HTML 面试题汇总

1. 什么是 <!DOCTYPE>？是否需要在 *HTML5* 中使用?

> 参考答案:
>
> 它是 *HTML* 的文档声明，通过它告诉浏览器，使用哪一个 *HTML* 版本标准解析文档。
>
> 在浏览器发展的历史中，*HTML* 出现过很多个版本，不同版本在元素、属性等书写格式上略有差异，如果不预先告诉浏览器，浏览器就不知道我们的文档标准是什么，在这种情况下，大部分浏览器将开启最大兼容模式来解析网页，我们称之为怪异模式。这不仅会降低解析效率，而且会在解析过程中产生一些难以预知的 *bug*，所以文档声明是必须的。
>
> 而文档声明有多种书写格式，对应不同的 *HTML* 版本，<!DOCTYPE> 这种书写是告诉浏览器，整个文档使用 *HTML5* 的标准进行解析。

2. 什么是可替换元素，什么是非可替换元素，它们各自有什么特点?

> 参考答案:
>
> 可替换元素是指这样一种元素，它在页面中的大部分展现效果不由 *CSS* 决定。
>
> 比如 *img* 元素就是一个可替换元素，它在页面中显示出的效果主要取决于你连接的是什么图片，图片是什么它就展示什么，*CSS* 虽然可以控制图片的尺寸位置，但永远无法控制图片本身。
>
> 再比如，*select* 元素也是一个典型的可替换元素，它在页面上呈现的是用户操作系统上的下拉列表样式，因此，它的展现效果是由操作系统决定的。所以，同一个 *select* 元素，放到不同操作系统的电脑上会呈现不同的外观。
>
> *img*、*video*、*audio*、大部分表单元素都属于可替换元素。
>
> 非可替换元素就是指的普通元素，它具体在页面上呈现什么，完全由 *CSS* 来决定。

3. *src* 和 *href* 的区别（京东）

> 参考答案:
>
> 它们都是 *HTML* 中特定元素的属性。
>
> *src* 是 *source* 的缩写，它通常用于 *img*、*video*、*audio*、*script* 元素，通过 *src* 属性，可以指定外部资源的来源地址。

> *href* 是 *hyper reference* 的缩写，意味「超引用」，它通常用于 *a*、*link* 元素，通过 *href* 属性，可以标识文档中引用的其他超文本。

4. 说说常用的 *meta* 标签

> 参考答案：
>
> *meta* 标签提供关于HTML文档的元数据。元数据不会显示在页面上，但是对于机器是可读的。它可用于浏览器（如何显示内容或重新加载页面），搜索引擎（关键词），或其他 *web* 服务。
>
> 常用的 *meta* 标签的属性有：
>
> 1. *content* ，设置或返回 *meta* 元素的 *content* 属性的值 。
> 2. *http-equiv*，把 *content* 属性连接到一个 *HTTP* 头部。
> 3. *name*，把 *content* 属性连接到某个名称。
>
> 关于 *meta* 标签，回答出常用的一些属性值即可，不用全部背下来。
>
> 具体的关于 *meta* 标签能够设置的属性，可以参阅：*https://www.runoob.com/w3cnote/meta.html*

5. 说说对 *html* 语义化的理解

> 参考答案：
>
> 1. 去掉或者丢失样式的时候能够让页面呈现出清晰的结构
> 2. 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；
> 3. 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；
> 4. 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。 *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）下面就是语义化 *html* 代码；
>
> ```
> 
>    <h1>标题</h1>
>    <h2>专注Web前端技术</h2>
> 
> ```
>
> 总结一下，总之就是：
>
> - 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。

6. *label* 的作用是什么？是怎么用的?

参考答案:

label 元素不会向用户呈现任何特殊效果。

不过，它为鼠标用户改进了可用性。

如果您在 *label* 元素内点击文本，就会触发此控件。

就是说，当用户选择该标签时，浏览器就会自动将焦点转到和标签相关的表单控件上。

解析:

最常用 *label* 标签的地方，应该就是表单中选择性别的单选框了。例如:

```
<label> 标签的 for 属性应当与相关元素的 id 属性相同。
<form>
<label for="male">男</label>
<input type="radio" name="sex" id="male" />
        <br />
<label for="female">女</label>
<input type="radio" name="sex" id="female" />
</form>
```

上面的代码中，当 *label* 和表单控件绑定后，用户不用必须点击单选框才能确定自己的选项，点击 *label* 所包裹的文字也能够自动聚焦绑定的表单控件。

7. *iframe* 框架有那些优缺点?

参考答案:

- *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，这样可以可以绕开以上两个问题。

8. *HTML* 与 *XHTML* 二者有什么区别，你觉得应该使用哪一个并说出理由。

参考答案:

HTML 与 *XHTML* 之间的差别，主要分为功能上的差别和书写习惯的差别两方面。

关于功能上的差别，主要是 *XHTML* 可兼容各大浏览器、手机以及 *PDA*，并且浏览器也能快速正确地编译网页。

由于 *XHTML* 的语法较为严谨，所以如果你是习惯松散结构的 *HTML* 编写者，那需要注意 *XHTML* 的规则。

下面列出了几条容易犯的错误，供理解。

1. **所有标签都必须小写**

在 *XHTML* 中，所有的标签都必须小写，不能大小写穿插其中，也不能全部都是大写。

2. **标签必须成双成对**

像是 ... 、<a>...、 ... 标签等，当出现一个标签时，必须要有对应的结束标签，缺一不可，就像在任何程序语言中的括号一样

3. **标签顺序必须正确**

标签由外到内，一层层包覆着，所以假设你先写 *div* 后写 *h1*，结尾就要先写 *h1* 后写 *div*。只要记住一个原则“先进后出”，先弹出的标签要后结尾。

4. **所有属性都必须使用双引号**

在 *XHTML 1.0* 中规定连单引号也不能使用，所以全程都得用双引号。

5. 不允许使用 *target="_blank"*

从 *XHTML 1.1* 开始全面禁止 *target* 属性，如果想要有开新窗口的功能，就必须改写为 *rel="external"*，并搭配 *JavaScript* 实现此效果。

9. *HTML5* 的 *form* 如何关闭自动完成功能?

参考答案:

HTML 的输入框可以拥有自动完成的功能，当你往输入框输入内容的时候，浏览器会从你以前的同名输入框的历史记录中查找出类似的内容并列在输入框下面，这样就不用全部输入进去了，直接选择列表中的项目就可以了。

使用 *autocomplete="off"*（给不想要提示的 *form* 或某个 *input* 设置为 *autocomplete=off*。）

很多时候，需要对客户的资料进行保密，防止浏览器软件或者恶意插件获取到;

可以在 *input* 中加入 *autocomplete=“off”* 来关闭记录系统需要保密的情况下可以使用此参数

> 提示：*autocomplete* 属性有可能在 *form* 元素中是开启的，而在 *input* 元素中是关闭的。
>
> 注意：*autocomplete* 适用于 <form> 标签，以及以下类型的 <input> 标签：*text, search, url, telephone, email, password, datepickers, range* 以及 *color*。

10. *title* 与 *h1* 的区别、*b* 与 *strong* 的区别、*i* 与 *em* 的区别?

> 参考答案:
>
> *title* 与 *h1*：*h1* 标签写在网页的 *body* 中，*title* 标签写在网页的 *head* 中，*h1* 标签控制一段文字的大小（从 *h1~h6*），*title* 是网页标题的意思，如 <title>这是网页标题</title>。
>
> *b* 与 *strong* 的区别：*b* 是以前的加粗元素，而新出来的 *strong* 元素虽然在表现上看上去也是加粗，但是却拥有语义，表示强调某段文字的信息。
>
> *i* 与 *em* 的区别:
>
> 同样，*i* 是 *italic*（斜体）的简写，是早期的斜体元素，而 *em* 是 *emphasize*（强调）的简写，是一个表示强调的元素。后者相比前者拥有语义。
>
> 经典的总结:
>
> - *title* 属性表示网页的标题，*h1* 元素则表示层次明确的页面内容标题，对页面信息的抓取也有很大的影响
> - *strong* 是标明重点内容，有语气加强的含义，使用阅读设备阅读网络时：会重读，而 <b>是展示强调内容
> - *i* 内容展示为斜体，*em* 表示强调的文本
>
> 除了上面所列举的那几组外，相似的还有如下的元素:
>
> 自然样式标签：*b*、*i*、*u*、*s*、*pre*
>
> 语义样式标签：*strong*、*em*、*ins*、*del*、*code*
>
> 应该准确使用语义样式标签，但不能滥用，如果不能确定时首选使用自然样式标签。

11. 请描述下 *SEO* 中的 *TDK*?

> 参考答案:
>
> 在 *SEO* 中，所谓的 *TDK* 其实就是 *title*、*description*、*keywords* 这三个标签，*title* 标题标签，*description* 描述标签，*keywords* 关键词标签。

12. 每个 *HTML* 文件头里都有个很重要的东西，*Doctype*，知道这是干什么的么?

> 参考答案：
>
> <!DOCTYPE> 声明位于文档中的最前面的位置，处于 <html> 标签之前，主要作用是告诉浏览器按照何种规范解析网页。

> 解析：
>
> *doctype* 声明是一种标准通用标记语言的文档类型声明，目的是告诉标准通用标记语言解析器要使用什么样的文档类型定义（*DTD*）来解析文档。
>
> 声明是用来指示 *web* 浏览器关于页面使用哪个 *HTML* 版本进行编写的指令。 声明必须是 *HTML* 文档的第一行，位于 <html> 标签之前。
>
> 浏览器本身分为两种模式，一种是标准模式，一种是怪异模式，浏览器通过 *doctype* 来区分这两种模式，*doctype* 在 *html* 中的作用就是触发浏览器的标准模式，如果 *html* 中省略了 *doctype*，浏览器就会进入到 *Quirks* 模式（怪异模式）。
>
> 在这种模式下，有些样式会和标准模式存在差异，而 *html* 标准和 *dom* 标准值规定了标准模式下的行为，没有对怪异模式做出规定，因此不同浏览器在怪异模式下的处理也是不同的，所以一定要在 *html* 开头使用 *doctype*。

13. 什么是严格模式与混杂模式?

> 参考答案：
>
> - 严格模式：以浏览器支持的最高标准运行
> - 混杂模式：页面以宽松向下兼容的方式显示，模拟老式浏览器的行为

14. 对于 *WEB* 标准以及 *W3C* 的理解与认识问题

> 参考答案：
>
> ***Web* 标准**简单来说可以分为**结构、表现和行为**。其中结构主要是有 *HTML* 标签组成。或许通俗点说，在页面 *body* 里面我们写入的标签都是为了页面的结构。表现即指 *css* 样式表，通过 *css* 可以是页面的结构标签更具美感。行为是指页面和用户具有一定的交互，同时页面结构或者表现发生变化，主要是有 *js* 组成。
>
> *W3C* 是一个定制各种标准的非盈利性组织，标准包括了 *HTML*、*CSS*、*XHTML* 等，*web* 标准制定后，有以下几个优点：
>
> 1. 学习成本降低，只学习标准即可，否则将学习各个浏览器厂商的标准。
> 2. 统一开发流程，用标准化工具开发（例如 *VSCode*、*WebStorm*、*Sublime* 等）再用标准化的浏览器测试，便于多人协作。

3. 简化网站代码的维护。
4. 跨平台，可方便迁移到不同设备中

15. 列举 *IE* 与其他浏览器不一样的特性?

参考答案:

- *IE* 的排版引擎是 *Trident* （又称为 *MSHTML*）
- *Trident* 内核曾经几乎与 *W3C* 标准脱节
- *Trident* 内核的大量 *Bug* 等安全性问题没有得到及时解决
- *JS* 方面，有很多独立的方法，例如绑定事件的 *attachEvent*、创建事件的 *createEventObject* 等
- *CSS* 方面，也有自己独有的处理方式，例如设置透明，低版本 *IE* 中使用滤镜的方式，盒模型也和 *W3C* 规定的盒模型不同

16. 前端页面有哪三层构成，分别是什么？作用是什么?

参考答案:

分成：结构层、表示层、行为层。

1. 结构层（*structural layer*）

由 *HTML* 或 *XHTML* 之类的标记语言负责创建。标签，也就是那些出现在尖括号里的单词，对网页内容的语义含义做出了描述，但这些标签不包含任何关于如何显示有关内容的信息。例如，*p* 标签表达了这样一种语义：“这是一个文本段。”

2. 表示层（*presentation layer*）

由 *CSS* 负责创建。 *CSS* 对“如何显示有关内容”的问题做出了回答。

3. 行为层（*behaviorlayer*）

负责回答“内容应该如何对事件做出反应”这一问题。这是 *Javascript* 语言和 *DOM* 主宰的领域。

17. 页面可见性（*Page Visibility*）*API* 可以有哪些用途?

参考答案:

所谓页面可见性，就是获取当前页面的可见状态。因为对于用户来讲可以打开好多标签页面来回切换，然而始终只有一个页面处于显示状态。所以我们可以通过页面可见性（*Page Visibility*）*API* 来判断当前页面是显示状态还是隐藏状态。

常用的 *API* 有三个，*document.hidden* 返回一个布尔值，如果是 *true*，表示页面可见，*false* 则表示页面隐藏。不同页面之间来回切换，会触发 *visibilitychange* 事件，还有一个 *document.visibilityState*，

> 表示页面所处的状态。

18. *Quirks*（怪癖）模式是什么？它和 *Standards*（标准）模式有什么区别

> 参考答案：
>
> 以 *IE6* 为例，如果写了 *DTD*，就意味着这个页面将采用对 *CSS* 支持更好的布局，而如果没有，则采用兼容之前的布局方式。这就是 *Quirks* 模式（怪癖模式，诡异模式，怪异模式）。
>
> 区别：总体会有布局、样式解析和脚本执行三个方面的区别，这里列举一些比较常见的区别：
>
> - 盒模型：在 *W3C* 标准中，如果设置一个元素的宽度和高度，指的是元素内容的宽度和高度，而在 *Quirks* 模式下，*IE* 的宽度和高度还包含了 *padding* 和 *border*。
> - 设置行内元素的高宽：在 *Standards* 模式下，给 等行内元素设置 *wdith* 和 *height* 都不会生效，而在 *Quirks* 模式下，则会生效。
> - 设置百分比的高度：在 *Standards* 模式下，一个元素的高度是由其包含的内容来决定的，如果父元素没有设置百分比的高度，子元素设置一个百分比的高度是无效的
> - 用 *margin:0 auto* 设置水平居中：使用 *margin:0 auto* 在 *Standards* 模式下可以使元素水平居中，但在 *Quirks* 模式下却会失效。

19. *div+css* 的布局较 *table* 布局有什么优点？

> 参考答案：
>
> - 样式的调整更加方便，内容和样式的分离，使页面和样式的调整变得更加方便。
> - 页面加载速度更快、结构化清晰、页面显示简洁。
> - 表现与结构相分离。
> - 易于优化（*SEO*）搜索引擎更友好，排名更容易靠前。
> - 符合 *W3C* 标准。

20. 请谈一下你对网页标准和标准制定机构重要性的理解。

> 参考答案：
>
> 任何东西都需要一个标准，有了标准才能够更好的进行交流和推广。不同的标准，得出的便是不同的结果。因此，制定什么样的标准，如何确立标准，至关重要。
>
> 正因为有了网页的标准，才能降低开发难度及开发成本，减少各种 *BUG*、安全问题， 提高网站易用性。

21. 知道什么是微格式吗？谈谈理解。在前端构建中应该考虑微格式吗？

> 参考答案：

参考答案：

所谓微格式，是建立在已有的、被广泛采用的标准基础之上的一组简单的、开放的数据格式。

具体表现是把语义嵌入到 *HTML* 中，以便有助于分离式开发，并通过制定一些简单的约定，来兼顾 *HTML* 文档的人机可读性，相当于对 *Web* 网页进行了语义注解。

采用微格式的 *Web* 页面，在 *HTML* 文档中给一些标签增加一些属性，这些属性对信息的语义结构进行注解，有助于处理 *HTML* 文档的软件，更好的理解该 *HTML* 文档。

在前端构建中微格式的意义

微格式按照某种已有的被广泛应用的标准，通过对内容块的语义标记，可以让外部应用程序、聚合程序和搜索引擎能够做以下事情：

1. 在爬取 *Web* 内容时，能够更为准确地识别内容块的语义；
2. 对内容进行操作，包括提供访问、校对，还可以将其转化成其他的相关格式，提供给外部程序和Web服务使用。

总结： 微格式可以对网站进行 *SEO* 优化，如果需要可以考虑。

22. *html* 常见兼容性问题?

参考答案：

1. *PNG24* 位的图片在 *IE6* 浏览器上出现背景

解决方案：做成 *PNG8*，也可以引用一段脚本处理。

2. 浏览器默认的 *margin* 和 *padding* 不同

解决方案：使用 *CSS* 重置文件。

3. *IE6* 双边距 *bug*

在 *IE6* 下，如果对元素设置了浮动，同时又设置了 *margin-left* 或 *margin-right*，*margin* 值会加倍。

```
#box{
 float:left;
 width:10px;
 margin:10px;
}
```

这种情况之下 *IE* 会产生 *20px* 的距离

解决方案：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）

4. 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* 游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。

```
.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*/
}
```

5. *IE* 下，获取自定义属性用获取常规

IE 下，可以使属性的方法来获取自定义属性用获取常规，也可以使用 *getAttribute( )* 方法获取自定义属性，*Firefox* 下，只能使用 *getAttribute( )* 方法获取自定义属性

解决方法：统一通过 *getAttribute( )* 方法获取自定义属性

6. *event* 对象的区别

IE 下，*event* 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。

7. *Chrome 12px* 像素

Chrome 中文界面下默认会将小于 *12px* 的文本强制按照 *12px* 显示

解决方法：可通过加入 *CSS* 属性 *-webkit-text-size-adjust: none;* 解决。

8. *hover* 和 *active* 失效

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：改变 *CSS* 属性的排列顺序 *L-V-H-A*

```
a:link {}
a:visited {}
a:hover {}
a:active {}
```

9. 怪异模式问题

漏写 *DTD* 声明，*Firefox* 仍然会按照标准模式来解析网页，但在 *IE* 中会触发怪异模式。

为避免怪异模式给我们带来不必要的麻烦，最好养成书写 *DTD* 声明的好习惯。

现在可以使用 *HTML5*, 推荐的写法：<!DOCTYPE html>

10. 上下 *margin* 重合问题

IE 和 *FireFox* 都存在，相邻的两个 *div* 的 *margin-left* 和 *margin-right* 不会重合，

但是 *margin-top* 和 *margin-bottom* 却会发生重合。

解决方法：养成良好的代码编写习惯，同时采用 *margin-top* 或者同时采用 *margin-bottom*。

11. *IE6* 对 *png* 图片格式支持不好

解决方案：引用一段脚本处理

23. *HTML5* 有哪些新特性、移除了那些元素?

参考答案:

- *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
- 语意化更好的内容元素，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
- 当然也可以直接使用成熟的框架、比如 *html5shim*

24. *HTML* 全局属性(*global attribute*)有哪些

参考答案:

所谓全局属性，就是指每个 *HTML* 元素都拥有的属性，大致有如下的属性:

- *class* :为元素设置类标识
- data-* : 为元素增加自定义属性
- *draggable* : 设置元素是否可拖拽
- *id* : 元素 *id* ，文档内唯一
- *lang* : 元素内容的的语言
- *style* : 行内 *css* 样式
- *title* : 元素相关的建议信息

25. *HTML5* 为什么只需要写 <!DOCTYPE HTML>?

参考答案:

为什么 *HTML5* 的顶部只需要一段

```
<!DOCTYPE html>
```

HTML4 却需要很长的一段

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TI
```

其主要原因，是因为 *HTML5* 不基于 *SGML*，所以不需要引用 *DTD*。

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。

26. 对 *web* 标准、可用性、可访问性的理解

参考答案:

- 可用性（*Usability*）：产品是否容易上手，用户能否完成任务，效率如何，以及这过程中用户的主观感受可好，是从用户的角度来看产品的质量。可用性好意味着产品质量高，是企业的核心竞争力。
- 可访问性（*Accessibility*）：Web 内容对于残障用户的可阅读和可理解性
- 可维护性（*Maintainability*）：一般包含两个层次，一是当系统出现问题时，快速定位并解决问题的成本，成本低则可维护性好。二是代码是否容易被人理解，是否容易修改和增强功能。

27. *HTML5* 引入什么新的表单属性?

参考答案:

***form* 新属性**:

- *autocomplete*：属性规定表单是否应该启用自动完成功能。*autocomplete* 属性适用于 <form>，以及下面的 <input>类型:

text, search, url, telephone, email, password, date，*pickers, range* 以及 *color*。语法是 <form autocomplete="on|off">

- *novalidate*：如果使用该属性，则提交表单时不进行内容的验证。*novalidate* 属性适用于：<form>，以及以下类型的 <input> 标签：*text, search, url, telephone, email, password, date pickers, range* 以及 *color*。语法：*novalidate="novalidate"*

***input* 新属性**:

- *autocomplete*：同上
- *autofocus*：规定输入字段在页面加载时是否获得焦点，加载完成后，光标马上定位在该 *input*
- *form*：*form* 属性的值必须是其所属表单的 *id*。如需引用一个以上的表单，请使用空格分隔的列表。
- *formaction*：属性覆盖 *form* 元素的 *action* 属性，比如两个提交按钮的时候，一个是正常提交，一个是管理员提交。该属性适用于 *type="submit"* 以及 *type="image"*。语法是 *formaction="#"*
- *formenctype*：*formenctype* 属性覆盖 *form* 元素的 *enctype* 属性。该属性与 *type="submit"* 和 *type="image"* 配合使用。属性规定在发送到服务器之前应该如何对表单数据进行编码。
- *formmethod*：覆盖表单的 method 属性。适用于 type=“submit” 和 type=“image”
- *formnovalidate*：*formnovalidate* 属性覆盖 *form* 元素的 *novalidate* 属性。如果使用该属性，则提交表单时按钮不会执行验证过程。
- *formtarget*：覆盖表单的 *target* 属性。适用于 *type="submit"* 和 *type="image"*， 该属性规定在何处打开 *action URL*。
- *height* 和 *width*：*height* 和 *width* 属性规定用于 *image* 类型的 <input> 标签的图像高度和宽度。
- *list*：引用包含输入字段的预定义选项的 *datalist* 。

- *min* 和 *max*：*min* 属性与 *max* 属性配合使用，可创建合法值范围，两个要一对用。语法是选择 *0-10* 数字，例如：<input type="number" name="points" min="0" max="10" />
- multiple：如果使用该属性，则允许一个以上的值，比如上传文件的时候，设置这个属性后可以一次选择几个图片；multiple 属性适用于以下类型的 `<input>` 标签：email 和 file。
- *pattern (regexp)*：描述了一个正则表达式用于验证 <input> 元素的值，*pattern* 属性适用于以下 <input> 类型：*text, search, url, telephone, email* 以及 *password*。
- *placeholder*：提供可描述输入字段预期值的提示信息 (*hint*)。该提示会在输入字段为空时显示，并会在字段获得焦点时消失。
- *required*：规定必需在提交之前填写输入字段。 如果使用该属性，则字段是必填（或必选）的。
- *step*：为输入域规定合法的数字间隔。 如果 *step="3"*，则合法的数是 *-3*、*0*、*3*、*6* 等。*step* 属性可以与 *max* 和 *min* 属性创建一个区域值。

28. *iframe* 的作用

参考答案：

参考答案：

iframe 也称作嵌入式框架，嵌入式框架和框架网页类似，它可以把一个网页的框架和内容嵌入在现有的网页中。

优点

- 重载页面时不需要重载整个页面，只需要重载页面中的一个框架页(减少了数据的传输，增加了网页下载速度)
- 方便制作导航栏

缺点

- 会产生很多页面，不容易管理
- 浏览器的后退按钮无效
- 无法被一些搜索引擎索引到
- 多数小型的移动设备（*PDA* 手机）无法完全显示框架
- 由于上面诸多缺点，因此不符合标准网页设计的理念，已经被标准网页设计抛弃

目前框架的所有优点完全可以使用 *Ajax* 实现，因此已经没有必要使用 *iframe* 框架了。

29. *img* 上 *title* 与 *alt*

参考答案：

- *alt*：如果无法显示图像，浏览器将显示 *alt* 指定的内容
- *title*：在鼠标移到元素上时显示 *title* 的内容

两者之间的区别：

通常当鼠标滑动到元素上的时候显示 *title*。

alt 是 <img> 的特有属性，是图片内容的等价描述，用于图片无法加载时显示、读屏器 阅读图片。可提图片高可访问性，除了纯装饰图片外都必须设置有意义的值，搜索引擎会重点分析。

30. *HTML5* 新增哪些新特性？

参考答案：

HTML5 新增特性有：

1. 拖拽释放
2. 语义化更好的内容标签
3. 视频、音频
4. 画布
5. 地理
6. 本地离线存储
7. 表单控件

31. 行内元素和块级元素区别，有哪些，怎样转换？（顶呱呱）

参考答案：

块级元素：

- 总是在新行上开始；
- 高度，行高以及外边距和内边距都可控制；
- 宽度缺省是它的容器的 *100%*，除非设定一个宽度。
- 它可以容纳内联元素和其他块元素

行内元素：

- 和其他元素都在一行上；
- 高，行高及外边距和内边距不可改变；
- 宽度就是它的文字或图片的宽度，不可改变
- 内联元素只能容纳文本或者其他内联元素

对行内元素，需要注意如下：

- 设置宽度 *width* 无效。
- 设置高度 *height* 无效，可以通过 *line-height* 来设置。
- 设置 *margin* 只有左右 *margin* 有效，上下无效。
- 设置 *padding* 只有左右 *padding* 有效，上下则无效。注意元素范围是增大了，但是对元素周围的内容是没影响的。

通过 *display* 属性对行内元素和块级元素进行切换(主要看第 *2*、*3*、*4* 个值):

值	**描述**
none	此元素不会被显示。
block	此元素将显示为块级元素，此元素前后会带有换行符。
inline	默认。此元素会被显示为内联元素，元素前后没有换行符。
inline-block	行内块元素。（CSS2.1 新增的值）
list-item	此元素会作为列表显示。
run-in	此元素会根据上下文作为块级元素或内联元素显示。
compact	CSS 中有值 compact，不过由于缺乏广泛支持，已经从 CSS2.1 中删除。
marker	CSS 中有值 marker，不过由于缺乏广泛支持，已经从 CSS2.1 中删除。
table	此元素会作为块级表格来显示（类似 <table>），表格前后带有换行符。
inline-table	此元素会作为内联表格来显示（类似 <table>），表格前后没有换行符。
table-row-group	此元素会作为一个或多个行的分组来显示（类似 <tbody>）。
table-header-group	此元素会作为一个或多个行的分组来显示（类似 <thead>）。
table-footer-group	此元素会作为一个或多个行的分组来显示（类似 <tfoot>）。
table-row	此元素会作为一个表格行显示（类似 <tr>）。
table-column-group	此元素会作为一个或多个列的分组来显示（类似 <colgroup>）。
table-column	此元素会作为一个单元格列显示（类似 <col>）
table-cell	此元素会作为一个表格单元格显示（类似 <td> 和 <th>）
table-caption	此元素会作为一个表格标题显示（类似 <caption>）
inherit	规定应该从父元素继承 display 属性的值。

html 中常见的块级元素：*p*、*div*、*form*、*ul*、*ol*、*table*

html 中常见的行内元素：*a*、*img*、*span*、*button*

32. *HTML5* 与 *HTML4* 的区别（华安永康）

参考答案:

1、语法简化

HTML、*XHTML* 的 *DOCTYPE*、*html*、*meta*、*script* 等标签，在 *HTML5* 中有大幅度的简化。

2、统一网页内嵌多媒体语法

以前，在网页中播放多媒体时，需要使用插件的方式来完成。有了 *HTML5* 之后，使用 `<video>` 或 `<audio>` 标签播放视频和音频，不需要在安装其他的什么来播放了。

3、新增了语义标签

为了增加网页的可读性，*HTML5* 增加了 <header>、<footer>、<section>、<article>、<nav>、<hgroup>、<aside>、<figure> 语义标签。

4、*HTML5* 废除了一些旧标签

废除的大部分是网页美化方面的标签，例如：<big>、<u>、<font>、<basefont>、 、<s>、<tt>。对 <frame>框架，不能使用。

5、全新的表单设计

表单是网页设计者最常用的功能，*HTML5* 对表单做了很大的更改，不但新增了几项新的标签，对原来的 <form> 标签也增加了许多属性。

6、新增了 <canvas> 标签，可以绘制图形

HTML5 新增了具有绘图功能的 <canvas>

7、新增许多新的 *API*

例如：*querySelector*、*querySelectorAll*、拖拽相关 *Api*

33. 如何处理 *HTML5* 新标签兼容问题

参考答案:

主要有两种方式:

方法 1：实现标签被识别

通过 *document.createElement(tagName)* 方法即可让浏览器识别新标签，浏览器支持新标签后，还可以为新标签添加 *CSS* 样式。

方法 2：*JavaScript* 解决方案

使用 *html5shim* 框架，在 `<head>` 中调用以下代码：

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

当然也可以直接把这个文件下载到自己的网站上，但这个文件必须在 *head* 标签中调用。

34. *h5* 和 *html5* 区别

参考答案：

H5 是一个产品名词，包含了最新的 *HTML5*、*CSS3*、*ES6* 等新的技术来制作的应用。

HTML5 是一个技术名词，指代的就仅仅是第五代 *HTML*。

35. *form* 表单上传文件时需要进行什么样的声明

参考答案：

需要添加如下的声明：

```
enctype="multipart/form-data"
```

36. 哪个属性可以做到当鼠标悬停在图片上时显示出文字

参考答案：

title 属性。当我们为图片设置了 *title* 属性后，鼠标悬停在图片上面，就会显示出 *title* 属性所设置的值。

37. 如何在一张图片上的某一个区域做到点击事件

参考答案：

可以使用图片热区技术。步骤如下：

1、插入图片，并设置好图像的有关参数，且在 <img> 标记中设置参数 *usemap="#Map"*，以表示对图像地图（*Map*）的引用；
2、用 <map> 标记设定图像地图的作用区域，并取名为：*Map*；
3、分别用 <area> 标记针对相应位置划分出多个矩形作用区域，并设定好其链接参数 *href*。

示例代码如下：

```
<body>
 <img src="url/to/your/image.jpg" alt="" usemap="#Map" />
    <map name="Map" id="Map">
     <area alt="" title="" href="#" shape="poly"
         coords="65,71,98,58,114,90,108,112,79,130,56,116,38,100,41,76,52,53,83,34,
     <area alt="" title="" href="#" shape="poly" coords="28,22,57,20,36,39,27,61" /
 </map>
</body>
```

38. 行内元素有哪些？块级元素有哪些？空（void）元素有哪些？

参考答案：

- 行内元素：最大的特点就是共享一行，常见的有 *span a em i b strong abbr input select* 等
- 块级元素：最大的特点就是独占一行，常见的有 *header footer nav section article aside div p ul li ol dl table* 等
- 空元素：就是没有内容的 *HTML* 元素，比较常见的空元素有 *img*、*link*、*meta*、*br*、*hr*

39. 什么是锚点？

参考答案：

锚点（*anchor*）是一种特殊连接，能定位到 *HTML* 文档中某个特定位置，通过 HTML 元素的 *id* 来设置锚点。

40. 图片与 *span* 元素混排图像下方会出现几像素的空隙的原因是什么？

参考答案：

img 作为可替换元素，它没有自己的基线，如果与不可替换元素混合排列，其行盒底端与基线对齐。由于与基线对齐，图像下方就会出现几像素的空隙。

41. *a* 元素除了用于导航外，还可以有什么作用？

参考答案：

> *href* 属性中的 *url* 可以是浏览器支持的任何协议，所以 *a* 可以用于手机拨号 <a href='tel:10086'>10086、发送短信 <a href="sms:10086?body=test"> 等。
>
> 当然，*a* 元素最常见的两个应用就是做锚点和下载文件。
>
> 锚点可以在点击时快速定位到一个页面的某一个位置，而下载的原理在于 *a* 标签所对应的资源浏览器无法解析，于是浏览器会选择将其下载下来。