

# PHP Tutorial - HTML & HTML5

领航教育 IT HTML posts

## HTML

### 介绍

HTML is a markup language for describing web documents (web pages).

HTML stands for Hyper Text Markup Language

A markup language is a set of markup tags HTML documents are described by HTML tags

Each HTML tag describes different document content

-- [W3CSchools](#)

HTML(超文本标记语言)

- 扩展名是 .html
- 不是编程语言!!!
- 如果一行出错 浏览器会直接忽略
- debug会比较麻烦 需要逐行检查 (右键页面 选择检查元素)
- 多平台兼容性好
- 首页通常命名为 index.html

### 编辑器

Window	Mac
<a href="#">Notepad</a>	<a href="#">Coda2</a>
<a href="#">Sublime</a>	<a href="#">Sublime</a>

### 格式

为提升服务稳定性，马克飞象将于10月7日正式迁移至全新域名 [maxiang.io](#)。 ×  
请尽快将当前域名下的本地内容同步至Evernote，然后开始使用[maxiang.io](#)。

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>

<tagname>content</tagname>
```

- 一开一闭: 所有的tagname都必须要有开有闭
- 所有的HTML页面都必须包括
  - `<!doctype html></html>` 声明HTML的版本
  - `<head></head>` 网页的头文件 之间所有的内容不会显示在网页的页面上 包含关键字, 标题, 脚本...
    - `<title></title>` 网页的Title 会显示在浏览器的标签页上
  - `<body></body>` 网页的正文内容

## 属性

- HTML的元素可以有属性
- 属性由名字和值组成: `name="value"`

## lang语言属性

页面的语言可以用 `<html lang="en-US">` 来修改 用来便于搜索引擎的抓取

```
<!DOCTYPE html>
<html lang="en-US">
<body>

    <h1>My First Heading</h1>
    <p>My first paragraph.</p>

</body>
</html>
```

## title标题属性

段落的标题可以用 `title=" "` 属性来直接生成

```
<p title="this is title">
    this is paragraph.
</p>
```

相比较与用 `<h1></h1>` 等方法, title更加方便, 但是缺点是没办法改变样式

## href链接属性

`href` 链接属性通常位于 `<a></a>` 里面, 用来添加网页的链接

```
<a href="http://www.google.com">Google</a>
```

## alt属性

`alt`属性通常位于 `<img></img>` 里面, 用来代替图片加载不出的情况, 同时方便搜索引擎抓取图片

```

```

# 元素

## Heading

```
<h1>This is a huge header.</h1>
<h2>This one's a little smaller.</h2>
<h3>And smaller still...</h3>
<h4>And so on</h4>
<h5>And on</h5>
<h6>Do you really want to go smaller than this?</h6>
```

**This is a huge header.**

**This one's a little smaller.**

**And smaller still...**

**And so on**

**And on**

**Do you really want to go smaller than this?**

**hr 水平线**

```
<p>This is a paragraph.</p>
<hr>
<p>This is a paragraph.</p>
<hr>
<p>This is a paragraph.</p>
```

This is a paragraph.

---

This is a paragraph.

---

This is a paragraph.

## meta

- 用于SEO搜索引擎的优化
  - 搜索的标题

```
<meta name="title" content="领航培训 | IT就业培训"  
itemprop="name">
```
  - 搜索的描述

```
<meta name="description" content="领航就业培训，贴心为你量身  
打造，挑战7万年薪，成就巅峰人生." />
```
  - 搜索的关键字

```
<meta name="keywords" content="领航，教育，墨尔本，就业，培  
训，iOS，Swift，PHP，SQL，IT，工作，墨尔本IT职业，墨尔本IT工  
作，墨尔本IT培训，领航职业培训，领航工作培训，领航IT就业，领航iOS  
就业班，领航PHP就业班，" />
```
  - 搜索的地理位置

```
<meta property="og:locale" content="zh_CN">
```

## Paragraphs

```
<p>This is a paragraph</p>
```

This is a paragraph

## Line Breaks <br>

```
<p>This is<br>a para<br>graph with line breaks</p>
```

This is  
a para  
graph with line breaks

## Preformatted Text

```
<pre>
    My Bonnie lies over the ocean.

    My Bonnie lies over the sea.

    My Bonnie lies over the ocean.

    Oh, bring back my Bonnie to me.
</pre>
```

My Bonnie lies over the ocean.

My Bonnie lies over the sea.

My Bonnie lies over the ocean.

Oh, bring back my Bonnie to me.

## Hyperlinks超链接

```
<a href="http://www.google.com" target="_blank">Google</a>
```

Google

- href分为绝对路径和相对路径
  - 绝对路径:
    - 无论网页放在那里 都通过绝对路径找到文件
    - 网页链接(www.google.com)都属于绝对路径
  - 相对路径:
    - 相对于网页的位置开始查找的路径 (css/main.css)
    - 需要链接的文件和网页放在同一个文件夹下 否则容易找不到文件
- **target="\_blank"** 用来在新窗口或者新tab中打开链接
- 可以把图片转化为超链接

```
<a href="google.com">
    
</a>
```

## Image

```

```

## Table

表格是以 `<table></table>` 来声明 包括了

- `th` 标题
- `tr` 行
- `td` 单元格

由于邮件需要适应不同的屏幕尺寸, 而且用HTML写出来的邮件没有办法加在 javascript等脚本 所以table可以让HTML Email完成自适应屏幕的最普遍的方法 —— [HTML Email](#)

```
<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Points</th>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
```

Firstname	Lastname	Points
Eve	Jackson	94

如果一个单元格占用了几列, 可以用 `colspan= " "` 来实现

```

<table style="width:100%">
  <tr>
    <th>Name</th>
    <th colspan="2">Telephone</th>
  </tr>
  <tr>
    <td>Bill Gates</td>
    <td>555 77 854</td>
    <td>555 77 855</td>
  </tr>
</table>

```

Name	Telephone	
Bill Gates	555 77 854	555 77 855

如果一个单元格要占用几行, 可以用 `rowspan=" "` 来实现

```

<table style="width:100%">
  <tr>
    <th>Name:</th>
    <td>Bill Gates</td>
  </tr>
  <tr>
    <th rowspan="2">Telephone:</th>
    <td>555 77 854</td>
  </tr>
  <tr>
    <td>555 77 855</td>
  </tr>
</table>

```

<b>Name:</b>	Bill Gates
<b>Telephone:</b>	555 77 854

table的标题可以用 `<caption>` 实现

```
<caption>Monthly savings</caption>
```

## Lists

列表分为有序的和无序的:

- 有序列表

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

1. Coffee
2. Tea
3. Milk

有序列表的标示符可以用 `type=" "` 来修改

类型	说明
<code>type="1"</code>	数字
<code>type="A"</code>	大写字母
<code>type="a"</code>	小写字母
<code>type="I"</code>	大写罗马数字
<code>type="i"</code>	小写罗马数字

- 无序列表

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

- Coffee
- Tea
- Milk

无序列表有不同的标识符 可以用 `style="list-style-type: "` 来实现:

```
<ul style="list-style-type:disc">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

- Coffee
- Tea
- Milk

```
<ul style="list-style-type:circle">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

- Coffee
- Tea
- Milk

```
<ul style="list-style-type:square">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

- Coffee
- Tea
- Milk

```
<ul style="list-style-type:none">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

Coffee

Tea

Milk

## Block-level

- <div>
- <h1>-<h6>
- <p>
- <form>

```
<div style="background-color:black; color:white; padding:20px;">

<h2>London</h2>
<p>London is the capital city of England. It is the most populous
city in the United Kingdom, with a metropolitan area of over 13
million inhabitants.</p>
</div>
```

# London

London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.

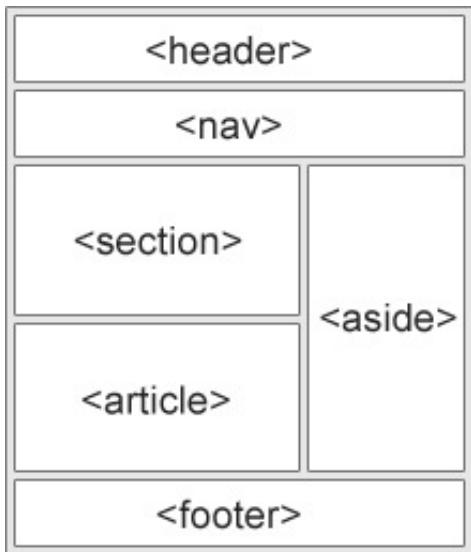
## Inline

- <span>
- <a>
- <img>

```
<h1>My <span style="color:red">Important</span> Heading</h1>
```

# My Important Heading

## Layout



类型
header
nav
section
article
aside
footer
details
summary

```
<!DOCTYPE html>
<html>
<head>
<style>
#header {
    background-color:black;
    color:white;
    text-align:center;
    padding:5px;
}
#nav {
    line-height:30px;
    background-color:#eeeeee;
    height:300px;
    width:100px;
    float:left;
    padding:5px;
}
#section {
    width:350px;
    float:left;
    padding:10px;
}
#footer {
    background-color:black;
    color:white;
    clear:both;
    text-align:center;
    padding:5px;
}
</style>
</head>
<body>

<div id="header">
<h1>City Gallery</h1>
</div>

<div id="nav">
London<br>
Paris<br>
Tokyo<br>
</div>

<div id="section">
<h2>London</h2>
<p>
London is the capital city of England. It is the most populous

```

```
city in the United Kingdom,  
with a metropolitan area of over 13 million inhabitants.  
</p>  
<p>  
Standing on the River Thames, London has been a major settlement  
for two millennia,  
its history going back to its founding by the Romans, who named it  
Londinium.  
</p>  
</div>  
  
<div id="footer">  
Copyright Navigation International  
</div>  
  
</body>  
</html>
```

## iframe

如果要在网页中嵌入别的网页, 可以使用 `<iframe src="URL"></iframe>` 来实现

```
<!DOCTYPE html>  
<html>  
<body>  
  
<iframe width="100%" height="300px" src="demo_iframe.htm"  
name="iframe_a"></iframe>  
<p><a href="http://www.google.com"  
target="iframe_a">google.com</a></p>  
  
<p>When the target of a link matches the name of an iframe, the  
link will open in the iframe.</p>  
  
</body>  
</html>
```

## 响应式布局

由于现在电脑, 手机, 平板的尺寸多样化, 所以现在网页编程最主要的最基本的核心就是响应式布局, 目的是可以让网页自动识别用户的屏幕尺寸, 然后相应的完成格式的改变 —— [自适应网页设计](#)

```
<!DOCTYPE html>
<html lang="en-us">
<head>
<style>
.city {
    float: left;
    margin: 5px;
    padding: 15px;
    width: 300px;
    height: 300px;
    border: 1px solid black;
}
</style>
</head>
<body>

<h1>Responsive Web Design Demo</h1>

<div class="city">
<h2>London</h2>
<p>London is the capital city of England.</p>
<p>It is the most populous city in the United Kingdom,
with a metropolitan area of over 13 million inhabitants.</p>
</div>

<div class="city">
<h2>Paris</h2>
<p>Paris is the capital of France.</p>
<p>The Paris area is one of the largest population centers in
Europe,
with more than 12 million inhabitants.</p>
</div>

<div class="city">
<h2>Tokyo</h2>
<p>Tokyo is the capital of Japan.</p>
<p>It is the center of the Greater Tokyo Area,
and the most populous metropolitan area in the world.</p>
</div>

</body>
</html>
```

## Color

在HTML中, 开发者可以用颜色的名字( `Red` ), 十六进制编码( `#dbb76c` ) 或者 RGB编码( `rgb(255,0,0)` )来选择颜色 —— [w3schools color](#)

## Safe Colors

Safe Colors是指在多年前计算机只支持256色的时候 所选出来的40个固定的系统颜色, 然而随着计算机的发展, 支持的颜色越来越多, 这概念现在已经不重要了.

## 特殊字符

由于编程的原因, 很多特殊字符没办法直接显示, 需要使用特别的编码来显示这些特殊字符

显示效果	名字	编码名字	编码数字
	空格	&nbsp	&#160
<	小于	&lt	&#60
>	大于	&gt	&#62
&	cent	&amp	&#38
¢	cent	&cent	&#162
£	pound	&pound	&#163
¥	yen	&yen	&#165
€	euro	&euro	&#8364
©	copyright	&copy	&#169
®	registered trademark	&reg	&#174

## XHTML

XHTML(可扩展超文本标记语言), 与HTML很类似, 但是更加严格. 结合了部分XML的功能和大多数HTML的简单特性.

XHTML与HTML的不同:

- XHTML元素必须正确的嵌套  
错误的: `<b><i>This text is bold and italic</b></i>`
- XHTML必须被关闭  
`<br /><hr />`
- 标签名必须小写
- XHTML文档必须有根元素

```
<html>
<head> ...
<body> ...
</html>
```

用[W3C Validator](#)来验证XHTML

## HTML Forms

### <form>

```
<form>
  .
  form elements
  .
</form>
```

### Text input

```
<form>
  First name:<br>
  <input type="text" name="firstname">
  <br>
  Last name:<br>
  <input type="text" name="lastname">
</form>
```

### Radio Button input

Radio button 只能单选

```
<form>
  <input type="radio" name="sex" value="male" checked>Male
  <br>
  <input type="radio" name="sex" value="female">Female
</form>
```

## Submit Button

```
<form action="action_page.php">
  First name:<br>
  <input type="text" name="firstname" value="Mickey">
  <br>
  Last name:<br>
  <input type="text" name="lastname" value="Mouse">
  <br><br>
  <input type="submit" value="Submit">
</form>
```

## Action Attribute & Method

`<form action="action_page.php" method = " ">` 定义了当表格submit后要做的事情, method有 Get 和 Post 两种方法 具体的会在PHP章节中详细解释

## Name & `<fieldset>`

如果表单要正确上传, input必须要有name

```
<form action="action_page.php">
  <fieldset>
    <legend>Personal information:</legend>
    First name:<br>
    <input type="text" name="firstname" value="Mickey">
    <br>
    Last name:<br>
    <input type="text" name="lastname" value="Mouse">
    <br><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```

## Drop-Down List

```
<select name="cars">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

## Text Area

```
<textarea name="message" rows="10" cols="30">
The cat was playing in the garden.
</textarea>
```

## Button

```
<button type="button" onclick="alert('Hello World!')">Click Me!
</button>
```

## Datalist

当用户输入的时候会有提示已经存入Datalist的数据

- input的list属性需要和datalist的id相匹配

```
<form action="action_page.php">
  <input list="browsers">
  <datalist id="browsers">
    <option value="Internet Explorer">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
  </datalist>
</form>
```

## <output>

<output> 显示计算的结果

```
<form action="action_page.asp"
  oninput="x.value=parseInt(a.value)+parseInt(b.value)">
  0
  <input type="range" id="a" name="a" value="50">
  100 +
  <input type="number" id="b" name="b" value="50">
  =
  <output name="x" for="a b"></output>
  <br><br>
  <input type="submit">
</form>
```

## Input Type

开发者可以根据修改输入框的不同属性, 比如text, password...

HTML5中新加了很多Type:

- color
- date
- datetime
- datetime-local
- email
- month
- number
- range
- search
- tel
- time
- url
- week

```
<form>
User name:<br>
<input type="text" name="username">
<br>
User password:<br>
<input type="password" name="psw">
</form>
```

## value

value = “” 是给输入框添加一个初始值

```
<form action="">
First name:<br>
<input type="text" name="firstname" value="John">
<br>
Last name:<br>
<input type="text" name="lastname">
</form>
```

## read only

被设为read only的input field不能被修改

```
<form action="">
First name:<br>
<input type="text" name="firstname" value="John" readonly>
<br>
Last name:<br>
<input type="text" name="lastname">
</form>
```

## disable

disable表示这个input file 不能使用, 不能被点击 同时不会被提交

```
<form action="">
First name:<br>
<input type="text" name="firstname" value="John" disabled>
<br>
Last name:<br>
<input type="text" name="lastname">
</form>
```

## size & maxlength

size表示的是输入框的大小

maxlength限定的最大输入字节数

```
<form action="">
First name:<br>
<input type="text" name="firstname" value="John" size="40">
<br>
Last name:<br>
<input type="text" name="lastname" maxlength="10">
</form>
```

# HTML5

- HTML5声明  
`<!DOCTYPE html>`
- character encoding  
`<meta charset="UTF-8">`

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Title of the document</title>
</head>

<body>
Content of the document.....
</body>

</html>
```

## Canvas

画布Canvas是HTML5新加的元素  
可以在上面用Javascript进行画图

```
<canvas id="myCanvas" width="200" height="100" style="border:1px
solid #000000;">
</canvas>
```

## SVG

SVG的全称是可伸缩向量图形(Salable Vector Graphics)  
SVG用来给网页定义一个图形

### SVG Circle

```
<!DOCTYPE html>
<html>
<body>

<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4"
  fill="yellow" />
</svg>

</body>
</html>
```

## SVG Rectangle

```
<svg width="400" height="100">
  <rect width="400" height="100" style="fill:rgb(0,0,255);stroke-
width:10;stroke:rgb(0,0,0)" />
</svg>
```

## SVG Rounded Rectangle

```
<svg width="400" height="180">
  <rect x="50" y="20" rx="20" ry="20" width="150" height="150"
  style="fill:red;stroke:black;stroke-width:5;opacity:0.5" />
</svg>
```

## SVG Star

```
<svg width="300" height="200">
  <polygon points="100,10 40,198 190,78 10,78 160,198"
  style="fill:lime;stroke:purple;stroke-width:5;fill-
rule:evenodd;" />
</svg>
```

## SVG Logo

```
<svg height="130" width="500">
  <defs>
    <linearGradient id="grad1" x1="0%" y1="0%" x2="100%" y2="0%">
      <stop offset="0%" style="stop-color:rgb(255,255,0);stop-
      opacity:1" />
      <stop offset="100%" style="stop-color:rgb(255,0,0);stop-
      opacity:1" />
    </linearGradient>
  </defs>
  <ellipse cx="100" cy="70" rx="85" ry="55" fill="url(#grad1)" />
  <text fill="#ffffff" font-size="45" font-family="Verdana" x="50" y="86">SVG</text>
  Sorry, your browser does not support inline SVG.
</svg>
```

## SVG与Canvas的区别

- SVG是基于XML来绘制2D图形而Canvas是用Javascript来绘制2D图形
- SVG中所有的图形是基于Object, 所以如果SVG的object被修改了, 那么浏览器会自动刷新图形  
Canvas是基于pixel来绘制图形的. 一旦图形绘制完毕, 浏览器就不会继续审核这个图形, 所以如果需要重画必须刷新浏览器

## HTML5 Media

### Video

HTML5中新加入了 `<video></video>` 来播放视频, 支持的格式有MP4, WebM, Ogg  
`autoplay` 可以让视频自动播放

```
<video width="320" height="240" controls autoplay>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
Your browser does not support the video tag.
</video>
```

### Youtube

在HTML5中可以通过Youtube的ID来插入视频

```
<iframe width="420" height="315">
  <!-- http://www.youtube.com/embed/XGSy3_Czz8k?autoplay=1 -->
</iframe>
```

## Audio

HTML5中加入了 `<audio></audio>` 来播放音频  
支持的格式有MP3, Ogg, Wav

```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
Your browser does not support the audio element.
</audio>
```

## HTML5 Plugin

在HTML5中, 开发者可以用 `<object></object>` 或者 `<embed></embed>` 来嵌入插件

```
<object width="100%" height="500px" data="snippet.html"></object>
<embed width="100%" height="500px" src="snippet.html">
```

## 习题(涵盖面试题目)

### 理论问题

- Doctype作用？严格模式与混杂模式如何区分？它们有何意义？
- HTML5 为什么只需要写？

- 行内元素有哪些？块级元素有哪些？空(void)元素有那些？
- 页面导入样式时，使用link和@import有什么区别？
- CSS选择符有哪些？哪些属性可以继承？
- CSS优先级算法如何计算？
- CSS3新增伪类有那些？
- 如何居中div？如何居中一个浮动元素？
- 常见的浏览器内核有哪些？
- html5有哪些新特性、移除了那些元素？如何处理HTML5新标签的浏览器兼容问题？如何区分 HTML 和 HTML5？
- 简述一下你对HTML语义化的理解？
- (写)描述一段语义的html代码吧。
- HTML5的离线储存怎么用，工作原理能不能解释一下？
- 浏览器是怎么对HTML5的离线储存资源进行管理和加载的呢？
- 请描述一下 cookies, sessionStorage 和 localStorage 的区别？
- iframe有那些缺点？
- Label的作用是什么？是怎么用的？（加 for）
- HTML5的form如何关闭自动完成功能？给不想要提示的input是设置 autocomplete=off即可
- 如何实现浏览器内多个标签页之间的通信？(阿里)
- 如何使用websocket？如何兼容低浏览器？(阿里)
- 页面可见性（Page Visibility）API 可以有哪些用途？

---

## 实战演练

- [HTML Element Task](#)

要求: 学生各自搭建最基础的HTML页面, 熟悉所有的元素(table, list, video, iframe, div, 链接和页内跳转)

p.s. 点击标题的链接可以进入Github Classroom, 上传作业

- [HTML Quiz](#)

要求: 学生能在不查阅资料的情况下独立完成Quiz.

## 答案

- Doctype作用? 标准模式与兼容模式各有什么区别?

(1) 、<!DOCTYPE>声明位于位于HTML文档中的第一行, 处于 <html> 标签之前。告知浏览器的解析器用什么文档标准解析这个文档。DOCTYPE不存在或格式不正确会导致文档以兼容模式呈现。

(2) 、标准模式的排版 和JS运作模式都是以该浏览器支持的最高标准运行。在兼容模式中, 页面以宽松的向后兼容的方式显示, 模拟老式浏览器的行为以防止站点无法工作。

- HTML5 为什么只需要写 ?

**HTML5** 不基于 SGML, 因此不需要对DTD进行引用, 但是需要doctype来规范浏览器的行为 (让浏览器按照它们应该的方式来运行) ;

而**HTML4.01**基于SGML, 所以需要对DTD进行引用, 才能告知浏览器文档所使用的文档类型。

- 行内元素有哪些? 块级元素有哪些? 空(void)元素有那些?

首先：CSS规范规定，每个元素都有`display`属性，确定该元素的类型，每个元素都有默认的`display`值，如`div`的`display`默认值为“block”，则为“块级”元素；`span`默认`display`属性值为“inline”，是“行内”元素。

- (1) 行内元素有：`a b span img input select strong`（强调的语气）
- (2) 块级元素有：`div ul ol li dl dt dd h1 h2 h3 h4...p`

- (3) 常见的空元素：

`<br> <hr> <img> <input> <link> <meta>`

鲜为人知的是：

`<area> <base> <col> <command> <embed> <keygen> <param>`  
`<source> <track> <wbr>`

- 页面导入样式时，使用`link`和`@import`有什么区别？

(1) `link`属于XHTML标签，除了加载CSS外，还能用于定义RSS，定义`rel`连接属性等作用；而`@import`是CSS提供的，只能用于加载CSS；

(2) 页面被加载的时，`link`会同时被加载，而`@import`引用的CSS会等到页面被加载完再加载；

(3) `import`是CSS2.1 提出的，只在IE5以上才能被识别，而`link`是XHTML标签，无兼容问题；

- 浏览器的内核分别是什么？

\* IE浏览器的内核Trident、Mozilla的Gecko、Chrome的Blink (WebKit的分支) 、Opera内核原为Presto，现为Blink；

- 常见兼容性问题？

- \* png24位的图片在IE6浏览器上出现背景，解决方案是做成PNG8.
- \* 浏览器默认的margin和padding不同。解决方案是加一个全局的\*  
`{margin:0; padding:0;}`来统一。
- \* IE6双边距bug:块属性标签float后，又有横行的margin情况下，在ie6显示margin比设置的大。  
`浮动ie产生的双倍距离 #box{ float:left; width:10px; margin:0 0 0 100px; }`

这种情况之下IE会产生20px的距离，解决方案是在float的标签样式控制中加入  
`--_display:inline;`将其转化为行内属性。（\_这个符号只有ie6会识别）

渐进识别的方式，从总体中逐渐排除局部。

首先，巧妙的使用“\9”这一标记，将IE浏览器从所有情况中分离出来。  
接着，再次使用“+”将IE8和IE7、IE6分离开来，这样IE8已经独立识别。

#### css

```
.bb{
  background-color:#f1ee18; /*所有识别*/
  .background-color:#00deff\9; /*IE6、7、8识别*/
  +background-color:#a200ff; /*IE6、7识别*/
  _background-color:#1e0bd1; /*IE6识别*/
}
```

- \* IE下，可以使用获取常规属性的方法来获取自定义属性，  
也可以使用`getAttribute()`获取自定义属性；  
Firefox下，只能使用`getAttribute()`获取自定义属性。  
解决方法：统一通过`getAttribute()`获取自定义属性。
- \* IE下，even对象有x,y属性，但是没有pageX,pageY属性；  
Firefox下，event对象有pageX,pageY属性，但是没有x,y属性。
- \* 解决方法：（条件注释）缺点是在IE浏览器下可能会增加额外的HTTP请求数。
- \* Chrome 中文界面下默认会将小于 12px 的文本强制按照 12px 显示，  
可通过加入 CSS 属性 `-webkit-text-size-adjust: none;` 解决。

超链接访问过后hover样式就不出现了 被点击访问过的超链接样式不在具有hover 和active了解决方法是改变CSS属性的排列顺序：

L-V-H-A : `a:link {} a:visited {} a:hover {} a:active {}`

- html5有哪些新特性、移除了那些元素？如何处理HTML5新标签的浏览器兼容问题？  
如何区分 HTML 和  
HTML5？

\* HTML5 现在已经不是 SGML 的子集，主要是关于图像，位置，存储，多任务等功能的增加。

\* 绘画 canvas

用于媒介回放的 video 和 audio 元素

本地离线存储 localStorage 长期存储数据，浏览器关闭后数据不丢失；  
sessionStorage 的数据在浏览器关闭后自动删除

语义化更好的内容元素，比如 article、footer、header、nav、section  
表单控件，calendar、date、time、email、url、search  
新的技术webworker，websocket，Geolocation

\* 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;

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，

浏览器支持新标签后，还需要添加标签默认的样式：

\* 当然最好的方式是直接使用成熟的框架、使用最多的是html5shim框架

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

如何区分： DOCTYPE声明\新增的结构元素\功能元素

- 语义化的理解？

用正确的标签做正确的事情！

html语义化就是让页面的内容结构化，便于对浏览器、搜索引擎解析；  
在没有样式CCS情况下也以一种文档格式显示，并且是容易阅读的。  
搜索引擎的爬虫依赖于标记来确定上下文和各个关键字的权重，利于 SEO。  
使阅读源代码的人对网站更容易将网站分块，便于阅读维护理解。

- HTML5的离线储存？

`localStorage` 长期存储数据，浏览器关闭后数据不丢失；  
`sessionStorage` 数据在浏览器关闭后自动删除。

- (写)描述一段语义的html代码吧。

(HTML5中新增加的很多标签（如：`<article>`、`<nav>`、`<header>`和`<footer>`等）

就是基于语义化设计原则)

```
< div id="header">
  < h1>标题< /h1>
  < h2>专注Web前端技术< /h2>
< /div>
```

- iframe有那些缺点？

\* iframe会阻塞主页面的Onload事件；

\* iframe和主页面共享连接池，而浏览器对相同域的连接有限制，所以会影响页面的并行加载。

使用iframe之前需要考虑这两个缺点。如果需要使用iframe，最好是通过 javascript

动态给iframe添加src属性值，这样可以绕开以上两个问题。

- HTML5的form如何关闭自动完成功能？

给不想要提示的 form 或下某个input 设置为 autocomplete=off。

- 请描述一下 cookies, sessionStorage 和 localStorage 的区别?

cookie在浏览器和服务器间来回传递。 sessionStorage和localStorage不会  
sessionStorage和localStorage的存储空间更大；  
sessionStorage和localStorage有更多丰富易用的接口；  
sessionStorage和localStorage各自独立的存储空间；

- 如何实现浏览器内多个标签页之间的通信? (阿里)

调用localStorage、cookies等本地存储方式

- webSocket如何兼容低浏览器? (阿里)

Adobe Flash Socket 、 ActiveX HTMLFile (IE) 、 基于 multipart  
编码发送 XHR 、 基于长轮询的 XHR

---

Author: Shaowei Zhang