# 前端面试指南 — html 篇面试题

# Html目前都只在笔试题中出现，如果没笔试的话，那面试肯定有呀，从html中可以让面试官了解到你的基础及其有没有相关工作经验，这块只能靠背了，因为其实在工作中也很少用到，但面试就偏偏会问道。

# 考点 1：语义化

## 问题：说说你对 html 语义化的理解。

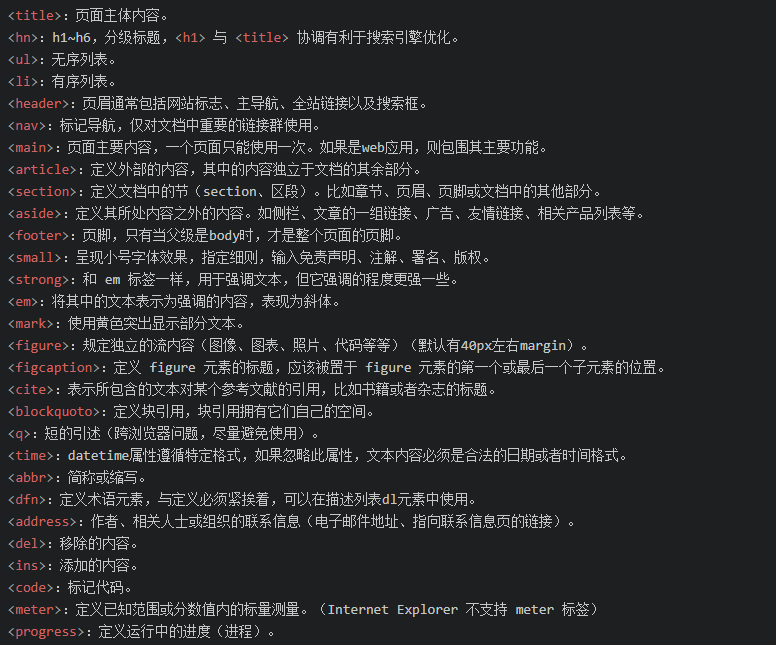
html 语义化是用正确的标签做正确的事情。有三大好处：

结构清晰：html 语义化让页面的内容结构化，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。

SEO：有利于 SEO ，可以让搜索引擎更好地获取到更多有效信息，搜索引擎的爬虫依赖于标签来确定上下文和各个关键字的权重，有效提升网页的搜索量。

可维护性：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。

## 问题：说说 html 语义化的标签有那些。（记住几个容易的，比如header，nav，section，aside，footer等及其作用）



# 考点 2：标签类型

## 问题：请说说你对块级元素、行内元素、空元素的理解，它们分别都有哪些常见的标签？

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。

常见的块级元素标签有：<h1>至<h6>，<div>，<p>，<ul>，<ol>，<li>，<dl>，<dt>，<dd>，

<table>，<article>，<aside>，<audio>，<video>，<footer>，<header>，<nav>， <section>

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； 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。

常见的行内元素标签有：<a>，<span>，<strong> ，<i>，<b>，<button>，<textarea>，<em>，

<label>，<select>

空元素：没有闭合标签的标签被称作为空标签。

常见的空元素标签有：<input /> <img /><hr/> <br>。

# 考点 3：标签对比

## 问题：请说说<strong>和<b>标签的区别。

<strong>标签和<b> 标签都能使得内容有加粗的是视觉效果，区别是：<strong>有重点强调的作用，

<strong>是“含有语义”的标签，搜索引擎会了解这些语义。其在 HTML 中是特意被设定为表示“强调” 的意思。而<b>标签则没有“强调”的含义。

## 问题：请说说<i>和<em>标签的区别。

<i>标签和<em> 标签都能使文本变为斜体，区别是<em>有重点强调的作用，在大多数浏览器里面看起来是斜体，如果单纯为了展示斜体的效果而不加以强调的话，可以使用<i>标签。如果含有强调的意思的话，需要使用<em>标签

# 考点 4：<img>标签对比

## 问题：请你说出<img>标签支持的图片格式，以及它们的区别?



使用技巧：jpg 是有损压缩格式，png 是无损压缩格式。所以，相同的图片，jpg 体积会小。比如我们一些官网的 banner 图，一般都很大，所以适合用 jpg 类型的图片。但 png 分 8 位的和 24 位的，8 位的体积会小很多，但在某些浏览器下 8 位的 png 图片会有锯齿。

## 问题：请你说出<img>中 alt 和 title 属性的区别。

alt 属性：

1. 如果图像没有下载或者加载失败，会用文字来代替图像显示。 这一作用是为了给加载不出网页图片的用户提供图片信息，方便用户浏览网页，也方便程序猿维护网页。
2. 搜索引擎可以通过这个属性的文字来抓取图片。

title 属性：

是当网页上的图片被加载完成后，鼠标移动到上面去，会显示这个图片指定的属性文字，以对图片信息进行补充性说明。

# 考点 5：<iframe>标签

## 问题：为什么要尽量少用<iframe>标签，请你说出<iframe>有哪些缺点？

1. iframe 会阻塞主页面的onload 事件；
2. 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，会产生很多页面，不容易管理。
3. 如果框架个数多的话，可能会出现上下、左右滚动条，会分散访问者的注意力，用户体验度差。
4. 代码复杂，无法被一些搜索引擎索引到，这一点很关键，现在的搜索引擎爬虫还不能很好的处理 iframe 中的内容，所以使用 iframe 会不利于搜索引擎优化（SEO）。
5. 很多的移动设备无法完全显示框架，设备兼容性差。
6. iframe 框架页面会增加服务器的http 请求，对于大型网站是不可取的。

# 考点 6：<label>标签

## 问题：Label 的作用是什么？是怎么用的？

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

<label for="number">号码：</label>

<input type="text" name="number" id="number"/>

# 考点 7：<meta>标签的viewport

## 问题：<mete>标签的 viewport 的作用和原理是什么？

作用：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。

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。

## 问题：viewport 属性值都有哪些？

width：设置 layout viewport 的宽度，为一个正整数，或字符串"width-device"。height 设置 layout viewport 的高度，这个属性对我们并不重要，很少使用。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。

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。

案例：

<meta name="viewport"

content="width=device-width,initial-scale=1.0,minimum-scale=1.0,maximum-scale=1.0,user-sc alable=no">

# 考点 8：HTML5

## 问题：请说说你用过哪些HTML5 的新标签。

1. <article> 用来定义独立的内容
2. <audio> 用来定义声音内容
3. <canvas> 用来定义图形
4. <header> 用来定义页眉
5. <footer> 用来定义页脚
6. <nav>用来 定义导航链接
7. <video> 用来定义视频

## 问题：请说出HTML5 的几个新特性。

绘画 canvas：HTML5 的一个新元素，它使用 JavaScript 在网页上绘制图形。video 和 audio：用于视频和音频的播放。

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。

新的结构标签：语义化更好的内容元素，比如 article，footer，header，nav，section。增强表单：input 的type 属性值新增 calendar,date,time,email,url 等。

新的结构标签：语义化更好的内容元素，比如 article，footer，header，nav，section。增强表单：input 的type 属性值新增 calendar,date,time,email,url 等。

新的技术 webworker，websockt，Geolocation

# 考点 9：增强表单

## 问题：你能说说 HTML5 增强表单中新的输入类型属性吗？

search：用于搜索域，比如站点搜索或 Google 搜索，域显示为常规的文本域。url ：用于应该包含 URL 地址的输入域在提交表单时，会自动验证 url 域的值。

email：用于应该包含 e-mail 地址的输入域，在提交表单时，会自动验证 email 域的值。datetime：选取时间、日、月、年（UTC 时间）。

date：选取日、月、年 month：选取月、年 。week：选取周和年。

time：选取时间（小时和分钟）。

datetime-local：选取时间、日、月、年（本地时间）。

number：用于应该包含数值的输入域，您还能够设定对所接受的数字的限定。range：用于应该包含一定范围内数字值的输入域，类型显示为滑动条。

# 考点 10：本地存储

## 问题： cookie、localstroage、sessionStorage 的区别？优缺点？

cookie：是服务器发给客户端的特殊信息，以文本形式存储在客户端，每次请求都会带上 cookie。

cookie 的保存时间：设置过期时间，浏览器关闭后不会清除，保存在硬盘中, 过期时间到期后失效。如果不设置过期时间，保存在内存中, 浏览器关闭后消失。

缺点：

1. 大小受限，单个 cookie 大小不能超过 4kb
2. 用户可以禁用 cookie, 使功能受限。
3. 安全性较低，有些状态不能保存在客户端。
4. 每次访问都要传送 cookie 给服务器，浪费带宽。
5. cookie 数据有路径（path）的概念，可以限制 cookie 只属于某个路径下。

localStorage 和 sessionStorage 存储大小都是 5MB，都保存在客户端不与服务器端进行交互，只能储存字符串类型，对于复杂的 json 格式可以进行 stringify 和 parse 来处理。区别是 localStorage 是永久储存, 除非主动删除, 否则不会消失；而 sessionStroage 的有效期只是网页在浏览器打开到关闭的时间段。

# 考点 10：其他问题

## 问题：前端页面有哪三层构成，分别是什么？作用是什么？

结构层、表示层、行为层。

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。”

结构层为页面的骨架，由 HTML 或 XHTML 标记语言创建，用于搭建文档的结构。

表示层为页面的样式，由 CSS （层叠样式表）负责创建，用于设置文档的呈现效果。

行为层（behaviorlayer）为页面的行为，由 JavaScript 语言创建，用于实现文档的行为

## 问题：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

## 问题：CSS引入的方式有哪些? link和@import的区别是?

内联 内嵌 外链 导入

区别 ：同时加载,前者无兼容性，后者CSS2.1以下浏览器不支持,Link 支持使用javascript改变样式，后者不可

## 问题：介绍一下你对浏览器内核的理解

内联 内嵌 外链 导入

主要分成两部分：渲染引擎(layout engineer或Rendering Engine)和JS引擎。

渲染引擎：负责取得网页的内容（HTML、XML、图像等等）、整理讯息（例如加入CSS等），以及计算网页的显示方式，然后会输出至显示器或打印机。浏览器的内核的不同对于网页的语法解释会有不同，所以渲染的效果也不相同。所有网页浏览器、电子邮件客户端以及其它需要编辑、显示网络内容的应用程序都需要内核。

JS引擎则：解析和执行javascript来实现网页的动态效果。

最开始渲染引擎和JS引擎并没有区分的很明确，后来JS引擎越来越独立，内核就倾向于只指渲染引擎。

## 问题：script标签中defer和async的区别

defer 浏览器指示脚本在文档被解析后执行，script被异步加载后并不会立即执行，而是等待文档被解析完毕后执行

<script type="text/javascript" src="x.min.js" defer="defer"></script>

defer只适用于外联脚本，如果script标签没有指定src属性，只是内联脚本，不要使用defer

如果有多个声明了defer的脚本，则会按顺序下载和执行

defer脚本会在DOMContentLoaded和load事件之前执行

async 同样是异步加载脚本，区别是脚本加载完毕后立即执行，这导致async属性下的脚本是乱序的，对于script又先后依赖关系的情况，并不适用

<script type="text/javascript" src="x.min.js" async="async"></script>

只适用于外联脚本，这一点和defer一致

如果有多个声明了async的脚本，其下载和执行也是异步的，不能确保彼此的先后顺序

async会在load事件之前执行，但并不能确保与DOMContentLoaded的执行先后顺序

## 问题：什么是防抖和节流？有什么区别？

防抖：触发高频事件后n秒内函数只会执行一次，如果n秒内高频事件再次被触发，则重新计算时间（思路：每次触发事件时都取消之前的延时调用方法）

节流：高频事件触发，但在n秒内只会执行一次，所以节流会稀释函数的执行频率（思路：每次触发事件时都判断当前是否有等待执行的延时函数）

节流场景：监听滚动事件，比如是否滑到底部自动加载更多，用throttle来判断

防抖场景：search搜索联想，用户在不断输入值时，用防抖来节约请求资源

## 问题：html标签中的src和href有什么区别？

href是超文本引用，它是指向资源的位置，建立与目标文件的联系

src目的是把资源下载到页面中

浏览器解析href不会阻塞对文档的处理（这就是官方建议使用link引入而不是@import的原因），src会阻塞对文档的处理。

## 问题：html页面渲染过程？

1. 解析html文件，创建dom树
2. 解析css，形成css对象模型
3. 将css与DOM合并，构建渲染树
4. 布局和绘制

重绘：重绘是改变不影响元素的网页中的位置的元素样式（重绘不会带来重新布局，所以并不一定伴随重排）

重排：渲染绘制，即根据计算好的信息绘制整个页面，渲染出最终的页面。

重绘不一定需要重排，重排必然导致重绘