1. 新的Doctype   
尽管使用<!DOCTYPE html>，即使浏览器不懂这句话也会按照标准模式去渲染   
2. Figure元素   
用<figure>和<figcaption>来语义化地表示带标题的图片   
<figure>   
<img src=”path/to/image” alt=”About image” />   
<figcaption>   
<p>This is an image of something interesting. </p>   
</figcaption>   
</figure>   
3. 重新定义的<small>   
<small>已经被重新定义了，现在被用来表示小的排版，如网站底部的版权声明   
4. 去掉link和script标签里面的type属性   
5. 加/不加 括号   
HTML5没有严格的要求属性必须加引号，闭合不闭合，但是建议加上引号和闭合标签   
6. 让你的内容可编辑，只需要加一个contenteditable属性   
7. Email Inputs   
如果我们给Input的type设置为email，浏览器就会验证这个输入是否是email类型，当然不能只依赖前端的校验，后端也得有相应的校验   
8. Placeholders   
这个input属性的意义就是不必通过javascript来做placeholder的效果了   
9. Local Storage   
使用Local Storage可以永久存储大的数据片段在客户端（除非主动删除），目前大部分浏览器已经支持，在使用之前可以检测一下window.localStorage是否存在   
10. 语义化的header和footer   
11. 更多的HTML5表单特性   
12. IE和HTML5   
默认的，HTML5新元素被以inline的方式渲染，不过可以通过下面这种方式让   
其以block方式渲染   
header, footer, article, section, nav, menu, hgroup {   
display: block;   
}   
不幸的是IE会忽略这些样式，可以像下面这样fix:   
document.createElement(”article”);   
document.createElement(”footer”);   
document.createElement(”header”);   
document.createElement(”hgroup”);   
document.createElement(”nav”);   
document.createElement(”menu”);   
13. hgroup   
一般在header里面用来将一组标题组合在一起，如   
<header>   
<hgroup>   
<h1> Recall Fan Page </h1>   
<h2> Only for people who want the memory of a lifetime. </h2>   
</hgroup>   
</header>   
14. Required属性   
required属性定义了一个input是否是必须的，你可以像下面这样声明   
<input type=”text” name=”someInput” required>   
或者   
<input type=”text” name=”someInput” required=”required”>   
15. Autofocus属性   
正如它的词义，就是聚焦到输入框里面   
<input type=”text” name=”someInput” placeholder=”Douglas Quaid” required autofocus>   
16. Audio支持   
HTML5提供了<audio>标签，你不需要再按照第三方插件来渲染音频，大多数现代浏览器提供了对于HTML5 Audio的支持，不过目前仍旧需要提供一些兼容处理，如   
<audio autoplay=”autoplay” controls=”controls”>   
<source src=”file.ogg” /><!–FF–>   
<source src=”file.mp3″ /><!–Webkit–>   
<a href=”file.mp3″>Download this file.</a>   
</audio>   
17. Video支持   
和Audio很像，<video>标签提供了对于video的支持，由于HTML5文档并没有给video指定一个特定的编码，所以浏 览器去决定要支持哪些编码，导致了很多不一致。Safari和IE支持H.264编码的格式，Firefox和Opera支持Theora和Vorbis 编码的格式，当使用HTML5 video的时候，你必须都提供：   
<video controls preload>   
<source src=”cohagenPhoneCall.ogv” type=”video/ogg; codecs=’vorbis, theora’” />   
<source src=”cohagenPhoneCall.mp4″ type=”video/mp4; ’codecs=’avc1.42E01E, mp4a.40.2′” />   
<p> Your browser is old. <a href=”cohagenPhoneCall.mp4″>Download this video instead.</a> </p>   
</video>   
18. 预加载视频   
preload属性就像它的字面意思那么简单，你需要决定是否需要在页面加载的时候去预加载视频   
<video preload>   
19. 显示视频控制   
<video preload controls>   
20. 正则表达式   
由于pattern属性，我们可以在你的markup里面直接使用正则表达式了   
<form action=”" method=”post”>   
<label for=”username”>Create a Username: </label>   
<input type=”text” name=”username” id=”username” placeholder=”4 <> 10″ pattern=”[A-Za-z]{4,10}” autofocus required>   
<button type=”submit”>Go </button>   
</form>   
21. 检测属性支持   
除了Modernizr之外我们还可以通过javascript简单地检测一些属性是否支持，如：   
<script>   
if (!’pattern’ in document.createElement(’input’) ) {   
// do client/server side validation   
}   
</script>   
22. Mark元素   
把<mark>元素看做是高亮的作用，当我选择一段文字的时候，javascript对于HTML的markup效果应该是这样的：   
<h3> Search Results </h3>   
<p> They were interrupted, just after Quato said, <mark>”Open your Mind”</mark>. </p>   
23. 什么时候用<div>   
HTML5已经引入了这么多元素，那么div我们还要用吗？div你可以在没有更好的元素的时候去用。   
24. 想立即使用HTML5?   
不要等2022了，现在就可以使用了，just do it.   
25. 哪些不是HTML5   
1)SVG   
2)CSS3   
3)Geolocation   
4)Client Storage   
5)Web Sockets   
26. Data属性   
<div id=”myDiv” data-custom-attr=”My Value”> Bla Bla </div>   
CSS中使用：   
<style>   
h1:hover:after {   
content: attr(data-hover-response);   
color: black;   
position: absolute;   
left: 0;   
}   
</style>   
<h1 data-hover-response=”I Said Don’t Touch Me!”> Don’t Touch Me </h1>   
27. Output元素   
<output>元素用来显示计算结果，也有一个和label一样的for属性   
28. 用Range Input来创建滑块   
HTML5引用的range类型可以创建滑块，它接受min, max, step和value属性   
可以使用css的:before和:after来显示min和max的值   
<input type=”range” name=”range” min=”0″ max=”10″ step=”1″ value=”">   
input[type=range]:before { content: attr(min); padding-right: 5px;   
}   
input[type=range]:after { content: attr(max); padding-left: 5px;}