

1. 简述一下你对 HTML 语义化的理解

①用正确的标签做正确的事情。

②html 语义化让页面的内容结构化，结构更清晰，便于对浏览器、搜索引擎解析；即使在没有样式 CSS 情况下也以一种文档格式显示，并且是容易阅读的；

③ 搜索引擎的爬虫也依赖于 HTML 标记来确定上下文和各个关键字的权重，利于 SEO；

④ 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。

2. Label 的作用是什么？是怎么用的？

答案：label 标签来定义表单控制间的关系，当用户选择该标签时，浏览器会自动将焦点转到和标签相关的表单控件上。

解析：两种用法：一种是 id 绑定，一种是嵌套

```
<label for="Name">Number:</label>
<input type="text" name="Name" id="Name"/>
<label>Date:<input type="text" name="B"/></label>
```

3. iframe 框架有那些优缺点？

优点：

- iframe 能够原封不动的把嵌入的网页展现出来。
- 如果有多个网页引用 iframe，那么你只需要修改 iframe 的内容，就可以实现调用的每一个页面内容的更改，方便快捷。
- 网页如果为了统一风格，头部和版本都是一样的，就可以写成一个页面，用 iframe 来嵌套，可以增加代码的可重用。
- 如果遇到加载缓慢的第三方内容如图标和广告，这些问题可以由 iframe 来解决。

缺点：

- 框架结构中出现各种滚动条
- iframe 会阻塞主页面的 Onload 事件
- 搜索引擎的检索程序无法解读这种页面，不利于 SEO
- 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。

4. 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。

应该使用 XHTML，因为 XHTML 是 XML 重写了 HTML 的规范，比 HTML 更加严格，表现如下：

- 1、XHTML 中所有的标记都必须有一个相应的结束标签；
- 2、XHTML 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、所有的属性都必须用引号“”括起来；
- 5、把所有<和&特殊符号用编码表示；
- 6、给所有属性附一个值；
- 7、不要在注释内容中使用“—”；
- 8、图片必须使用说明文字。

5. HTML5 的 form 如何关闭自动完成功能？

答案：将不想要自动完成的 form 或 input 设置为 autocomplete=off

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

①title 用于网站信息标题，突出网站标题或关键字，一个网站可以有多个 title，seo 权重高于 H1；H1 概括的是文章主题，一个页面最好只用一个 H1，seo 权重低于 title。

②b 为了加粗而加粗，strong 为了标明重点而加粗

③ 同②i 为了斜体而斜体，em 为了标明重点而斜体，且对于搜索引擎来说 strong 和 em 比 b 和 i 要重视的多

7. 请描述下 SEO 中的 TDK？

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

8. 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？

答案：<!DOCTYPE> 声明位于文档中的最前面的位置，处于 <html> 标签之前。

1. 告知浏览器文档使用哪种 HTML 或 XHTML 规范。

2. 告诉浏览器按照何种规范解析页（如果你的页面没有 DOCTYPE 的声明，那么 compatMode 默认就是 BackCompat, 浏览器按照自己的方式解析渲染页面）

解析：

doctype 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 模式的怪异状态，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。

9. 简述一下 src 与 href 的区别。

- 答案：src 用于引用资源，替换当前元素；href 用于在当前文档和引用资源之间确立联系。href 标识超文本引用，用在 link 和 a 等元素上，href 是引用和页面关联，是在当前元素和引用资源之间建立联系。若在文档中添加 href，浏览器会识别该文档为 CSS 文件，就会并行下载资源并且不会停止对当前文档的处理。这也是为什么建议使用 link 方式加载 CSS，而不是使用 @import 方式。
- src 表示引用资源，替换当前元素，用在 img, script, iframe 上，src 是页面内容不可缺少的一部分。当浏览器解析到 src，会暂停其他资源的下载和处理（图片不会暂停其他资源下载和处理），直到将该资源加载、编译、执行完毕，图片和框架等也如此，类似于将所指向资源应用到当前内容。这也是为什么建议把 js 脚本放在底部而不是头部的原因。

10. 严格模式与混杂模式

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

11. 对于 WEB 标准以及 W3C 的理解与认识问题

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。

web 标准一般是将该三部分独立分开，使其更具有模块化。但一般产生行为时，就会有结构或者表现的变化，也使这三者的界限并不那么清晰。

W3C 对 web 标准提出了规范化的要求，也就是在实际编程中的一些代码规范：包含如下几点

1. 对于结构要求：（标签规范可以提高搜索引擎对页面的抓取效率，对 SEO 很有帮助）

- 1) 标签字母要小写
- 2) 标签要闭合
- 3) 标签不允许随意嵌套

2. 对于 css 和 js 来说

- 1) 尽量使用外链 css 样式表和 js 脚本。是结构、表现和行为分为三块，符合规范。同时提高页面渲染速度，提高用户的体验。
- 2) 样式尽量少用行内样式表，使结构与表现分离，标签的 id 和 class 等属性命名要做到见文知义，标签越少，加载越快，用户体验提高，代码维护简单，便于改版
- 3) 不需要变动页面内容，便可提供打印版本而不需要复制内容，提高网站易用性。

12. 列举 IE 与其他浏览器不一样的特性？

- a. IE 的排版引擎是 Trident（又称为 MSHTML）
- b. Trident 内核曾经几乎与 W3C 标准脱节（2005 年）
- c. Trident 内核的大量 Bug 等安全性问题没有得到及时解决
- d. 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
- e. 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

13. 前端页面有哪三层构成，分别是什么？作用是什么？

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 (behavior layer)

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。

14. 网页验证码是干嘛的，是为了解决什么安全问题？

答案：

- 区分用户是计算机还是人的公共全自动程序。可以防止恶意破解密码、刷票、论坛灌水
- 有效防止黑客对某一个特定注册用户用特定程序暴力破解方式进行不断的登陆尝试

15. 为什么用多个域名存储网站资源更有效？

- 1、CDN 缓存更方便
- 2、突破浏览器并发限制
- 3、节约 cookie 带宽
- 4、节约主域名的连接数，优化页面响应速度
- 5、防止不必要的安全问题

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

页面可见性：就是对于用户来说，页面是显示还是隐藏，所谓显示的页面，就是我们正在看的页面；隐藏的页面，就是我们没有看的页面。因为，我们一次可以打开好多标签页面来回切换着，始终只有一个页面在我们眼前，其他页面就是隐藏的，还有一种就是.....，（把浏览器最小化，所有的页面就都不可见了）。

API 很简单，document.hidden 就返回一个布尔值，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，触发 visibilitychange 事件。还有一个 document.visibilityState，表示页面所处的状态，取值：visible, hidden 等四个。

```
document.addEventListener("visibilitychange", function() {  
    if (document.hidden) {  
        document.title = "hidden";  
    } else {  
        document.title = "visible";  
    }  
});
```

我们打开这个页面，然后再打开另一个页面，来回点击这两个页面，当我们看到这个页面时，标题显示 visible，当我们看另一个页面时，标题显示 hidden；

动画，视频，音频都可以在页面显示时打开，在页面隐藏时关闭

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

1 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）。

2 区别：总体会有布局、样式解析和脚本执行三个方面的区别。

设置一个元素的宽度和高度

给等行内元素设置 width 和 height

用 margin:0 auto 设置水平居中

从 IE6 开始，引入了 Standards 模式，标准模式中，浏览器尝试给符合标准的文档在规范上的正确处理达到在指定浏览器中的程度。

在 IE6 之前 CSS 还不够成熟，所以 IE5 等之前的浏览器对 CSS 的支持很差，IE6 将对 CSS 提供更好的支持，然而这时的问题就来了，因为有很多页面是基于旧的布局方式写的，而如果 IE6 支持 CSS 则将令这些页面显示不正常，如何在即保证不破坏现有页面，又提供新的渲染机制呢？

在写程序时我们也会经常遇到这样的问题，如何保证原来的接口不变，又提供更强大的功能，尤其是新功能不兼容旧功能时。遇到这种问题时的一个常见做法是增加参数和分支，即当某个参数为真时，我们就使用新功能，而

如果这个参数 不为真时，就使用旧功能，这样就能不破坏原有的程序，又提供新功能。IE6 也是类似这样做的，它将 DTD（文档类型定义）当成了这个“参数”，因为以前的页面大家都不会去写 DTD，所以 IE6 就假定如果写了 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。

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 width 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。

18. div+css 的布局较 table 布局有什么优点？

答案：分离 方便改版 快清晰简洁 seo

1. 改版的时候更方便 只要改 css 文件。
2. 页面加载速度更快、结构化清晰、页面显示简洁。
3. 表现与结构相分离。
4. 易于优化（seo）搜索引擎更友好，排名更容易靠前。

19. 你能描述一下渐进增强和优雅降级之间的不同吗？

渐进增强 progressive enhancement：针对低版本浏览器进行构建页面，保证最基本的功能，然后再针对高级浏览器进行效果、交互等改进和追加功能达到更好的用户体验。

（一开始保证最基本的功能，再改进和追加功能）

优雅降级 graceful degradation：一开始就构建完整的功能，然后再针对低版本浏览器进行兼容。

（一开始就构建完整的功能，再针对低版本浏览器进行兼容。）

区别：优雅降级是从复杂的现状开始，并试图减少用户体验的供给，而渐进增强则是从一个非常基础的，能够起作用的版本开始，并不断扩充，以适应未来环境的需要。降级（功能衰减）意味着往回看；而渐进增强则意味着朝前看，同时保证其根基处于安全地带。

20. 请谈一下你对网页标准和标准制定机构重要性的理解。

答案：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？

答案：微格式（Microformats）是一种让机器可读的语义化 XHTML 词汇的集合，是结构化数据的开放标准。是为特殊应用而制定的特殊格式。

优点：将智能数据添加到网页上，让网站内容在搜索引擎结果界面可以显示额外的提示。（应用范例：豆瓣，有兴趣自行 google）

22. html 常见兼容性问题？

1. 双边距 BUG float 引起的，解决办法：使用 display 解决
2. 3 像素问题 使用 float 引起的，解决办法：使用 display:inline -3px
3. 超链接 hover 点击后失效，解决办法：使用正确的书写顺序 link visited hover active
4. Ie z-index 问题，解决办法：给父级添加 position:relative
5. Png 透明，解决办法：使用 js 代码
6. Min-height 最小高度，解决办法：! Important 解决
7. select 在 ie6 下遮盖，解决办法：使用 iframe 嵌套
8. 为什么没有办法定义 1px 左右的宽度容器，解决办法：（IE6 默认的行高造成的，使用 over:hidden, zoom:0.08 line-height:1px）
9. IE5-8 不支持 opacity，解决办法：

```
.opacity {  
    opacity: 0.4;
```



```
filter: alpha(opacity=60); /* for IE5-7 */
-ms-filter: "progid:DXImageTransform.Microsoft.Alpha(Opacity=60)"; /* for IE 8 */
}
```

10. IE6 不支持 PNG 透明背景，解决办法：IE6 下使用 gif 图片

23. 对 WEB 标准以及 W3C 的理解与认识

答案：标签闭合、标签小写、不乱嵌套、提高搜索机器人搜索几率、使用外链 css 和 js 脚本、结构行为表现的分离、文件下载与页面速度更快、内容能被更多的用户所访问、内容能被更广泛的设备所访问、更少的代码和组件，容易维护、改版方便，不需要变动页面内容、提供打印版本而不需要复制内容、提高网站易用性。

25. 前端需要注意哪些 SEO

1. 合理的 title、description、keywords：搜索对着三项的权重逐个减小，title 值强调重点即可，重要关键词出现不要超过 2 次，而且要靠前，不同页面 title 要有所不同；description 把页面内容高度概括，长度合适，不可过分堆砌关键词，不同页面 description 有所不同；keywords 列举出重要关键词即可
2. 语义化的 HTML 代码，符合 W3C 规范：语义化代码让搜索引擎容易理解网页
3. 重要内容 HTML 代码放在最前：搜索引擎抓取 HTML 顺序是从上到下，有的搜索引擎对抓取长度有限制，保证重要内容一定会被抓取
4. 重要内容不要用 js 输出：爬虫不会执行 js 获取内容
5. 少用 iframe：搜索引擎不会抓取 iframe 中的内容
6. 非装饰性图片必须加 alt
7. 提高网站速度：网站速度是搜索引擎排序的一个重要指标

26. html5 有哪些新特性、移除了那些元素？

新特性：

1. 拖拽释放(Drag and drop) API
2. 语义化更好的内容标签 (header, nav, footer, aside, article, section)
3. 音频、视频 API(audio, video)
4. 画布(Canvas) API
5. 地理(Geolocation) API
6. 本地离线存储 localStorage 长期存储数据，浏览器关闭后数据不丢失；
7. sessionStorage 的数据在浏览器关闭后自动删除
8. 表单控件，calendar、date、time、email、url、search
9. 新的技术 webworker, websocket, Geolocation

移除的元素：

1. 纯表现的元素：basefont, big, center, font, s, strike, tt, u;
2. 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风格：
- 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]>
```

27. HTML5 的离线储存怎么使用，工作原理能不能解释一下？

在用户没有与因特网连接时，可以正常访问站点或应用，在用户与因特网连接时，更新用户机器上的缓存文件。

原理：HTML5 的离线存储是基于一个新建的 .appcache 文件的缓存机制(不是存储技术)，通过这个文件上的解析清单离线存储资源，这些资源就会像 cookie 一样被存储了下来。之后当网络在处于离线状态下时，浏览器会通过被离线存储的数据进行页面展示。

使用方法

只要在头部加一个 manifest 属性就 ok 了

```
<!DOCTYPE html>
<html manifest="cache.manifest">
  ...
</html>
```

然后 cache.manifest 文件的书写方式如下:

```
CACHE MANIFEST
#v0.11
CACHE:
js/app.js
css/style.css
NETWORK:
resource/logo.png
FALLBACK:
/ /offline.html
```

离线存储的 manifest 一般由三个部分组成:

1. CACHE:表示需要离线存储的资源列表, 由于包含 manifest 文件的页面将被自动离线存储, 所以不需要把页面自身也列出来。
2. NETWORK:表示在它下面列出来的资源只有在在线的情况下才能访问, 他们不会被离线存储, 所以在离线情况下无法使用这些资源。不过, 如果在 CACHE 和 NETWORK 中有一个相同的资源, 那么这个资源还是会被离线存储, 也就是说 CACHE 的优先级更高。
3. FALLBACK:表示如果访问第一个资源失败, 那么就使用第二个资源来替换他, 比如上面这个文件表示的就是如果访问根目录下任何一个资源失败了, 那么就去访问 offline.html。

28. 浏览器是怎么对 HTML5 的离线储存资源进行管理和加载的呢

答案: 在线的情况下, 浏览器发现 html 头部有 manifest 属性, 它会请求 manifest 文件, 如果是第一次访问 app, 那么浏览器就会根据 manifest 文件的内容下载相应的资源并且进行离线存储。如果已经访问过 app 并且资源已经离线存储了, 那么浏览器就会使用离线的资源加载页面, 然后浏览器会对比新的 manifest 文件与旧的 manifest 文件, 如果文件没有发生改变, 就不做任何操作, 如果文件改变了, 那么就会重新下载文件中的资源并进行离线存储。 离线的情况下, 浏览器就直接使用离线存储的资源。

29. HTML 全局属性(global attribute)有哪些

- accesskey:设置快捷键, 提供快速访问元素如 [aaa](#) 在 windows 下的 firefox 中按 alt + shift + a 可激活元素
- class:为元素设置类标识, 多个类名用空格分开, CSS 和 javascript 可通过 class 属性获取元素
- contenteditable: 指定元素内容是否可编辑
- contextmenu: 自定义鼠标右键弹出菜单内容
- data-*: 为元素增加自定义属性
- dir: 设置元素文本方向
- draggable: 设置元素是否可拖拽
- dropzone: 设置元素拖放类型: copy, move, link
- hidden: 表示一个元素是否与文档。样式上会导致元素不显示, 但是不能用这个属性实现样式效果
- id: 元素 id, 文档内唯一
- lang: 元素内容的语言
- spellcheck: 是否启动拼写和语法检查
- style: 行内 css 样式
- tabindex: 设置元素可以获得焦点, 通过 tab 可以导航
- title: 元素相关的建议信息
- translate: 元素和子孙节点内容是否需要本地化

30. Canvas 和 SVG 有什么区别?

答案: Canvas 和 SVG 都允许您在浏览器中创建图形, 但是它们在根本上是不同的。

Canvas

通过 Javascript 来绘制 2D 图形。是逐像素进行渲染的。其位置发生改变，会重新进行绘制。

SVG

一种使用 XML 描述的 2D 图形的语言 SVG 基于 XML 意味着，SVG DOM 中的每个元素都是可用的，可以为某个元素附加 Javascript 事件处理器。在 SVG 中，每个被绘制的图形均被视为对象。如果 SVG 对象的属性发生变化，那么浏览器能够自动重现图形。

比较

Canvas

- 依赖分辨率
- 不支持事件处理器
- 弱的文本渲染能力
- 能够以 .png 或 .jpg 格式保存结果图像
- 最适合图像密集型的游戏，其中的许多对象会被频繁重绘

SVG

- 不依赖分辨率
- 支持事件处理器
- 最适合带有大型渲染区域的应用程序（比如谷歌地图）
- 复杂度高会减慢渲染速度（任何过度使用 DOM 的应用都不快）
- 不适合游戏应用

31. HTML5 为什么只需要写 `<!DOCTYPE HTML>`?

答案：HTML 4.01 中的 doctype 需要对 DTD 进行引用，因为 HTML 4.01 基于 SGML。而 HTML 5 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。其中，SGML 是标准通用标记语言，简单的说，就是比 HTML, XML 更老的标准，这两者都是由 SGML 发展而来的。BUT，HTML5 不是的。

`<!DOCTYPE>` 声明位于 HTML 文档中的第一行，处于 `<html>` 标签之前。作用：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怪异模式呈现。

32. meta viewport 原理是什么？

meta viewport 标签的作用是让当前 viewport 的宽度等于设备的宽度，同时不允许用户进行手动缩放

viewport 的原理：移动端浏览器通常都会在一个比移动端屏幕更宽的虚拟窗口中渲染页面，这个虚拟窗口就是 viewport；目的是正常展示没有做移动端适配的网页，让他们完整的展示给用户；

解析：

Viewport：字面意思为视图窗口，在移动 web 开发中使用。表示将设备浏览器宽度虚拟成一个特定的值（或计算得出），这样利于移动 web 站点跨设备显示效果基本一致。移动版的 Safari 浏览器最新引进了 viewport 这个 meta tag，让网页开发者来控制 viewport 的大小和缩放，其他手机浏览器也基本支持。

在移动端浏览器当中，存在着两种视口，一种是可见视口（也就是我们说的设备大小），另一种是视窗视口（网页的宽度是多少）。举个例子：如果我们的屏幕是 320 像素 * 480 像素的大小（iPhone4），假设在浏览器中，320 像素的屏幕宽度能够展示 980 像素宽度的内容。那么 320 像素的宽度就是可见视口的宽度，而能够显示的 980 像素的宽度就是视窗视口的宽度。

为了显示更多的内容，大多数的浏览器会把自己的视窗视口扩大，简易的理解，就是让原本 320 像素的屏幕宽度能够容下 980 像素甚至更宽的内容（将网页等比例缩小）。

Viewport 属性值

- width 设置 layout viewport 的宽度，为一个正整数，或字符串“width=device”
- initial-scale 设置页面的初始缩放值，为一个数字，可以带小数
- minimum-scale 允许用户的最小缩放值，为一个数字，可以带小数
- maximum-scale 允许用户的最大缩放值，为一个数字，可以带小数
- height 设置 layout viewport 的高度，这个属性对我们并不重要，很少使用
- user-scalable 是否允许用户进行缩放，值为“no”或“yes”，no 代表不允许，yes 代表允许这些属性可以同时使用，也可以单独使用或混合使用，多个属性同时使用时用逗号隔开就行了。

33. 对 web 标准、可用性、可访问性的理解

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。

34. HTML5 引入什么新的表单属性？

答案：Datalist datetime output keygen date month week time number range email url

35. 新的 HTML5 文档类型和字符集是？

答案：

HTML5 文档类型：<!doctype html>

HTML5 使用的编码<meta charset=" UTF-8" >

36. HTML5 Canvas 元素有什么用？

答案：Canvas 元素用于在网页上绘制图形，该元素标签强大之处在于可以直接在 HTML 上进行图形操作。

37. HTML5 存储类型有什么区别？

答案：Media API、Text Track API、Application Cache API、User Interaction、Data Transfer API、Command API、Constraint Validation API、History API

38. iframe 的作用

答案：iframe 是用来在网页中插入第三方页面，早期的页面使用 iframe 主要是用于导航栏这种很多页面都相同的部分，这样在切换页面的时候避免重复下载。

优点

1. 便于修改，模拟分离，像一些信息管理系统会用到。
2. 但现在基本不推荐使用。除非特殊需要，一般不推荐使用。

缺点

1. iframe 的创建比一般的 DOM 元素慢了 1-2 个数量级
2. iframe 标签会阻塞页面的加载，如果页面的 onload 事件不能及时触发，会让用户觉得网页加载很慢，用户体验不好，在 Safari 和 Chrome 中可以通过 js 动态设置 iframe 的 src 属性来避免阻塞。
3. iframe 对于 SEO 不友好，替换方案一般就是动态语言的 Include 机制和 ajax 动态填充内容等。

39. 为什么最好把 CSS 的<link>标签放在<head></head>之间？为什么最好把 JS 的<script>标签恰好放在</body>之前，有例外情况吗？

把<link>放在<head>中

把<link>标签放在<head></head>之间是规范要求的内容。此外，这种做法可以让页面逐步呈现，提高了用户体验。将样式表放在文档底部附近，会使许多浏览器（包括 Internet Explorer）不能逐步呈现页面。一些浏览器会阻止渲染，以避免在页面样式发生变化时，重新绘制页面中的元素。这种做法可以防止呈现给用户空白的页面或没有样式的内容。

把<script>标签恰好放在</body>之前

脚本在下载和执行期间会阻止 HTML 解析。把<script>标签放在底部，保证 HTML 首先完成解析，将页面尽早呈现给用户。

例外情况是当你的脚本里包含 document.write() 时。但是现在，document.write() 不推荐使用。同时，将<script>标签放在底部，意味着浏览器不能开始下载脚本，直到整个文档（document）被解析。也许，对此比较好的做法是，<script>使用 defer 属性，放在<head>中。

40. 什么是渐进式渲染 (progressive rendering) ？

答案：渐进式渲染是用于提高网页性能（尤其是提高用户感知的加载速度），以尽快呈现页面的技术。

在以前互联网带宽较小的时期，这种技术更为普遍。如今，移动终端的盛行，而移动网络往往不稳定，渐进式渲染在现代前端开发中仍然有用武之地。

一些举例：

- 图片懒加载——页面上的图片不会一次性全部加载。当用户滚动页面到图片部分时，JavaScript 将加载并显示图像。

- 确定显示内容的优先级（分层次渲染）——为了尽快将页面呈现给用户，页面只包含基本的最少量的 CSS、脚本和内容，然后可以使用延迟加载脚本或监听 DOMContentLoaded/load 事件加载其他资源和内容。
- 异步加载 HTML 片段——当页面通过后台渲染时，把 HTML 拆分，通过异步请求，分块发送给浏览器。

41. DOM 和 BOM 有什么区别

- DOM

Document Object Model，文档对象模型

DOM 是为了操作文档出现的 API，document 是其的一个对象

DOM 和文档有关，这里的文档指的是网页，也就是 html 文档。DOM 和浏览器无关，他关注的是网页本身的内容。

- BOM

Browser Object Model，浏览器对象模型

BOM 是为了操作浏览器出现的 API，window 是其的一个对象

window 对象既为 javascript 访问浏览器提供 API，同时在 ECMAScript 中充当 Global 对象

42. img 上 title 与 alt

答案: title 指图片的信息、alt 指图片不显示时显示的文字

43. 一个 iframe，内嵌了一个 A 页面，iframe 的宽高不停变化，如何让 A 页面的宽高实时自适应这个 iframe 的宽高大小。请说出至少 3 种方法，越难越好

答案:

1. css 的方案
2. onresize
3. 监听鼠标动作，鼠标释放后重新定宽

44. DOM Tree 是如何构建的？

答案:

1. HTML 解释器

HTML 解释器的工作就是将网络或者本地磁盘获取的 HTML 网页和资源从字节流解释成 DOM 树结构。

2. JavaScript 的执行

在 HTML 解释器的工作过程中，可能会有 JavaScript 代码需要执行，它发生在将字符串解释成词语之后、创建各种节点的时候。这也是为什么全局执行的 JavaScript 代码不能访问 DOM 的原因——因为 DOM 树还没有被创建完呢。

45. 100*100 的 canvas 占多少内存？

100 * 100 canvas 占的内存是 100 * 100 * 4 bytes = 40,000 bytes