1. src和href的区别

src和href都是用来引用外部的资源，它们的区别如下：

- **src：**表示对资源的引用，它指向的内容会嵌入到当前标签所在的位置。src会将其指向的资源下载并应用到文档内，如请求js脚本。当浏览器解析到该元素时，会暂停其他资源的下载和处理，直到将该资源加载、编译、执行完毕，所以一般js脚本会放在页面底部。
- **href：**表示超文本引用，它指向一些网络资源，建立和当前元素或本文档的链接关系。当浏览器识别到它他指向的文件时，就会并行下载资源，不会停止对当前文档的处理。 常用在a、link等标签上。

2. 对HTML语义化的理解

语义化是指****根据内容的结构化（内容语义化），选择合适的标签（代码语义化）。通俗来讲就是用正确的标签做正确的事情。

语义化的优点如下：

- 对机器友好，带有语义的文字表现力丰富，更适合搜索引擎的爬虫爬取有效信息，有利于SEO。除此之外，语义类还支持读屏软件，根据文章可以自动生成目录；
- 对开发者友好，使用语义类标签增强了可读性，结构更加清晰，开发者能清晰的看出网页的结构，便于团队的开发与维护。

常见的语义化标签：

```
<header></header>  头部

<nav></nav>  导航栏

<section></section>  区块（有语义化的div）

<main></main>  主要区域

<article></article>  主要内容

<aside></aside>  侧边栏

<footer></footer>  底部
```

3. DOCTYPE(文档类型) 的作用

DOCTYPE是HTML5中一种标准通用标记语言的文档类型声明，它的目的是**告诉浏览器（解析器）应该以什么样（html或xhtml）的文档类型定义****来解析文档**，不同的渲染模式会影响浏览器对 CSS 代码甚至 JavaScript 脚本的解析。它必须声明在HTML文档的第一行。

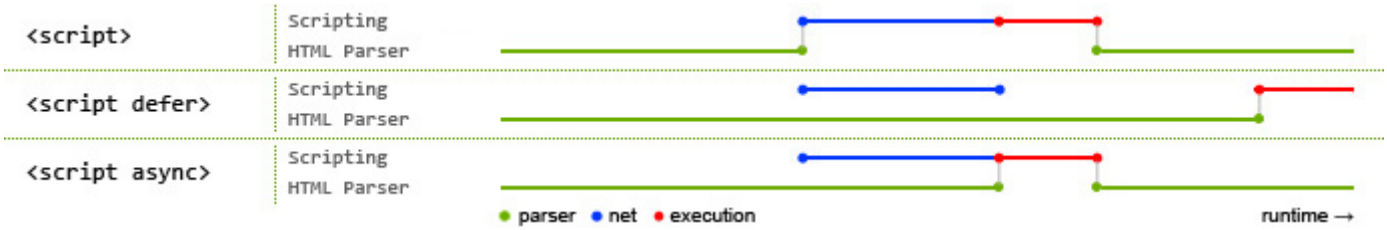
浏览器渲染页面的两种模式（可通过document.compatMode获取，比如，语雀官网的文档类型是**CSS1Compat**）：

- **CSS1Compat：标准模式（Strick mode）**，默认模式，浏览器使用W3C的标准解析渲染页面。在标准模式中，浏览器以其支持的最高标准呈现页面。
- **BackCompat：怪异模式(混杂模式)(Quick mode)**，浏览器使用自己的怪异模式解析渲染页面。在怪异模式中，页面以一种比较宽松的向后兼容的方式显示。

4. script标签中defer和async的区别

如果没有defer或async属性，浏览器会立即加载并执行相应的脚本。它不会等待后续加载的文档元素，读取到就会开始加载和执行，这样就阻塞了后续文档的加载。

下图可以直观的看出三者之间的区别:



其中蓝色代表js脚本网络加载时间，红色代表js脚本执行时间，绿色代表html解析。

defer 和 async属性都是去异步加载外部的JS脚本文件，它们都不会阻塞页面的解析，其区别如下：

- **执行顺序：**多个带async属性的标签，不能保证加载的顺序；多个带defer属性的标签，按照加载顺序执行；
- **脚本是否并行执行：async属性，表示后续文档的加载和执行与js脚本的加载和执行是并行进行的，**即异步执行；defer属性，加载后续文档的过程和js脚本的加载(此时仅加载不执行)是并行进行的(异步)，js脚本需要等到文档所有元素解析完成之后才执行，DOMContentLoaded事件触发执行之前。

5. 常用的meta标签有哪些

meta 标签由 name 和 content 属性定义，**用来描述网页文档的属性**，比如网页的作者，网页描述，关键词等，除了HTTP标准固定了一些name作为大家使用的共识，开发者还可以自定义name。

常用的meta标签：

（1）charset，用来描述HTML文档的编码类型：

```
<meta charset="UTF-8" >
```

（2） keywords，页面关键词：

```
<meta name="keywords" content="关键词" />
```

（3）description，页面描述：

```
<meta name="description" content="页面描述内容" />
```

（4）refresh，页面重定向和刷新：

```
<meta http-equiv="refresh" content="0;url=" />
```

（5）viewport，适配移动端，可以控制视口的大小和比例：

```
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-
scale=1">
```

其中，content 参数有以下几种：

- width viewport：宽度(数值/device-width)
- height viewport：高度(数值/device-height)
- initial-scale：初始缩放比例
- maximum-scale：最大缩放比例
- minimum-scale：最小缩放比例
- user-scalable：是否允许用户缩放(yes/no）

（6）搜索引擎索引方式：

```
<meta name="robots" content="index,follow" />
```

其中，content 参数有以下几种：

- all：文件将被检索，且页面上的链接可以被查询；
- none：文件将不被检索，且页面上的链接不可以被查询；
- index：文件将被检索；
- follow：页面上的链接可以被查询；
- noindex：文件将不被检索；
- nofollow：页面上的链接不可以被查询。

6. HTML5有哪些更新

1. 语义化标签

- header：定义文档的页眉（头部）；
- nav：定义导航链接的部分；
- footer：定义文档或节的页脚（底部）；

- article：定义文章内容；
- section：定义文档中的节（section、区段）；
- aside：定义其所处内容之外的内容（侧边）；

2. 媒体标签

（1） audio：音频

```
<audio src='' controls autoplay loop='true'></audio>
```

属性：

- controls 控制面板
- autoplay 自动播放
- loop=‘true’ 循环播放

（2）video视频

```
<video src='' poster='imgs/aa.jpg' controls></video>
```

属性：

- poster：指定视频还没有完全下载完毕，或者用户还没有点击播放前显示的封面。默认显示当前视频文件的第一针画面，当然通过poster也可以自己指定。
- controls 控制面板
- width
- height

（3）source标签

因为浏览器对视频格式支持程度不一样，为了能够兼容不同的浏览器，可以通过source来指定视频源。

```
<video>
    <source src='aa.flv' type='video/flv'></source>
    <source src='aa.mp4' type='video/mp4'></source>
</video>
```

3. 表单

表单类型：

- email ： 能够验证当前输入的邮箱地址是否合法
- url ： 验证URL
- number ： 只能输入数字，其他输入不了，而且自带上下增大减小箭头，max属性可以设置为最大值，min可以设置为最小值，value为默认值。
- search ： 输入框后面会给提供一个小叉，可以删除输入的内容，更加人性化。
- range ： 可以提供给一个范围，其中可以设置max和min以及value，其中value属性可以设置为默认值
- color ： 提供了一个颜色拾取器
- time ： 时分秒
- data ： 日期选择年月日
- datatime ： 时间和日期(目前只有Safari支持)
- datatime-local ： 日期时间控件
- week ： 周控件
- month：月控件

表单属性：

- placeholder ： 提示信息
- autofocus ： 自动获取焦点
- autocomplete="on" 或者 autocomplete="off" 使用这个属性需要有两个前提：
- - 表单必须提交过
  - 必须有name属性。
- required：要求输入框不能为空，必须有值才能够提交。
- pattern=" " 里面写入想要的正则模式，例如手机号patte="^(+86)?\d{10}$"
- multiple：可以选择多个文件或者多个邮箱
- form=" form表单的ID"

表单事件：

- oninput 每当input里的输入框内容发生变化都会触发此事件。
- oninvalid 当验证不通过时触发此事件。

4. 进度条、度量器

- progress标签：用来表示任务的进度（IE、Safari不支持），max用来表示任务的进度，value表示已完成多少
- meter属性：用来显示剩余容量或剩余库存（IE、Safari不支持）
- - high/low：规定被视作高/低的范围
  - max/min：规定最大/小值
  - value：规定当前度量值

设置规则：min < low < high < max

5.DOM查询操作

- document.querySelector()
- document.querySelectorAll()

它们选择的对象可以是标签，可以是类(需要加点)，可以是ID(需要加#)

6. Web存储

HTML5 提供了两种在客户端存储数据的新方法：

- localStorage - 没有时间限制的数据存储
- sessionStorage - 针对一个 session 的数据存储

7. 其他

- 拖放：拖放是一种常见的特性，即抓取对象以后拖到另一个位置。设置元素可拖放：

```
<img draggable="true" />
```

- 画布（canvas ）： canvas 元素使用 JavaScript 在网页上绘制图像。画布是一个矩形区域，可以控制其每一像素。canvas 拥有多种绘制路径、矩形、圆形、字符以及添加图像的方法。

```
<canvas id="myCanvas" width="200" height="100"></canvas>
```

- SVG：SVG 指可伸缩矢量图形，用于定义用于网络的基于矢量的图形，使用 XML 格式定义图形，图像在放大或改变尺寸的情况下其图形质量不会有损失，它是万维网联盟的标准

- 地理定位：Geolocation（地理定位）用于定位用户的位置。‘

总结：

（1）新增语义化标签：nav、header、footer、aside、section、article

（2）音频、视频标签：audio、video

（3）数据存储：localStorage、sessionStorage

（4）canvas（画布）、Geolocation（地理定位）、websocket（通信协议）

（5）input标签新增属性：placeholder、autocomplete、autofocus、required

（6）history API：go、forward、back、pushstate

移除的元素有：

- 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；

7. img的srcset属性的作用？

响应式页面中经常用到根据屏幕密度设置不同的图片。这时就用到了 img 标签的srcset属性。srcset属性用于设置不同屏幕密度下，img 会自动加载不同的图片。用法如下：

```
<img src="image-128.png" srcset="image-256.png 2x" />
```

使用上面的代码，就能实现在屏幕密度为1x的情况下加载image-128.png, 屏幕密度为2x时加载image-256.png。

按照上面的实现，不同的屏幕密度都要设置图片地址，目前的屏幕密度有1x,2x,3x,4x四种，如果每一个图片都设置4张图片，加载就会很慢。所以就有了新的srcset标准。代码如下：

```
<img src="image-128.png"
     srcset="image-128.png 128w, image-256.png 256w, image-512.png 512w"
     sizes="(max-width: 360px) 340px, 128px" />
```

其中srcset指定图片的地址和对应的图片质量。sizes用来设置图片的尺寸零界点。对于 srcset 中的 w 单位，可以理解成图片质量。如果可视区域小于这个质量的值，就可以使用。浏览器会自动选择一个最小的可用图片。

sizes语法如下：

```
sizes="[media query] [length], [media query] [length] ... "
```

sizes就是指默认显示128px, 如果视区宽度大于360px, 则显示340px。

8. 行内元素有哪些？块级元素有哪些？ 空(void)元素有那些？

- 行内元素有：`a b span img input select strong`;
- 块级元素有：`div ul ol li dl dt dd h1 h2 h3 h4 h5 h6 p`;

空元素，即没有内容的HTML元素。空元素是在开始标签中关闭的，也就是空元素没有闭合标签：

- 常见的有：`<br>`、`<hr>`、`<img>`、`<input>`、`<link>`、`<meta>`;
- 鲜见的有：`<area>`、`<base>`、`<col>`、`<colgroup>`、`<command>`、`<embed>`、`<keygen>`、`<param>`、`<source>`、`<track>`、`<wbr>`。

9. 说一下 web worker

在 HTML 页面中，如果在执行脚本时，页面的状态是不可相应的，直到脚本执行完成后，页面才变成可相应。web worker 是运行在后台的 js，独立于其他脚本，不会影响页面的性能。 并且通过 postMessage 将结果回传到主线程。这样在进行复杂操作的时候，就不会阻塞主线程了。

如何创建 web worker：

1. 检测浏览器对于 web worker 的支持性
2. 创建 web worker 文件（js，回传函数等）
3. 创建 web worker 对象

10. HTML5的离线储存怎么使用，它的工作原理是什么

离线存储指的是：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：

（1）创建一个和 html 同名的 manifest 文件，然后在页面头部加入 manifest 属性：

```
<html lang="en" manifest="index.manifest">
```

（2）在 `cache.manifest` 文件中编写需要离线存储的资源：

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

- **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。
- **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。
- **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 。

（3）在离线状态时，操作 `window.applicationCache` 进行离线缓存的操作。

如何更新缓存：

（1）更新 manifest 文件

（2）通过 javascript 操作

（3）清除浏览器缓存

注意事项：

（1）浏览器对缓存数据的容量限制可能不太一样（某些浏览器设置的限制是每个站点 5MB）。

（2）如果 manifest 文件，或者内部列举的某一个文件不能正常下载，整个更新过程都将失败，浏览器继续全部使用老的缓存。

（3）引用 manifest 的 html 必须与 manifest 文件同源，在同一个域下。

（4）FALLBACK 中的资源必须和 manifest 文件同源。

（5）当一个资源被缓存后，该浏览器直接请求这个绝对路径也会访问缓存中的资源。

（6）站点中的其他页面即使没有设置 manifest 属性，请求的资源如果在缓存中也从缓存中访问。

（7）当 manifest 文件发生改变时，资源请求本身也会触发更新。

11. 浏览器是如何对 HTML5 的离线储存资源进行管理和加载？

- **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，如果是第一次访问页面 ，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。如果已经访问过页面并且资源已经进行离线存储了，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，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**，浏览器会直接使用离线存储的资源。

12. title与h1的区别、b与strong的区别、i与em的区别？

- strong标签有语义，是起到加重语气的效果，而b标签是没有的，b标签只是一个简单加粗标签。b标签之间的字符都设为粗体，strong标签加强字符的语气都是通过粗体来实现的，而搜索引擎更侧重strong标签。
- title属性没有明确意义只表示是个标题，H1则表示层次明确的标题，对页面信息的抓取有很大的影响
- **i内容展示为斜体，em表示强调的文本**

13. iframe 有那些优点和缺点？

iframe 元素会创建包含另外一个文档的内联框架（即行内框架）。

优点：

- 用来加载速度较慢的内容（如广告）
- 可以使脚本可以并行下载
- 可以实现跨子域通信

缺点：

- iframe 会阻塞主页面的 onload 事件
- 无法被一些搜索引擎索识别
- 会产生很多页面，不容易管理

14. label 的作用是什么？如何使用？

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

- 使用方法1：

```
<label for="mobile">Number:</label>
<input type="text" id="mobile"/>
```

- 使用方法2：

```
<label>Date:<input type="text"/></label>
```

15. Canvas和SVG的区别

（1）SVG：

SVG可缩放矢量图形（Scalable Vector Graphics）是基于可扩展标记语言XML描述的2D图形的语言，SVG基于XML就意味着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。

其特点如下：

- 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

（2）Canvas：

Canvas是画布，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，就会重新进行绘制。

其特点如下：

- 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

注：矢量图，也称为面向对象的图像或绘图图像，在数学上定义为一系列由线连接的点。矢量文件中的图形元素称为对象。每个对象都是一个自成一体的实体，它具有颜色、形状、轮廓、大小和屏幕位置等属性。

16. head 标签有什么作用，其中什么标签必不可少？

标签用于定义文档的头部，它是所有头部元素的容器。 中的元素可以引用脚本、指示浏览器在哪里找到样式表、提供元信息等。

文档的头部描述了文档的各种属性和信息，包括文档的标题、在 Web 中的位置以及和其他文档的关系等。绝大多数文档头部包含的数据都不会真正作为内容显示给读者。

下面这些标签可用在 head 部分：，，，