7.13
猎豹浏览器	4.28.3
火狐浏览器（Gecko 内核）	50.0.1

以上除火狐浏览器之外，其它都是 Webkit 内核或其修改版。国内手机浏览器厂商宣称的“自主内核”如 UC 的 U3 内核、手机 QQ 浏览器的 X5 内核以及华为天天的 T9 内核都

是基于开源内核 Webkit 开发的，是在 Webkit 的基础上进行的二次优化，并不能算是完全的自主内核。

如果说 IE 是桌面浏览器中的坑的话，UC 浏览器则是手机浏览器中的奇葩。当你做好一个手机页面，在各种浏览器上测试都没问题的时候，你信心满满，认为现代手机对 H5 的支持已经相当好了，这时 UC 浏览器却冷不防给你一记闷棍！打得你找不到北。



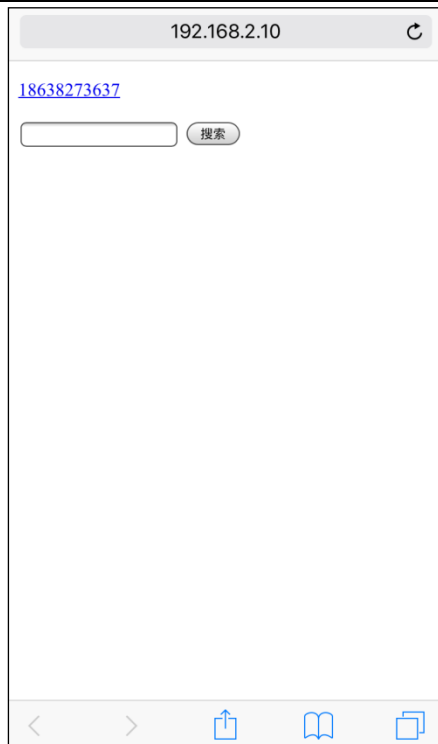
UC 浏览器的安卓版，哦，My GOD，怎么不让它下地狱？

不过值得庆幸的是，UC 已经不再固执，终于在最近刚发布的 11.x 版本开启了“极速模式”，全面支持 H5 了。



Safari 兼容性专区

在手机上开发有一些基本问题值得注意，这就是表单元素和电话号码的识别问题：



这个页面只使用了普通的 p 标签和 input 标签，没有使用 a 标签，也没有使用任何圆角的样式。那为什么手机号会变成超链接，而输入框和按钮会变成圆角的？

是因为在苹果的 safari 手机浏览器，同样包括苹果中的微信浏览器，都会自动开启手机号码和电子邮件地址识别功能，当手机浏览器匹配到页面上的手机号码和电子邮件地址后就会自动将其变成超链接。如果想禁用这个行为，需要在<head>标签中添加以下代码：

```
<meta name="format-detection" content="telephone=no">
```

而输入框和按钮变圆则要通过样式设置来解决：

```
appearance: none;  
-webkit-appearance: none;
```

另外下拉列表也与安卓中的不一样，而且安卓中下拉列表长什么样取决于手机系统，同是安卓，小米、华为、OPPO 各是各的显示方式。苹果则是这样的：



还有,一个普通页面在手机上显示时,内容会缩小。如果你近视比较严重,忘带眼镜了,玩手机岂不是很尴尬,连个字都看不清?表担心!现在的手机浏览器,在 head 标签中都有视口设置来控制缩放。小伙伴儿,这样设置一下你就可以在手机浏览器上看新闻,看小说啦:

```
<meta name="viewport" content="initial-scale=1, maximum-scale=1, user-scalable=no">
```




当我看到上边截图时，是不是有种 APP 的感觉？不过其中的地址栏和导航栏，瞬间又把你拉回了页面的现实。表急，在 Safari 浏览器中，在页面加载的过程中，可以自动隐藏顶部的地址栏和底部的导航栏，干净利落。因为在 iOS 7.1 的 Safari 中为 meta 标签新增 minimal-ui 属性：

```
<meta name="viewport" content="minimal-ui">
```

如果你想在全屏模式 full-screen mode 下删除默认的苹果工具栏和菜单栏，no problem !只需要把是否启动 webapp 功能设置为 yes ,开启对 web app 程序的支持就 ok 了！

```
<meta name="apple-mobile-web-app-capable" content="yes">
<meta name="apple-touch-fullscreen" content="yes">
```



注意过你的 iPhone 上 Safari 状态栏的颜色吗，肯定没注意过，面壁去！整天玩手机都

不关心你的浏览器的心情！浏览器状态栏的样式也是通过 meta 标签控制的：

```
<meta name="apple-mobile-web-app-status-bar-style" content="default" />
<meta name="apple-mobile-web-app-status-bar-style" content="black" />
<meta name="apple-mobile-web-app-status-bar-style" content="black-translucent" />
```

default (白色) black (黑色) black-translucent (灰色半透明)。

说明：除非你先使用 apple-mobile-web-app-capable 指定全屏模式，否则这个 meta 标签不会起任何作用。

如果 content 设置为 default，则状态栏正常显示。如果设置为 blank，则状态栏会有一个黑色的背景。如果设置为 black-translucent，则状态栏显示为黑色半透明。如果设置为 default 或 blank，则页面显示在状态栏的下方，即状态栏占据上方部分，页面占据下方部分，二者没有遮挡对方或被遮挡。如果设置为 black-translucent，则页面会充满屏幕，其中页面顶部会被状态栏遮盖住，会覆盖页面 20px 高度，而 iPhone4 和 iPod touch4 的 Retina 内部资料

屏幕为 40px)。默认值是 default。

整天看小说，肯定收集了不少免费看小说的网站吧，别告诉我你还保存书签，看这里！

在 Safari 中可以使用添加到主屏按钮将网站添加到主屏幕上！下次看的时候直接手机屏幕上就有啦，不用再去翻那一堆书签啦！

当然，前提是你的 apple-mobile-web-app-capable 设置为 yes 了，而通过设置相应 apple-touch-icon 标签，则添加到主屏上的图标就会使用我们指定的图片，是不是在想准备换成谁的照片了 😏

以下是针对分辨率不同的设备，选择一个最优 icon。默认 iPhone 的大小为 60px，ipad 为 76px，retina 屏乘以 2 倍。

```
<link rel="apple-touch-icon" href="touch-icon-iphone.png">
<link rel="apple-touch-icon" sizes="76x76" href="touch-icon-ipad.png">
<link-rel="apple-touch-icon"sizes="120x120" href="touch-icon-iphone-retina.png">
<link-rel="apple-touch-icon"sizes="152x152"href="touch-icon-ipad-retina.png">
```

图标使用的优先级如下：

- 1) 如果没有跟相应设备推荐尺寸一致的图标，那个会优先使用比推荐尺寸大，但最接近推荐尺寸的图标。
- 2) 如果没有比推荐尺寸大的图标，会优先选择最接近推荐尺寸的图标。
- 3) 如些有多个图标符合推荐尺寸，会优先选择包含关键字 precomposed 的图标。
- 4) 如果没有用 link 标签指定图标，会自动搜索网站根目录下以 apple-touch-icon 为前缀的 png 图标。



IE 兼容性专区

本来不想说 IE 的，不过，想想能说的也不多，KO 一个算一个吧。先来避免 IE 使用兼容模式：

```
<meta http-equiv="X-UA-Compatible" content="IE=edge">
```

都说大丈夫能屈能伸的，人家微软偏偏反其道而行，为 IE Mobile 版设置了一个定义宽度标记，这样浏览器不会自动调整文件的大小，也就是说固定大小，不会随着浏览器拉伸缩放。

```
<meta name="MobileOptimized" content="320">
```

总算还有点人性的 IE 啊，至少可以设置个对手持设备友好，不友好试试，分分钟都不用 IE 了！哼！（反正也没人用，呵呵）

```
<meta name="HandheldFriendly" content="true" />
```

UC 兼容性浏览器

看到 UC 浏览器这两字，就让我想起来上次辛辛苦苦写出来的页面，在其他浏览器上都完美呈现，到了 UC，瞬间整个世界都不好了。

强制竖屏<meta name="screen-orientation" content="portrait">

强制横屏<meta name="screen-orientation" content="landscape">

强制全屏<meta name="full-screen" content="yes">

应用模式<meta name="browsermode" content="application">

QQ 兼容性浏览器

强制竖屏<meta name="x5-orientation" content="portrait">

强制全屏<meta name="x5-fullscreen" content="true">

应用模式<meta name="x5-page-mode" content="app">

强制横屏<meta name="x5-orientation" content="landscape">

360 兼容性浏览器

360 在国内市场不小，但褒贬不一，我个人并不喜欢 360，捆绑太多，占地太大，江湖人称“流氓软件”。说是这么说，但 360 浏览器的速度还是有一定口碑的。Come on！开启急速模式（等等，这个，好像是桌面浏览器吧）：

```
<meta name="renderer" content="webkit">
```

手机浏览器常见兼容性问题总结

- 1) 用同等比例的图片在 PC 机上很清楚，但是手机上很模糊，原因是什么呢？经过研究，是 devicePixelRatio 作怪，想让图片在手机里显示更为清晰，必须使用 2x 的背景图来代替 img 标签（一般情况都是用 2 倍）。例如一个 div 的宽高是 100*100，背景图必须得 200*200 然后 background-size:contain; 这样显示出来的图片就比较清晰了。

代码可以如下：

```
background:url(../images/icon/all.png) no-repeat center center;
-webkit-background-size:50px 50px;
background-size:50px 50px;
display:inline-block;
width:100%;
height:50px;
```

或者指定 background-size:contain; 都可以，大家试试！

- 2) 若遇到图片加载很慢的问题，可以考虑用 canvas 方法加载。
- 3) 防止手机中网页放大和缩小。

使用 viewport 使页面禁止缩放。通常把 user-scalable 设置为 0 来关闭用户对页面视图缩放的行为。

```
<meta name="viewport" content="user-scalable=0"/>
```

但是为了更好的兼容，我们会使用完整的 viewport 设置。

复制代码代码如下：

```
<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0" />
```

4) 调用安卓或者 iOS 的拨号功能

只要在 a 标签的 href 中添加 tel:就可以了。

如下：`400-810-6999 转 1034`

拨打手机直接：`点击拨打 15677776767`

5) GPS 定位功能

具体请看：<http://www.jb51.net/html5/69861.html>

6) 上下拉动滚动条时卡顿、慢

```
body {  
    -webkit-overflow-scrolling:touch;  
    overflow-scrolling: touch;  
}
```

Android3+和 iOS5+支持 CSS3 的新属性为 overflow-scrolling

7) 禁止复制、选中文本

```
筛选器 {  
    -webkit-user-select:none;  
    -moz-user-select:none;  
    -khtml-user-select:none;  
    user-select:none;  
}
```

解决移动设备可选中页面文本(视产品需要而定)

8) 长时间按住页面出现闪退，或者禁止弹出编辑工具条

```
筛选器 {  
    -webkit-touch-callout:none;  
}
```

9) iOS 和 Android 下触摸元素时出现半透明灰色遮罩

```
筛选器 {  
    -webkit-tap-highlight-color:rgba(255,255,255,0)
```

```
}
```

设置 alpha 值为 0 就可以去除半透明灰色遮罩 备注 transparent 的属性值在 android 下无效。

10) active 兼容处理 即伪类 :active 失效

方法一：body 添加 ontouchstart : `<body ontouchstart="">`

方法二：js 给 document 绑定 touchstart 或 touchend 事件

```
<style>
  a {
    color:#000;
  }
  a:active {
    color:#fff;
  }
</style>
<a href=foo>bar</a>
<script>
  document.addEventListener('touchstart',function(){},false);
</script>
```

11) 动画定义 3D 启用硬件加速

```
筛选器 {
  -webkit-transform:translate3d(0,0,0)
  transform: translate3d(0,0,0);
}
```

注意：3D 变形会消耗更多的内存与功耗

12) Retina 屏的 1px 边框

```
筛选器 {
  border-width:thin;
}
```

13) 旋转屏幕时，字体大小调整的问题

```
html, body, form, fieldset, p, div, h1, h2, h3, h4, h5, h6{
  -webkit-text-size-adjust:100%;
}
```

14) transition 闪屏

设置内嵌的元素在 3D 空间如何呈现：保留 3D

```
-webkit-transform-style: preserve-3d;
```

设置进行转换的元素的背面在面向用户时是否可见：隐藏

```
-webkit-backface-visibility: hidden;
```

15) 圆角 bug

某些 Android 手机圆角失效：`background-clip: padding-box;`

16) 启动画面

```
<link rel="apple-touch-startup-image" href="start.png"/>
```

iOS 下页面启动加载时显示的画面图片，避免加载时的白屏。

可以通过 media 来指定不同的尺寸。

17) iOS 中 input 键盘事件 keyup、keydown、keypress 支持不是很好。

可以用 html5 的 oninput 事件去代替 keyup

18) H5 网站 input 设置为 type=number 的问题

H5 网页 input 的 type 设置为 number 一般会产生三个问题，一个是 maxlength 属性不好用了。另外一个是在 form 提交的时候，默认给取整了。三是部分安卓手机出现样式问题。

问题一解决，我目前用的是 js。如下

```
<input type="number" oninput="checkTextLength(this ,10)">
function checkTextLength(obj, length) {
    if(obj.value.length > length) {
        obj.value = obj.value.substr(0, length);
    }
}
```

问题二，是因为 form 提交默认做了表单验证，step 默认是 1，要设置 step 属性，假

如保留 2 位小数，写法如下：

```
<input type="number" step="0.01"/>
```

问题三，去除 input 默认样式


```
input[type=number] {  
  -moz-appearance:textfield;  
}  
input[type=number]::-webkit-inner-spin-button,  
input[type=number]::-webkit-outer-spin-button {  
  -webkit-appearance:none;  
  margin:0;  
}
```

19) iOS 键盘字母输入，默认首字母大写

解决方案，设置如下属性 `<input type="text"autocapitalize="off"/>`

20) select 下拉选择设置右对齐

设置如下：

```
select option {  
  direction: rtl;  
}
```

21) 通过 transform 进行 skew 变形，rotate 旋转会造成出现锯齿现象

可以设置如下：

```
-webkit-transform: rotate(-4deg) skew(10deg) translateZ(0);  
transform: rotate(-4deg) skew(10deg) translateZ(0);  
outline: 1px solid rgba(255,255,255,0)
```

22) 移动端点击 300ms 延迟

300ms 尚可接受，不过因为 300ms 产生的问题，我们必须要解决。300ms 导致用户体验并不是很好，解决这个问题，我们一般在移动端用 tap 事件来取代 click 事件。

推荐两个 js，一个是 fastclick，一个是 tap.js

关于 300ms 延迟，具体请看：

<http://thx.github.io/mobile/300ms-click-delay/>

23) 移动端点透问题

案例如下：

```
<div id="haorooms">点透事件测试</div>  
<a href="www.baidu.net">www.baidu.com</a>
```

div 是绝对定位的蒙层,并且 z-index 高于 a。而 a 标签是页面中的一个链接,我们给

div 绑定 tap 事件:

```
$('#haorooms').on('tap',function(){  
    $('#haorooms').hide();  
});
```

我们点击蒙层时 div 正常消失,但是当我们在 a 标签上点击蒙层时,发现 a 链接被触发,这就是所谓的点透事件。

原因:touchstart 早于 touchend 早于 click。亦即 click 的触发是有延迟的,这个时间大概在 300ms 左右,也就是说我们 tap 触发之后蒙层隐藏,此时 click 还没有触发,300ms 之后由于蒙层隐藏,我们的 click 触发到了下面的 a 链接上。

解决:

- (1) 尽量都使用 touch 事件来替换 click 事件。例如用 touchend 事件(推荐)。
- (2) 用 fastclick, <https://github.com/ftlabs/fastclick>
- (3) 用 preventDefault 阻止 a 标签的 click
- (4) 延迟一定的时间(300ms+)来处理事件 (不推荐)
- (5) 以上一般都能解决,实在不行就换成 click 事件。

下面介绍一下 touchend 事件,如下:

```
$("#haorooms").on("touchend",function(event) {  
    event.preventDefault();  
});
```

24) 关于 iOS 与 OS X 端字体的优化(横竖屏会出现字体加粗不一致等)

iOS 浏览器横屏时会重置字体大小,设置 text-size-adjust 为 none 可以解决 iOS 上的问题,但桌面版 Safari 的字体缩放功能会失效,因此最佳方案是将 text-size-adjust 为 100%。

```
-webkit-text-size-adjust:100%;  
-ms-text-size-adjust:100%;
```

```
text-size-adjust:100%;
```

25) 关于 iOS 系统中，中文输入法输入英文时，字母之间可能会出现一个六分之一空格

可以通过正则去掉 `this.value = this.value.replace(/\u2006/g, '');`

26) 移动端 HTML5 audio autoplay 失效问题

这个不是 BUG，由于自动播放网页中的音频或视频，会给用户带来一些困扰或者不必要的流量消耗，所以苹果系统和安卓系统通常都会禁止自动播放和使用 JS 的触发播放，必须由用户来触发才可以播放。

解决方法思路：先通过用户 touchstart 触碰，触发播放并暂停（音频开始加载，后面用 JS 再操作就没问题了）。

解决代码：

```
document.addEventListener('touchstart',function() {  
    document.getElementsByTagName('audio')[0].play();  
    document.getElementsByTagName('audio')[0].pause();  
});
```

27) 移动端 HTML5 input date 不支持 placeholder 问题

这个我感觉没有什么好的解决方案，用如下方法

复制代码代码如下：

```
<input placeholder="Date" class="textbox-n" type="text" onfocus=  
"(this.type='date')" id="date">
```

有的浏览器可能要点击两遍！

28) 部分机型存在 type 为 search 的 input，自带 close 按钮样式修改方法

有些机型的搜索 input 控件会自带 close 按钮（一个伪元素），而通常为了兼容所有浏览器，我们会自己实现一个，此时去掉原生 close 按钮的方法为

```
#Search::-webkit-search-cancel-button{  
    display:none;  
}
```

如果想使用原生 close 按钮，又想使其符合设计风格，可以对这个伪元素的样式进行修

改。

29) 唤起 select 的 option 展开

zepto 方式:

```
$(sltElement).trrrgger("mousedown");
```

原生 js 方式:

```
functionshowDropdown(sltElement) {  
    varevent;  
    event = document.createEvent('MouseEvents');  
    event.initMouseEvent('mousedown',true,true, window);  
    sltElement.dispatchEvent(event);  
};
```

30) 做 App 内嵌页面在 iOS 上显示良好，但在低版本的 Android 上出现很大问题。

低版本 Android 对 H5 的支持比较弱。解决这个问题有两个方案：

方案一、不要使用太新的技术，如 flex、动画等，尽可能使用普通的样式，但这种方案有可能会使页面效果打折扣，不利于用户体验；也会束缚前端开发，降低前端开发的效率和积极性。另外应该尽可能尝试使用-webkit 前缀开启实验特性。

方案二、使用腾讯 X5 WebView 替代 Android 自带的 WebView，这种方案可以达到微信对 H5 的支持程度，因为微信和 QQ 浏览器使用的就是 X5 内核。这种方案是比较推荐的，但需要 Android 开发工程师配合一下。

参考[这里](http://blog.csdn.net/qq_17387361/article/details/52396338)：http://blog.csdn.net/qq_17387361/article/details/52396338

说明

本文由智游教育常伟主编、夏梦君补编、舒涛整理了部分资料

本文从网上摘录了大量的内容，非常感谢这些作者的贡献，但恕不——标明出处了！

欢迎大家共同修改增补！

2016 年 12 月 13 日

