# **1.    HTML5推出的理由**

解决Web上存在的问题：

**Web浏览器间的兼容性低:**在一个浏览器中可以运行的HTML、Css、Javascript，在另一个浏览器中不能运行。

原因：各浏览器规范不统一，没有被标准化。

解决方案：使各浏览器的功能符合通用标准。

**文档结构不够明确**：HTML4中元素不能把文档结构表示清楚。

解决方案：增加与结构相关的元素。

**Web应用程序的功能受到限制**：HTMLL4对Web应用程序的贡献很小，比如：不允许同时上传多个文件。

解决方案：提供供Web应用程序使用的API。

**世界知名浏览器厂商对HTML5的支持**

* 微软的IE<分两个阶段IE9以上支持HTML5>
* Google的Chrome
* 苹果的Safari
* Opera
* Mozilla Firefox

扩展性上：增加一系列的API支持（Canvas、本地存储、文件操作、重力感应等），非常的强大

# **2.    HTML5语法的改变**

**内容类型不变**

HTML5的文件扩展符（html或.htm）与内容类型（text/html）保持不变。

**DOCTYPE声明变化**

HTML4中需要指明是HTML的哪个版本，HTML5不需要，只使用<!DOCTYPE html>即可。

HTML4的DOCTYPE声明方式:

|  |
| --- |
| <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  "http://www.w3.org/TR/html4/loose.dtd">  <html>  <head>  <title>HTML4的DOCTYPE声明</title>  </head>  <body>  </body>  </html> |

HTML5 的DOCTYPE声明方式

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <title>HTML5的DOCTYPE声明</title>  </head>  <body>  </body>  </html> |

**指定字符编码变化**

HTML4：<meta http-equiv=‶content-type″ content=‶text/html; charset=UTF-8″>

HTML5:<meta charset=‶UTF-8″>

|  |
| --- |
| <!--示例代码-->  <meta charset="UTF-8"> |

**可以省略元素的标记**

HTML5中很多元素标记可以省略

**具有boolean值的属性调整**

不指定属性值、属性名设定为属性值、字符串设为空时表示属性值为true；

不写该属性表示属性值为false。

例如：

|  |
| --- |
| <!--示例代码-->  <input type = "checkbox" checked=""> <!--true-->  <input type = "checkbox" checked="checked"> <!--true-->  <input type = "checkbox" checked> <!--true-->  <input type = "checkbox" > <!--false--> |

**可省略引号**

HTML5可省略指定属性值时的引号。

|  |
| --- |
| <!--示例代码-->  <input type = "checkbox" checked=checked> |

# **3.    新增的元素和废除的元素**

## **新增元素**

**新增的结构元素**

**section**：表示页面中内容块，比如章节、页眉、页脚或页面中的其他部分，可与<h1>到<h6>结合使用表示文档结构。

**article**:表示页面中一块与上下文不相关的独立内容，比如博客中的一篇文章或报纸中的一篇文章。

**aside**:表示article内容之外，与article内容相关的辅助信息。

**header**：表示页面中的区域块，通常用它表示标题。

**hgroup**：用于对整个页面或页面中标题进行整合。

**footer**：表示页面中的区域块，通常表示区域快的脚部或底部，用于承载作者姓名、创作日期等与作者的元素。

**nav**:表示页面中导航部分。

**figure**：表示一段独立的流内容，一般表示主体流内容的一个独立单元。

**新增的其他元素**

**video**:定义电影片段、视频流等视频。

**audio**：定义音乐或音频流。

**canvas**：画布，本身没有行为，仅提供一块画布，但它的API展现给JavaScript及脚本，能够把想绘制的东西绘制在canvas上。

**embed mark progress meter time ruby rt rp wbr command details detalist**

**datagrid keygen output source menu**

**新增的input元素的类型**

**email**：表示必须输的email地址

**url**：表示文本框输入的一个地址

**number**：表示数字

**range**：表示数字范围值

**DataPickers**：表示日历的日期、时间

## **废除的元素**

**能使用css代替的元素**

basefont big center font s tt u等

**不再使用frame框架**

由于frame框架对网页可用性存在负面影响，HTML5中已不支持frame框架，只支持iframe框架或者用服务器方式创建的由多个页面组成的复合页面的形式，同时将frameset元素、frame元素、noframes元素废除。

**只有部分浏览器支持的元素**

**其他被废除的元素**

# **4.    新增的属性和废除的属性**

* 能被CSS样式表替代的属性全部废除掉；
* 多余属性，例如：target、profile、version等被废除掉；

**新增的属性**

**表单相关的属性**

**链接相关的属性**

**其他属性**

**废除的属性**

# **5.    全局属性**

HTML5中新增全局属性的概念，全局属性指可以对任何元素都使用的属性。

**contentEditable属性**

允许用户编辑元素中内容，使用该属性的元素必须为可以获得鼠标焦点的元素，而且在点击鼠标后向用户提供一个插入符号，提示用户该元素允许进行编辑。

是boolean值类型，可以设为true、false或继承状态。其中，true代表可编辑，false代表不可编辑，当未指定true或false时与父元素的继承状态相同。

|  |
| --- |
| <!-- 示例代码 -->  <h2>可编辑列表</h2>  <ul contenteditable="">  <li>列表1</li>  <li>列表2</li>  <li>列表3</li>  </ul> |

**designMode属性**

用来指定整个页面是否可编辑，当页面可编辑时，页面中所有支持contentEditable属性的元素都变为可编辑状况。designMode属性只能在JavaScript脚本中被修改、编辑。属性值可取on（可编辑）或off（不可编辑）。

**hidden属性**

HTML5中所有元素都允许使用hidden属性，该属性类似于input元素中hidden元素，boolean值，可设为true（不可见）、false（可见）。当某元素的hidden属性值为true时，浏览器不渲染该元素，使该元素处于不可见状态，但浏览器创建该元素内容，即页面加载后允许使用JavaScript脚本将该属性值取消，使该元素可见。

|  |
| --- |
| <!--示例代码-->  <h2>可编辑列表</h2>  <!-- 运行结果可以发现ul元素被隐藏了 -->  <ul hidden="">  <li>列表1</li>  <li>列表2</li>  <li>列表3</li>  </ul> |

**spellcheck属性**

针对input（type=text）与textarea这两个文本输入框提供的一个新属性，主要对用户输入内容进行拼写与语法检查。属性值为boolean值，可取true或false。

|  |
| --- |
| <!--示例代码-->  <input type="text" spellcheck=""> |

**tableindex属性**

当点击Tab键时，让窗口或页面中可获得焦点的链接元素或表单元素进行遍历，tableindex表示该元素第几个被访问到。

若tableindex值为"-1"时表示无法获取该元素.

|  |
| --- |
| <!--示例代码-->  <a tabindex="1">I</a>  <a tabindex="3">need</a>  <a tabindex="2">you</a>  <!-- 运行以上代码,摁下tab键，会依次选中 I you need -->  <ul tabindex = "-1">  <li>1</li>  <li>2</li>  <li>3</li>  </ul>  <!--上边的代码可以在程序中获取到tabindex，但是运行中不能通过tab键获取焦点--> |