**Web前端面试题目**

1. **HTML5+CSS3新特性?**

**HTML5是HTML的新标准，它的推出是为了解决HTML4时代的兼容性问题、优化代码结构、扩展Web应用程序功能，用来构建更好的Web项目。**

**HTML5新特性：**

1. **W3C制定了新的HTML标准，并使得各浏览器都统一遵守新的HTML5标准，解决了HTML诞生到HTML4时代的兼容性问题。**
2. **语义化标签，使得页面的内容结构化**

**新增了一些语义标签，比如header、footer、nav、section、article、aside、details、summary、dialog标签等**

1. **增强型表单标签，扩展Web程序的功能**

**新增了一些标签，主要用来构建负责的Web应用，比如**

**time、date、datetime、month、week、**

**range、number、**

**search、**

**tel、email、url**

1. **新增表单属性，扩展Web程序的功能，比如**

**placeholder：输入数据前，输入框默认提示**

**boolean：true/false，输入框内容是否为空**

**autofocus & multiple：指定某元素自动获得焦点，输入域支持选择多个值**

**min & max：设置元素的最小值和最大值**

**pattern：正则表达式验证input元素的值是否符合要求**

**data-\*：新增了标签的数据属性，使得用户可以用自定义的方式存储数据**

**contentEditable：true/false，规定了当前元素是否可以编辑**

**draggable：Enum(true/false/auto)，规定了当前元素是否可以拖拽**

1. **网页多媒体和图形特性audio & video & (canvas/svg)**

**新增了audio video canvas/svg标签，用来在页面上渲染音视频以及canvas/svg绘制图形**

1. **本地存储特性sessionStorage/localStorage**

**这两者对数据的存储都是以键值对的形式，容量限制一致都为5兆，且都保存在客户端（浏览器）中。localStorage的生命周期是永久，意味着除非是主动清除localStorage信息，否则这些信息将永远存在。sessionStorage信息仅在当前会话下有效，关闭当前页面或者清除浏览器数据，即失效。**

1. **拖放API drag&drop**

**HTML5新增加的拖拽API**

1. **设置被拖拽的对象属性draggable:true**
2. **拖拽API一共有8个监听事件，3个在被拖拽对象上，5个在拖拽目标上**
3. **在具体的监听事件中，从dataTransfer获取业务数据并且处理具体的业务逻辑**
4. **地理定位 geolocation**

**HTML5新增地理位置信息：浏览器本地通过WIFI、GPS、IP地址、蓝牙MAC地址、RFID、GSM/CDMA等**

1. **在获取地理位置信息之前，浏览器会向用户提示是否共享位置信息**
2. **在代码中，获取window.navigator对象，并通过getCurrentPosition（successCallback,errorCallback,configObject）函数获得地理位置信息**
3. **在successCallback函数里面实现对地理位置信息的获取**
4. **WebSocket API**

**HTML5新增了WebSocket API来支持WebSocket，WebSocket是基于TCP协议的一种全双工网路协议。全双工网络协议指的是，客户端服务端只建立一次链接，便可以互相通信。WebSocket的优点，避免了系统使用AJAX轮询或者长轮询Long polling，更好的节省了带宽。**

**Web Socket多用于实时系统中，比如直播、社交、视频会议、基于位置的在线服务、股票基金等交易系统、协同编辑系统等**

**使用方式：**

1. [**https://juejin.cn/post/6844903779251257358**](https://juejin.cn/post/6844903779251257358)
2. [**https://wangdoc.com/webapi/websocket.html**](https://wangdoc.com/webapi/websocket.html)
3. [**https://segmentfault.com/a/1190000021183700**](https://segmentfault.com/a/1190000021183700)
4. **Web Workers**

**HTML5新增了Web Workers，Web Workers可以在Javascript单线程执行的基础上，开启子线程并行执行其他任务，执行处理器密集型任务的情况下不会阻碍UI线程。**

**Web Wokers提供了一个独立于浏览器主线程的context上下文环境，在当前worker线程中，只能使用受限的API。比如，不能操作DOM，不能访问Cache。**

**当前worker如何与主线程通信:双方都使用postMessage()发送各自的消息，使用onmessage事件处理函数响应消息。这个过程中，数据被复制传递而不是共享。**

**关于Web Workers**

1. [**https://developer.mozilla.org/zh-CN/docs/Web/API/Web\_Workers\_API/Using\_web\_workers**](https://developer.mozilla.org/zh-CN/docs/Web/API/Web_Workers_API/Using_web_workers)
2. [**https://segmentfault.com/a/1190000016756466**](https://segmentfault.com/a/1190000016756466)
3. [**https://segmentfault.com/a/1190000012528806**](https://segmentfault.com/a/1190000012528806)

**Web Workers的使用场景：音频/视频编码解码、图片预处理、排序算法、客户端模板**

**比如VS Code在线版就使用了Web Workers**

[**https://segmentfault.com/a/1190000016638898**](https://segmentfault.com/a/1190000016638898)

**CSS3新特性：**

1. **选择器：**

**新增了结构伪类选择器，比如E:frist-child、E:nth-child(n)、E:last-child，分别为父元素的第一个、第n个、最后一个子元素。**

**新增了属性选择器，E[attr=”value”] 指定属性名，并指定其对应属性值。**

1. **边框属性：**
2. **border-raduis：边框圆角**
3. **border-image：边框图片**
4. **box-shadow：边框阴影**
5. **文本属性：**
6. **text-shadow：文本阴影**
7. **text-overflow：文字溢出**
8. **white-space：控制空白字符的显示，已经是否换行**
9. **text-wrap：文本换行**
10. **word-wrap：控制长度超过一行的单词如何换行**
11. **word-break：控制单词（连续的字符）如何被拆分换行**

[**https://www.cnblogs.com/dfyg-xiaoxiao/p/9640422.html**](https://www.cnblogs.com/dfyg-xiaoxiao/p/9640422.html)

[**https://juejin.cn/post/6844903667863126030**](https://juejin.cn/post/6844903667863126030)

[**https://codepen.io/YGYOOO/pen/jvyrWK**](https://codepen.io/YGYOOO/pen/jvyrWK)

1. **渐变**
2. **线性渐变**

**从一个方向到另一个方向颜色的逐渐变化，设置好方向和颜色，也可以设置分段渐变色。**

**background:linear-gradient(direction, color-stop1, color-stop2, ...);**

**background:linear-gradient(red 10%,blue 20%,green 30%,yellow 40%);**

1. **径向渐变**

**从起点到终点，颜色从内向外圆形渐变**

**background: radial-gradient(circle farthest-side at 0 0,#f00 0%,#ff0 100%);**

1. **重复渐变**

**分为 线性重复渐变 和径向重复渐变**

1. **Flex布局**

**Flex布局是Flexible Box的缩写，对父级元素设置display:flex；在flex布局父级元素上对子元素设置flex-direction、justify-content、align-items、flex-wrap、flex-flow（flex-direction 和 flex-wrap）、align-content。**

**在flex子元素上设置flex(flex-grow、flex-shrink、flex-basis)、order、align-self。分别设置子元素放大缩小、排序、垂直主轴位置**

1. **flex-direction：**

**设置flex布局的主轴方向，即是子元素排列的方向。**

**有以下几个选项：**

1. **ROW：子元素按行排列，从左到右**
2. **ROW-REVERSE：子元素按行排列，从右到做**
3. **COLUMN：子元素按列排列，从上到下**
4. **COLUMN-REVERSE：子元素案列排列，从下到上**
5. **justify-content：**

**justify-content决定了内容元素沿着主轴之间的相对位置。**

**包含了flex-start、flex-end、center、space-around、space-between。**

1. **align-items：**

**align-items和justify-content决定的方向不一样，是主轴**

**相垂直的的方向。包含了flex-start、flex-end、center、stretch、base-line。**

1. **align-content：**

**align-content对多列情况下的flex布局，对整体的列进行**

**相对的布局。包含了flex-start、flex-end、center、stretch、space-around、space-between**

1. **flex-wrap：**

**flex-wrap 决定了flex布局时，当子元素过多，是否换行。**

**这个属性包含了三个值，no-wrap、wrap、wrap-reverse。**

1. **flex：**

**flex是flex-grow、flex-shrink、flex-basis的合称。**

[**http://www.ruanyifeng.com/blog/2015/07/flex-grammar.html**](http://www.ruanyifeng.com/blog/2015/07/flex-grammar.html)

[**https://zhuanlan.zhihu.com/p/25303493**](https://zhuanlan.zhihu.com/p/25303493)

1. **媒体查询**

**媒体查询的引入，其作用就是允许添加表达式用以确定媒体的环境情况，以此来应用不同的样式表。换句话说，其允许我们在不改变内容的情况下，改变页面的布局以精确适应不同的设备。**

**一般情况下：**

**@media media-type and (media-feature-rule) {**

**/\* CSS rules go here \*/**

**}**

**媒体类型media-type:[all、print、screen、speech]**

**媒体特征media-feature-rule：**

**一般包含：**

1. **aspect-ratio（宽高比）**
2. **device-aspect-ratio（设备宽高比）**
3. **device-height/device-width（设备宽度/设备高度）**
4. **orientation（方向）**

**逻辑操作符：and、not、only、（，）**

**举例:**

**@media screen and (max-width: 300px) {**

**body { background-color:lightblue; }**

**}**

1. **转换Transform**

**CSS3 转换可以对元素进行移动、缩放、转动、拉长或拉伸。**

**Transform包含了集中不同的设置**

1. **2D转换**
   1. **rotate（xx deg）：顺时针转换xx度**
   2. **translate（xx px, yypx）：相对左上顶点在x轴上移动了xx px，在y轴上移动了yy px。**
   3. **scale（x，y）：在x轴方向放大了x倍，在y轴方向放大了y倍**
   4. **skew（x,y）：在x轴倾斜了x度，在y轴倾斜了y 度**
2. **3D转换**
   1. **rotateX/ rotateY（xx deg）：顺时针沿轴旋转xx度**
   2. **translateX/ translateY/ translateZ（xx px）**
   3. **scaleX/ scaleY/ scaleZ（xx px）**
   4. **rotate3d/** **translate3d/** **scale3d/** **matrix3d**
3. **过渡**

**CSS3 Transition可以使得元素从一种样式改变为另一种样式，要实现这种效果，必须规定两种内容**

1. **对应样式的CSS属性**
2. **该样式的持续时间**

**主要由以下属性来实现：transition-property、transition-duration、transition-timing-function、transition-delay。或者使用单独的transition属性也可以包含上面全部属性。**

**Transition:[property]，[duration]，[timing-function]，[delay]**

**div { width:100px;transition: width 1s linear 2s; }**

**div:hover{width:300px;}**

**上述效果实现为：hover在此div上，延时两秒，在接下来的一秒中，宽度从100px以liner方式增加到300px**

1. **动画@keyframes**

**通过@keyframes定义动画，并且在指定元素中指定animation属性来使用对应的动画，动画就会生效**

**Keyframes定义规则：**

**@keyframes name{**

**0% { //css property },**

**20% { //css property }**

**………**

**}**

**Animation使用规则：**

**Animation:[name] [duration] [timing-function] [delay] [iteration-count] [direction] [play-state]，依次规定了动画的名称、执行时间、以xx动画曲线执行、延迟x秒执行、是否在下一周期逆向、是否暂停动画。**

**animation: name 5s linear 2s infinite alternate;**

**上述效果实现为：延迟2秒执行name动画，在5秒内的过渡时间内以liner方式变化，并且反向执行动画**

1. **什么是HTML的语义化，有什么好处？**

**HTML的语义化就是用语义化的标签取代原来的DIV+SPAN布局。**

1. **有助于构建良好结构的前端代码，便于团队的开发和维护**
2. **在去掉或者丢失样式的时候，能够让页面呈现清晰的结构**
3. **有助于构建良好的html结构，有利于SEO。主要体现在方便搜索引擎（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）抓取网站内容建立索引，并且根据关键字权重确定网站排名**
4. **方便其他设备对网页的解析(比如屏幕阅读器)，利于无障碍阅读，提高可访问性**

[**https://juejin.cn/post/6844903527987281934**](https://juejin.cn/post/6844903527987281934)

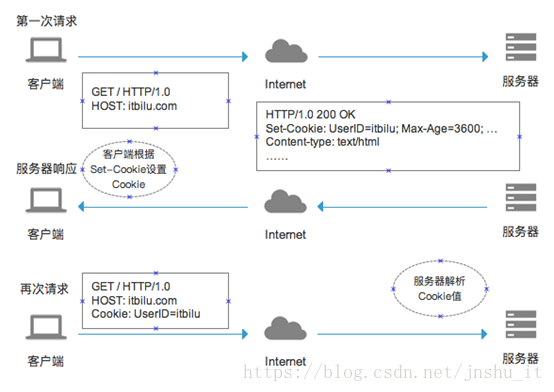
[**https://juejin.cn/post/6876826035946684424**](https://juejin.cn/post/6876826035946684424)

1. **Cookie、Session、Token的区别**

**这些区别需要从网络协议的状态开始讲起。在Web1.0时代，HTTP协议（超文本传输协议）设计的目的是为了传输静态HTML页面，在设计之初就没有考虑好Web动态功能的需求，比如交互认证保存会话信息等，这也是HTTP是无状态协议的由来。HTTP协议的无状态性主要体现在，每一个HTTP请求都是相互独立，每个请求包含了处理请求的全部数据。**

**Cookie&Session技术的出现是为了支持客户端与服务端的交互，是对HTTP协议无状态性的补充。Cookie&Session技术的实现一般是**

1. **首次请求，由服务端调用**request.getSession()**获取Session实例**
2. **由服务端返回一个Set-Cookie 响应标头，浏览器接受之后自动存储到Cookie中**
3. **浏览器获取Cookie信息并保存在本地文件中，下次请求同一网站便蒋Cookie信息设置在HTTP协议请求头中**



**具体来说，有以下区别**

1. **session 在服务器端，cookie 在客户端（浏览器）**
2. **session 默认被存在在服务器的一个文件里（不是内存）**
3. **单个Cookie保存的数据不能超过4K，很多浏览器都限制一个站点最多保存20个Cookie**
4. **session 的运行依赖 session id，而 session id 是存在 cookie 中的，也就是说，如果浏览器禁用了 cookie ，同时 session 也会失效（但是可以通过其它方式实现，比如在 url 中传递 session\_id）**
5. **session 可以放在 文件、数据库、或内存中都可以**

**Token技术**

1. **Cookie、sessionStorage、localStorage的区别**

