

0.

置换元素 <http://www.jianshu.com/p/d99428ca8065>

## 1. h5 的改进:

### 新元素

canvas 画布，2D/3D 制图，用于定义图形（图表等），只是图形容器，必须使用脚本来绘制图形。

### 语义性

audio：用于音频播放。

video：用于视频播放。

article：规定独立的自包含内容。

header：定义文档的页眉，介绍相关信息。

section：定义文档中的节。

footer：定义文档的页脚，通常有文档的作者、版权信息、联系方式等。

nav：定义导航链接

aside：<aside> 标签定义 article 以外的内容。aside 的内容应该与 article 的内容相关。

hgroup：使用 <hgroup> 标签对网页或区段（section）的标题进行组合：

time 时间

<menu>标签定义菜单列表。当希望列出表单控件时使用该标签（H4 弃用，H5 重用）

<q>标签定义一个短的引用；非 H5 标签

<ins>标签定义文档的其余部分之外的插入文本；非 H5 标签)

表单：

datalist 规定输入域的选项列表，通过 option 创建；

keygen 提供一种验证用户的可靠方法，密钥对生成器，私钥存于客户端，公钥发到服务器，用于之后验证客户端证书；

output 元素用于不同类型的输出。

表单控件：

calender

date

time

email

url

search

autofocus, <input type="text" name="user\_name" autofocus="autofocus"/> 页面加载后，浏览器将自动聚焦 user\_name 字段, button 和 input 标签都可以有

getCurrentPosition()用于获得用户的当前位置

getLocation()地图定位

### 新属性

draggable <img draggable="true">拖放

contenteditable 可编辑

### 新事件

ondrag ondrop 拖放

onunload 关闭页面

onresize 窗口大小改变

移除的元素

纯表现的元素：basefont, big, center, font, s, strike, tt, u；

对可用性产生负面影响的元素：frame, frameset, noframes；

新的 DOCTYPE 声明 <!DOCTYPE html>

完全支持 CSS3

本地存储

本地 SQL 数据

Web 应用

如何支持 HTML5 新标签：

IE8/IE7/IE6 支持通过 document.createElement 方法产生的标签，可以利用这一特性让这些浏览器支持 HTML5 新标签，如直接使用成熟的框架 html5shim 框架：

```
<!--[if lt IE 9]> <script> src="http://html5shim.googlecode.com/svn/trunk/html5.js"</script> <![endif]-->
```

在不考虑 H5 的情况下,下列哪些值是可以作为标签的 type 属性的属性值。

text

submit

button

~~date~~

HTML5 <canvas> 元素用于图形的绘制，通过脚本（通常是 JavaScript）来完成，canvas 对象的 getContext() 返回一个用于在画布上绘图的环境

canvas 是 HTML5 的一部分，允许脚本动态渲染位图图像和矢量图

~~canvas 是作为 Flash 的替代技术出现，H5 是 flash 的替代技术~~

canvas 拥有多种绘制路径、矩形、圆形、字符及添加图像的方法

Internet Explorer 自 Internet Explorer9 起已经可以支持 canvas

使用 strokeStyle 属性指定边框颜色

```
var c=document.getElementById("myCanvas");
```

```
var ctx=c.getContext("2d");
```

```
ctx.strokeStyle="#0000ff";
```

```
ctx.strokeRect(20,20,150,100);
```

Ps：

矢量图是根据几何特性来绘制图形，是用线段和曲线描述图像，矢量可以是一个点或一条线，矢量图只能靠软件生成，矢量图文件占用内在空间较小，因为这种类型的图像文件包含独立的分离图像，可以自由无限制的重新组合；

位图图像也称为点阵图像，位图使用我们称为像素的一格一格的小点来描述图像

H5 的拖放事件提供了多个接口：

- 1、drag：当元素或者选中的文本被拖动时触发（每几百毫秒触发一次），应用在被拖拽元素上
- 2、dragend：当拖动操作结束时触发（通过释放鼠标按钮或者点击转义键），应用在被拖拽元素上
- 3、dragenter：当一个被拖动的元素或者选中的文本进入一个有效的放置目标时触发，应用在目标元素上
- 4、dragexit：当元素不再是拖动操作的直接选择元素时触发（很少使用）
- 5、dragleave：当拖动元素或者选中的文本离开有效的放置区域时触发，应用在目标元素上
- 6、dragover：当元素或者选中的文本被拖动到有效放置区域上方时触发（每几百毫秒触发一次），应用在目标元素上
- 7、dragstart：当用户开始拖动元素或者拖动选中文本时触发，应用在被拖拽元素上
- 8、drop：当元素或选中的文本在有效区域放置时触发，应用在目标元素上

~~border~~

## 2. 什么是语义化的 html?

what?

根据内容的结构(内容语义化),选择合适的标签(代码语义化)便于开发者阅读和写出更优雅的代码的同时让浏览器的爬虫和机器很好地解析。

why?

为了在没有 CSS 的情况下,页面也能呈现出很好地内容结构、代码结构:为了裸奔时好看;

用户体验:例如 title、alt 用于解释名词或解释图片信息、label 标签的活用;

有利于 SEO:和搜索引擎建立良好沟通,有助于爬虫抓取更多的有效信息:爬虫依赖于标签来确定上下文和各个关键字的权重;

SEO 是由英文 Search Engine Optimization 缩写而来,中文意译为“搜索引擎优化”!SEO 是指通过对网站进行站内优化和修复(网站 Web 结构调整、网站内容建设、网站代码优化和编码等)和站外优化,从而提高网站的网站关键词排名以及公司产品的曝光度。通过搜索引擎查找信息是当今网民们寻找网上信息和资源的主要手段。而 SEM,搜索引擎营销,就是根据用户使用搜索引擎的方式,利用用户检索信息的机会尽可能将营销信息传递给目标用户。

方便其他设备解析(如屏幕阅读器、盲人阅读器、移动设备)以意义的方式来渲染网页;

便于团队开发和维护,语义化更具可读性,是下一步吧网页的重要动向,遵循 W3C 标准的团队都遵循这个标准,可以减少差异化。

how?

尽可能少的使用无语义的标签 div 和 span;

在语义不明显时,既可以使用 div 或者 p 时,尽量用 p,因为 p 在默认情况下有上下间距,对兼容特殊终端有利;不要使用纯样式标签,如:b、font、u 等,改用 css 设置。

需要强调的文本,可以包含在 strong 或者 em 标签中(浏览器预设样式,能用 CSS 指定就不用他们),strong 默认样式是加粗(不要用 b),em 是斜体(不用 i);

使用表格时,标题要用 caption,表头用 thead,主体部分用 tbody 包围,尾部用 tfoot 包围。表头和一般单元格要区分开,表头用 th,单元格用 td;

表单域要用 fieldset 标签包起来,并用 legend 标签说明表单的用途;

每个 input 标签对应的说明文本都需要使用 label 标签,并且通过为 input 设置 id 属性,在 label 标签中设置 for=someld 来让说明文本和相对应的 input 关联起来。

## 3. 从前端角度出发谈谈做好 seo 需要考虑什么?

语义化 html 标签

合理的 title, description, keywords;

重要的 html 代码放前面

少用 iframe, 搜索引擎不会抓取 iframe 中的内容

图片加上 alt

## 4. 文档类型(DOCTYPE)

作用:doctype 声明位于文档中最前面的位置,处于标签之前,告知浏览器使用的是哪种规范。

如果不声明:不写 doctype,浏览器会进入 quirks mode (混杂模式)。即,如果不声明 doctype,浏览器不引入 w3c 的标准,那么早期的浏览器会按照自己的解析方式渲染页面。浏览器采用自身方式解析页面的行为称为“quirks mode (混杂模式也称怪异模式)”;采用 w3c 方式解析就是“strict mode (标准模式)”。如果完全采用 strictmode(严格模式)就不会出任何的差错,但这样会降低程序的容错率,加重开发人员的难度

用哪种:

没有 doctype 声明的采用 quirks mode 解析

对于有 doctype 的大多数采用 standard mord。

特殊情况:

对于那些浏览器不能识别的 doctype ,浏览器采用 quirks mode ;

没有声明 DTD 或者 html 版本声明低于 4.0 采用 quirks mode, 其他使用 standard mode ;

ie6 中,如果在 doctype 声明前有一个 xml 声明(比如:<?xml version="1.0" encoding="iso-8859-1"?>),则采用 quirks mode 解析

在 HTML 4.01 中有三种 <!DOCTYPE> 声明。在 HTML 4.01 中，<!DOCTYPE> 声明引用 DTD，因为 HTML 4.01 基于 SGML。DTD 规定了标记语言的规则，这样浏览器才能正确地呈现内容。HTML5 不基于 SGML，所以不需要引用 DTD。

在 HTML5 中只有一种：<!DOCTYPE html>

它的责任就是告诉浏览器文档使用哪种 html 或者 xhtml 规范

不同文档模式主要影响 CSS 内容的呈现，尤其是浏览器对盒模型的解析

不同浏览器在混杂模式下的行为差异非常大，如果不使用某些 hack 技术，跨浏览器的行为根本就没有一致性可言。

标准模式与怪异模式的区别:

标准模式：浏览器根据规范呈现页面

混杂模式（怪异模式）：页面以一种比较宽松的兼容方式显示。

他们最大的不同是对盒模型的解析。

在 strict mode 中：width 是内容宽度，也就是说，元素真正的宽度 = margin-left + border-left-width + padding-left + width + padding-right + border-right-width + margin-right;

在 quirks mode 中：width 则是元素的实际宽度，内容宽度 = width - (margin-left + margin-right + padding-left + padding-right + border-left-width + border-right-width)

## 5.HTML 与 XHTML——二者有什么区别

区别：

- 1.所有的标记都必须要有个相应的结束标记
- 2.所有标签的元素和属性的名字都必须使用小写
- 3.所有的 XML 标记都必须合理嵌套
- 4.所有的属性必须用引号""括起来
- 5.把所有<和&特殊符号用编码表示
- 6.给所有属性赋一个值
- 7.不要在注释内容中使"--"
- 8.图片必须有说明文字

## 6. 使用 XHTML 的局限有哪些?

XHTML 较为严格，标签必须闭合，必须要 body, head 等

如果页面使用 'application/xhtml+xml' 一些老的浏览器并不兼容

## 7. 如果网页内容需要多语言,要怎么做?

HTML 中的 META 标签：

<META HTTP-EQUIV="Content-Type" CONTENT="text/html; CHARSET=字符集">

采用统一编码 utf-8 模式

## 8. data-\*属性的作用

html5 规范里增加了一个自定义 data 属性

为前端开发者提供自定义的属性，这些属性集可以通过对象的 dataset 属性获取，不支持该属性的浏览器可以通过 getAttribute 方法获取

<div data-author="david" data-time="2011-06-20" data-comment-num="10">...</div>

div.dataset.commentNum; // 可通过 js 获取 10

data- 为前端开发者提供自定义的属性，这些属性集可以通过对象的 dataset 属性获取，不支持该属性的浏览器可以通过 getAttribute 方法获取。ppk 提到过使用 rel 属性，lightbox 库推广了 rel 属性，HTML5 提供了 data-做替代，这样可以更好地使用自定义的属性

需要注意的是，data- 之后的以连字符分割的多个单词组成的属性，获取的时候使用驼峰风格。

## 9. 如果把 HTML5 看作做一个开放平台，那它的构建模块有哪些？

- 1) Web Storage API
- 2) 基于位置服务 LBS
- 3) 无插件播放音频视频
- 4) 调用相机和 GPU 图像处理单元等硬件设备
- 5) 拖拽和 Form API

## 10. canvas 和 svg 的区别？

svg:

SVG 是一种使用 XML 描述 2D 图形的语言

SVG 基于 XML，这意味着 SVG DOM 中的每个元素都是可用的。所以可以为每个元素附加 JavaScript 事件处理器。在 SVG 中，每个被绘制的图像均被视为对象。如果 SVG 对象的属性发生变化，那么浏览器能够自动重现图像

Canvas

Canvas 通过 js 来绘制 2D 图形

Canvas 是逐像素进行渲染的

在 Canvas 中，一旦图形被绘制完成，它就不会继续得到浏览器的关注。如果其位置发生变化，那么整个场景也需要重新绘制，包括任何或许已被图形覆盖的对象。

区别

Canvas 支持分辨率，SVG 不支持

Canvas 不支持事件处理器，SVG 支持

Canvas 只有弱的文本渲染能力，而 SVG 最适合带有大型渲染区域的应用程序(比如谷歌地图)

Canvas 能够以.png 或.jpg 格式保存结果图像

SVG 的复杂度过高的话会减慢渲染速度(任何过度使用 DOM 的应用都不快)

Canvas 最适合图像密集型的游戏，其中的许多对象会被频繁重绘。而 SVG 不适合游戏应用

Canvas 是基于位图的图像，它不能够改变大小，只能缩放显示；SVG 是基于矢量的，所以它能够很好地处理图形大小的改变

Canvas 提供的功能更原始，适合像素处理，动态渲染和大数据量绘制；SVG 功能更完善，适合静态图片显示，高保真文档查看和打印的应用场景

绘制 Canvas 对象后，不能使用脚本和 CSS 对它进行修改；而 SVG 对象是文档对象模型的一部分，所以可以随时使用脚本和 CSS 修改它们

## 11. src 和 href 的区别？

src 指向外部资源的位置，用于替换当前元素，比如 js 脚本，图片等元素

href 指向网络资源所在的位置，用于在当前文档和引用资源间确定联系，加载 css

## 12. Web Worker 和 WebSocket

worker 主线程是运行在后台的 JavaScript 程序，不会影响页面的性能。

当在 HTML 页面中执行脚本时，页面的状态是不可响应的，直到脚本已完成。web worker 是运行在后台的 JavaScript，独立于其他脚本，不会影响页面的性能。您可以继续做任何愿意做的事情：点击、选取内容等等，而此时 web worker 在后台运行。

1.通过 worker = new Worker( url ) 加载一个 JS 文件来创建一个 worker，同时返回一个 worker 实例。

2.通过 worker.postMessage( data ) 方法来向 worker 发送数据。

3.绑定 worker.onmessage 方法来接收 worker 发送过来的数据。

4.可以使用 worker.terminate() 来终止一个 worker 的执行。

web worker 线程可以嵌套子线程；

WebSocket 是 Web 应用程序的传输协议，它提供了双向的，按序到达的数据流。他是一个 HTML5 协议，WebSocket 的连接是持久的，他通过在客户端和服务端之间保持双工连接，服务端的更新可以被及时推送给客户端，而不需要

客户端以一定时间间隔去轮询。

websocket 允许通过 JavaScript 建立与远程服务器的连接，从而实现客户端与服务器间双向的通信。在 websocket 中有两个方法：

- 1、send() 向远程服务器发送数据
- 2、close() 关闭该 websocket 链接

websocket 同时还定义了几个监听函数

- 1、onopen 当网络连接建立时触发该事件
- 2、onerror 当网络发生错误时触发该事件
- 3、onclose 当 websocket 被关闭时触发该事件

4、onmessage 当 websocket 接收到服务器发来的消息的时触发的事件，也是通信中最重要的一个监听事件。msg.data

websocket 还定义了一个 readyState 属性，这个属性可以返回 websocket 所处的状态：

- 1、CONNECTING(0) websocket 正尝试与服务器建立连接
- 2、OPEN(1) websocket 与服务器已经建立连接
- 3、CLOSING(2) websocket 正在关闭与服务器的连接
- 4、CLOSED(3) websocket 已经关闭了与服务器的连接

websocket 的 url 开头是 ws，如果需要 ssl 加密可以使用 wss，当我们调用 websocket 的构造方法构建一个 websocket 对象 (new WebSocket(url)) 的之后，就可以进行即时通信了。

//实现化 WebSocket 对象，指定要连接的服务器地址与端口

```
socket = new WebSocket("ws://192.168.1.2:8888");
```

### 13.<strong>，<em>和<b>，<i> 标签

<strong> 强调加粗

<b>无意义加粗

<em>强调斜体

<i> 无意义斜体

a 中不能再嵌套 a

<HR> ：表示一条横线

<BR> :可插入一个简单的换行符

<TR>产生一个表行

<ul>ul 只能紧挨着 li、无序列表

<li> </li>

</ul>

<ol>带每行图标、有序列表

<li> </li>

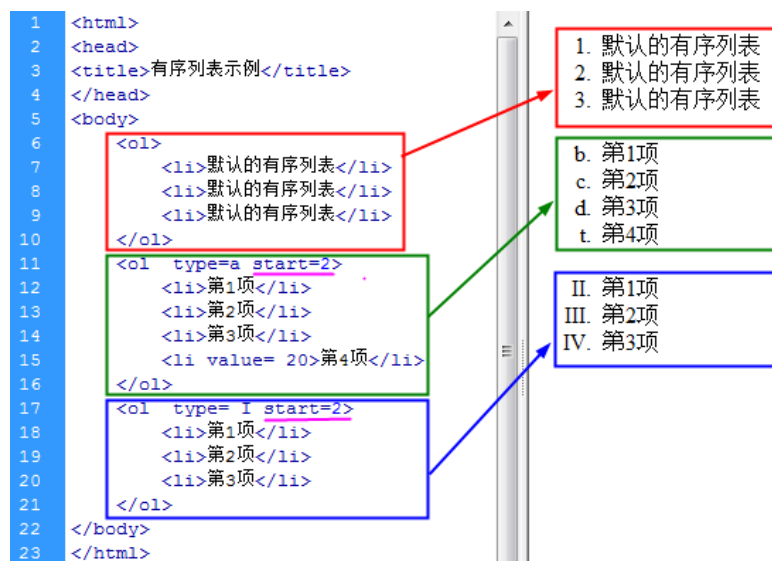
</ol>



```

1 <html>
2 <head>
3 <title>有序列表示例</title>
4 </head>
5 <body>
6   <ol>
7     <li>默认的有序列表</li>
8     <li>默认的有序列表</li>
9     <li>默认的有序列表</li>
10  </ol>
11  <ol type=a start=2>
12    <li>第1项</li>
13    <li>第2项</li>
14    <li>第3项</li>
15    <li value= 20>第4项</li>
16  </ol>
17  <ol type= I start=2>
18    <li>第1项</li>
19    <li>第2项</li>
20    <li>第3项</li>
21  </ol>
22 </body>
23 </html>

```



```

<dl>
<dt> </dt> definition title
<dd> </dd> definition description
</dl>

```

```

<table>
<tr> <td></td> </tr>
</table>
<table>代表表格</table>
<tr>代表表格中的一行</tr>
<td>代表表格中的一列</td>

```

```

<table border="10"> : 10px

```

```

<mark>用于高亮文字
<del>的文字会带删除线
<ins>的文字会带下划线

```

h1-h6 标签可定义标题

应该将 h1 用作主标题 (最重要的), 其后是 h2 (次重要的), 再其次是 h3, 以此类推。  
 应该根据页面中内容的层级关系来决定使用合适的标题标签。  
 网页中最上层的标题是 title。

<code>&lt;em&gt;</code>	把文本定义为强调的内容。
<code>&lt;strong&gt;</code>	把文本定义为语气更强的强调的内容。

<a href="#"><code>&lt;dfn&gt;</code></a>	定义一个定义项目。
<a href="#"><code>&lt;code&gt;</code></a>	定义计算机代码文本。
<a href="#"><code>&lt;samp&gt;</code></a>	定义样本文本。
<a href="#"><code>&lt;kbd&gt;</code></a>	定义键盘文本。它表示文本是从键盘上键入的。它经常用在与计算机相关的文档或手册中。
<a href="#"><code>&lt;var&gt;</code></a>	定义变量。您可以将此标签与 <code>&lt;pre&gt;</code> 及 <code>&lt;code&gt;</code> 标签配合使用。
<a href="#"><code>&lt;cite&gt;</code></a>	定义引用。可使用该标签对参考文献的引用进行定义，比如书籍或杂志的标题。

`<svg>` 标签的目的是什么？用可伸缩矢量图来画图或文本

`<blockquote>`引用内容`</blockquote>`

## bootstrap

bootstrap 框架的话 small 是可以用来表示副标题的

bootstrap 可以给文字加上背景

主要蓝（bg-primary）、成功绿（bg-success）、信息蓝（bg-info）、警告黄（bg-warning）、危险红（bg-danger）  
list-inline 类是 bootstrap 排版的 css 类, list-inline 原理是通过设置 display: inline-block; 并添加少量的内补(padding), 将所有元素放置于同一行。

.text-muted:文字的颜色换成灰色 /.text-primary:文字的颜色变成蓝色 /.text-success:文字的颜色变成绿色 /.text-info:文字的颜色变成暗蓝色 /.text-warning:文字颜色变成暗黄色 /.text-danger : 文字的颜色变成红色

html5 标准的链接默认都是带下划线的

bootstrap 链接只有在 hover 状态下才带下划线

center-block 是 bootstrap 的样式，可以让块元素居中。

Html5 中`<center>`也可以让块元素居中

Bootstrap 在 input 两侧添加额外元素

```
<div class="input-group">
  <span class="input-group-addon">$</span>
  <input type="text" class="form-control" aria-label="Amount (to the nearest dollar)">
  <span class="input-group-addon">.00</span>
</div>
```

```
<button type="button" class="close"><span aria-hidden="true">×</span><span class="sr-only">Close</span></button><br>可以用来表示一个关闭按钮
```

```
<span class="caret"></span>可以用来表示一个三角符号
```

navbar 可使用 navbar-left 类来定义

## 组件排列

通过添加 `.navbar-left` 和 `.navbar-right` 工具类让导航链接、表单、按钮或文本对齐。两个类都会通过 CSS 设置特定方向的浮动样式。例如，要对齐导航链接，就要把它们放在个分开的、应用了工具类的 `<ul>` 标签里。

这些类是 `.pull-left` 和 `.pull-right` 的 mixin 版本，但是他们被限定在了媒体查询（media query）中，这样可以更容易的在各种尺寸的屏幕上处理导航条组件。



	Chrome	Firefox	Internet Explorer	Opera	Safari
Android	✓ 支持	✓ 支持	N/A	✗ 不支持	N/A
iOS	✓ 支持	N/A		✗ 不支持	✓ 支持
Mac OS X	✓ 支持	✓ 支持		✓ 支持	✓ 支持
Windows	✓ 支持	✓ 支持	✓ 支持	✓ 支持	✗ 不支持

bootstrap 中，一个元素需要在打印时隐藏，使用的类是 hidden-print

Bootstrap 框架的网格系统工作原理如下：

1、数据行 (.row) 必须包含在容器（.container）中，以便为其赋予合适的对齐方式和内距 (padding)。

如：<div class="container">

<div class="row"></div>

</div>

2、在行 (.row) 中可以添加列 (.column)，但列数之和不能超过平分的总列数，比如 12。

如：<div class="container">

<div class="row">

<div class="col-md-4"></div>

<div class="col-md-8"></div>

3、具体内容应当放置在列容器（column）之内，而且只有列（column）才可以作为行容器 (.row) 的直接子元素

4、通过设置内距 (padding) 从而创建列与列之间的间距。然后通过为第一列和最后一列设置负值的外距 (margin) 来抵消内距(padding) 的影响

AB :栅格系统的工作原理中有，为了创建列 (colume) 之间的间隔 需要设置 padding,而且需要设置 margin 为负值，消除第一个和最后一个之间 padding 的影响。而 padding 是自动设置的，margin 需要手动修改；

C : bootstrap 将全局的 font-size 设置为 14px;

D: box-sizing 初始为 border-box。

a:link 是未访问的链接；a:visited 是已访问的链接；a:hover 是鼠标悬停在链接上；a:active 是被选择的链接  
符合浏览器解析 CSS 所遵循的就近原则，将一般的放在上面，将特殊的放在下面。

固定顺序驴 lv 哈 ha

## img

HTML 中的标签分为闭合标签和自闭合标签。 自闭合标签有<input/><img/><br/><link/><hr/>等

</img> 因为 img 是自闭合标签

<meta>是不闭合标签

```
<meta http-equiv="expires" content="31 Dec 2008">
```

Img 的 alt 和 title 同时设置的时候，alt 作为图片的替代文字出现，title 是图片的解释文字

嵌入在 HTML 文档中的图像格式可以是 GIF、JPG、PNG；TIF 文件格式适用于在应用程序之间和计算机平台之间的交换文件，它的出现使得图像数据交换变得简单

## <table>标签中

bordercolor 表示边框颜色；

cellspacing 表示单元格之间的间隔；

cellpadding 表示单元格内边距。

设置边框宽度是 border 其中一个参数值

## <a>

target 的值：

- \_blank 在新窗口打开链接
  - \_self 在当前框架中打开链接
  - \_parent 在父框架打开链接
  - \_top 在当前窗口打开链接
  - framename 在指定框架打开链接
- <a href=URL target=\_blank>../</a>

## <input>

<input>元素的 type 属性的取值可以是 image、checkbox、button  
语法

```
<input type="value">
```

### 属性值

值	描述
button	定义可点击按钮（多数情况下，用于通过 JavaScript 启动脚本）。
checkbox	定义复选框。
file	定义输入字段和“浏览”按钮，供文件上传。
hidden	定义隐藏的输入字段。
image	定义图像形式的提交按钮。
password	定义密码字段。该字段中的字符被掩码。
radio	定义单选按钮。
reset	定义重置按钮。重置按钮会清除表单中的所有数据。
submit	定义提交按钮。提交按钮会把表单数据发送到服务器。
text	定义单行的输入字段，用户可在其中输入文本。默认宽度为 20 个字符。

同一个 name 的单选按钮都是同一组，同一组只有一个能被选中

## <label>

html5 规范不允许加 id，但是浏览器厂商却实现了这个功能，  
label 标签只有两个属性

for（规定 label 绑定到哪个表单元素。）

form（规定 label 字段所属的一个或多个表单）

<form>

<label for="male">Male</label>

<input type="radio" name="sex" id="male" />

<br />

<label for="female">Female</label>

<input type="radio" name="sex" id="female" />

</form>

## 14.发邮件

<a href="mailto:ming.zhou@nowcoder.com">发送邮件</a>

## 15.图片请求

基于以下 HTML 结构，判断浏览器会发送多少个图片请求？3 个

```

<head>
<style>
#bg {
background-url('/img/bg.png');
}
#test {
background-url('/img/test.png');
display: none;
}
</style>
</head>
<body>
<div id="test"></div>


</body>

```

15.1 display:none;和 visibility:hidden;的 img 也会发出请求。display:none;的 opera 除外。

15.2 对于重复的图片请求(包括背景)，所有浏览器都只产生一次请求。

15.3 背景仅在应用的元素在页面中存在时，才会产生请求。因此#bg 没有产生请求

15.4 Opera 和 Firefox 对于用 display: none 隐藏的元素背景，不会产生 HTTP 请求。

15.5 多重定义的背景图，一般浏览器都只产生后一次图片的请求。但是 webkit 引擎浏览器对背景图都请求，是因为支持 CSS3 中的多背景图。

15.6 触发 hover 时，才会请求 hover 状态下的背景。不触发的话，只请求默认的背景图片。

15.7 JS 里 innerHTML 中的图片，一般浏览器都会立即发送请求；只有当添加到 DOM 树上时，Opera 才会发送请求。

15.8 图片预加载也会发出请求

## 图片预加载

最常用的是 JS 方案：

**JavaScript Code** 复制内容到剪贴板

```

1. <script type="text/javascript">
2.     new Image().src = 'haorooms.jpg';
3.     new Image().src = 'http2.jpg';
4. </script>

```

在无 JS 支持的环境下，可以采用隐藏元素来预加载：

**CSS Code** 复制内容到剪贴板

```

1. 

```

## 16.DHTML

DHTML 将 HTML、JavaScript、DOM 以及 CSS 组合在一起，用于创造动态性更强的网页。

实现了网页从 Web 服务器下载后无需再经过服务的处理，而在浏览器中直接动态地更新网页的内容、排版样式和动画的功能。例如，当鼠标指针移到文章段落中时，段落能够变成蓝色，或者当鼠标指针移到一个超级链接上时，会自动生成一个下拉式子链接目录等。

包括：

①动态内容(Dynamic Content)：动态地更新网页内容，可“动态”地插入、修改或删除网页的元件，如文字、图像、标记等。

②动态排版样式(Dynamic Style Sheets)：W3C 的 CSS 样式表提供了设定 HTML 标记的字体大小、字形、样式、粗细、文字颜色、行高度、加底线或加中间横线、缩排、与边缘距离、靠左右或置中、背景图片或颜色等排版功能，而“动态排版样式”即可以“动态”地改变排版样式。

## 17. CSS 样式，下面哪一个能够用于边距的单位

em 的值并不是固定的，会继承父级元素的字体大小。px 为固定单位。pt 绝对长度单位，pt 的效果是不随浏览器分辨率的变化而发生视觉的变化

## 18.浏览器内核

一直到 IE9，都是 Trident 内核，前缀使用-ms  
firefox 是 Gecko 内核  
chrome 是 webkit 内核，前缀使用-Webkit

### 19.显示优先级：帧元素 表单元素 非表单元素

在 html 中，帧元素（frameset）的优先级最高，表单元素比非表单元素的优先级要高。

表单元素包括：文本输入框，密码输入框，单选框，复选框，文本输入域，列表框等等；

非表单元素包括：连接（a），div,table,span 等。

所有的 html 元素又可以根据其显示分成两类：有窗口元素以及无窗口元素。有窗口元素总是显示在无窗口元素的前面。

有窗口元素包括：select 元素，object 元素，以及 frames 元素等等。

无窗口元素：大部分 html 元素都是无窗口元素。

### 20.iframe：内联框架

iframe 可用在以下几个场景中：

- 1：典型系统结构，左侧是功能树，右侧就是一些常见的 table 或者表单之类的。为了每一个功能，单独分离出来，采用 iframe。
- 2：ajax 上传文件。
- 3：加载别的网站内容，例如 google 广告，网站流量分析。
- 4：在上传图片时，不用 flash 实现无刷新。
- 5：跨域访问的时候可以用到 iframe，使用 iframe 请求不同域名下的资源。