# HTML20230614

第一次50题 70%正确率

1. SEO、SVG和canvas是什么
   1. **SEO**（Search Engine Optimization）：SEO 是指搜索引擎优化，它是一系列的技术和策略，旨在提升网站在搜索引擎中的排名，从而增加网站的曝光和流量。通过优化网站的内容、结构、链接等方面，使搜索引擎更容易理解和索引网站的内容，并将其排名靠前，使用户能够更容易地找到相关的网站。
   2. **SVG**（Scalable Vector Graphics）：SVG 是一种用于描述二维矢量图形的 XML 标记语言。与传统的栅格图像（如 JPEG、PNG）不同，SVG 使用矢量图形来表示图像，它基于数学方程描述图形的形状、颜色和样式，因此可以无损地缩放和放大而不失真。SVG 可以通过文本编辑器进行创建和编辑，并可以通过在 HTML 中嵌入或作为独立文件使用来显示图形。
   3. **Canvas**：Canvas 是 HTML5 提供的一个用于绘制图形的元素。它通过 JavaScript 提供了一组 API，可以在网页上动态地绘制图形、图像和动画。Canvas 提供了绘制路径、图形填充、线条样式、渐变、图像处理等功能，使开发者能够以程序化的方式创建各种图形效果和交互式图形应用。相比于 SVG，Canvas 绘制的图形是基于像素的位图，因此在缩放时会有一定程度的失真。
2. SVG和Canvas
   1. 使用**<svg>**标签创建一个画布，然后使用**<rect>**标签创建一个矩形，并设置其位置、大小和填充颜色
   2. rect中的x和y表示元素坐标，fill表示颜色
   3. 使用**<canvas>**标签创建一个画布，并通过JavaScript获取上下文对象**ctx**，然后使用**fillStyle**属性设置填充颜色，最后使用**fillRect()**方法绘制矩形。
   4. canvas标签通过id获取，canvas.getContext(‘2d’)获取上下文对象，fillRect()方法分别是左上坐标和宽高
   5. 经过将网页方法，确实canvas有失真，且不可修改
   6. canvas通过js生成，不能被引擎抓取；svg通过标签实现，可以通过引擎抓取，支持事件绑定
3. 如何优化SEO效果
   1. 静态资源对SEO友好，js动态生成网页title、descriptionhe keyword适得其反
   2. 关键词优化：研究并确定与你的网站内容相关的关键词，并在网站的标题、元标签、URL、正文和图片描述等位置合理地使用这些关键词。
   3. 内容质量：创建高质量、有价值的内容，包括文章、博客、视频、图像等，以吸引用户并增加网站的可信度和权威性。
   4. 网站结构优化：确保网站的结构清晰、简洁，并使用合适的标题标签（如 H1、H2）进行内容组织，以便搜索引擎更好地理解和索引你的网站。
   5. 友好的 URL 结构：使用简洁、描述性的 URL，包含相关关键词，同时避免使用过长或含有无意义字符的 URL。
   6. 内部链接优化：在网站的不同页面之间建立合适的内部链接，以便搜索引擎能够更好地发现和索引你的页面内容。
   7. 外部链接建设：争取其他高质量网站的链接指向你的网站，这有助于提升你网站的可信度和排名。
   8. 网站速度优化：优化网站的加载速度，包括优化图像大小、压缩文件、减少 HTTP 请求等，以提升用户体验和搜索引擎对网站的评价。
   9. 移动优化：确保你的网站在移动设备上有良好的用户体验，因为搜索引擎越来越重视移动友好的网站。
   10. 社交媒体整合：利用社交媒体平台扩大你网站的影响力，增加网站的曝光和流量。
   11. 持续监测和改进：定期监测你的网站的 SEO 效果，了解用户行为和搜索引擎的变化，并根据需求进行优化和改进。
4. 元标签
   1. 元标签（Meta Tags）是HTML文档中的一种特殊标签，用于提供关于文档的元数据（metadata）。元数据是描述数据的数据，它提供关于网页内容的信息，但并不直接显示在页面上。
   2. 元标签以**<meta>**标签的形式出现在HTML文档的**<head>**标签内部。它们通常用于指定网页的编码字符集、关键词、描述、作者、视口设置等。
   3. 常见如字符集、关键词、描述、视口设置等
5. form表单readonly设置位true和disabled设置为true
   1. readonly界面上无法修改，但是js可以对其修改，内容会被提交
   2. disabled可以锁定空间，用户无法改变值，而且完全无法使用，包括背景颜色，checkbox无法选中，内容不被提交
   3. 所有控件都有disabled，但不一定有readonly，如select下拉框
   4. disabled 属性：当将一个表单元素的 disabled 属性设置为 true 时，该元素会被禁用，用户无法与之交互，也无法通过表单提交该元素的值。禁用的表单元素在外观上通常会有一种灰色或禁用状态的样式，具体样式由浏览器决定。禁用的表单元素不会被包括在表单的提交数据中。
   5. readonly 属性：当将一个表单元素的 readonly 属性设置为 true 时，该元素仍然可以显示内容，但用户无法编辑或修改它。只读的表单元素仍然可以被选中、复制和提交，但用户无法直接修改其值。只读的表单元素在外观上通常与可编辑元素相似，可能没有明显的视觉差异。
6. a标签的target属性默认值
   1. \_self，默认在当前窗口打开
7. key标签、abbr标签、search标签和bdo标签
   1. <key>标签是HTML5中的一个标签，用于表示键盘或其他输入设备的按键。它通常与其他标签结合使用，用于定义按键的功能或快捷键。例如：
      1. <key key="Enter" action="submit">提交</key>
   2. <abbr>标签用于表示缩写或首字母缩写。它可以帮助读者理解并识别文本中的缩写内容。可以使用title属性提供完整的解释或全称。例如：
      1. <p>HTML是<abbr title="HyperText Markup Language">HTML</abbr>的缩写。</p>
   3. <search>标签是HTML5中的一个标签，用于定义搜索输入字段。它通常与<input>标签结合使用，并使用type="search"来创建一个用于搜索的文本输入框。例如：
      1. <search> <input type="search" name="search" placeholder="搜索..."> <button type="submit">搜索</button> </search>
   4. <bdo>标签用于覆盖文本的默认方向，即从左到右（LTR）或从右到左（RTL）。它可以用于调整特定文本块或短语的阅读方向。dir属性用于指定方向，可取值为ltr（从左到右）或rtl（从右到左）。例如：
      1. <bdo dir="rtl">这段文本将从右到左显示。</bdo>
8. iframe标签用于指定内联框架
   1. **<iframe>**标签用于在网页中嵌入另一个文档或网页。它可以用于显示来自同一域或其他域的内容，提供了一种将外部资源嵌入到当前页面的方法。
   2. **src** 属性指定了要嵌入的外部网页的 URL。当页面加载时，浏览器将会在 **<iframe>** 标签所在的位置显示嵌入的网页内容。
   3. <iframe src="https://www.example.com" width="500" height="300" frameborder="0" scrolling="no"></iframe>
   4. 可以通过设置 **<iframe>** 标签的各种属性来控制其行为和样式。例如，可以设置宽度和高度、边框样式、是否显示滚动条等。
   5. 通过 **<iframe>** 标签，您可以在当前页面中嵌入其他网页、嵌入媒体内容（如视频、音频）、显示地图、加载广告等等。请注意，由于安全性和隐私方面的考虑，浏览器可能会对 **<iframe>** 的某些功能进行限制。
9. 什么是w3c
   1. W3C（World Wide Web Consortium）是国际互联网标准组织，致力于制定和推广互联网技术的标准。W3C的目标是确保Web的长期发展，并促进Web的互操作性和可持续性。
   2. W3C制定了一系列的技术规范和标准，其中最著名的是HTML、CSS和JavaScript。这些标准定义了Web页面的结构（HTML）、样式（CSS）和行为（JavaScript），使得不同的浏览器和设备能够一致地解释和呈现网页内容。
   3. W3C的标准制定过程是公开透明的，参与者包括来自各个组织和公司的技术专家、开发者和用户。W3C的标准制定过程经历了草案、候选推荐、推荐等阶段，确保了标准的质量和可靠性。
   4. 通过遵循W3C的标准，开发者可以创建符合标准的网页和应用程序，提高跨平台和跨浏览器的兼容性，同时也为用户提供更好的体验和访问性。
   5. 总之，W3C的标准是Web技术的基石，通过制定统一的规范，推动Web的发展和进步，促进互联网的普及和使用。w3c标准
10. cite标签、hr标签、address标签
    1. **<cite>**标签用于表示引用或参考的作品、书籍、文章或其他来源的标题。它可以用于标记出文本中的引用部分，使其在视觉上与其他文本有所区别。
    2. **<hr>**标签用于创建水平分隔线，它在文档中插入一条水平线来分隔内容。它通常用于分隔不同的主题、章节或段落。**<hr>** 标签是一个自闭合标签，不需要闭合标签。
    3. **<address>**标签用于表示联系信息，通常用于标记文档或网页中的作者、拥有者或相关组织的联系信息。它可以包含文本、链接和其他标记。浏览器通常以斜体显示 **<address>** 元素的内容。
11. H5新引入的特性
    1. 语义化标签：HTML5引入了一系列的语义化标签，如**<header>、<nav>、<section>、<article>、<footer>**等，以更清晰地描述网页结构和内容，提高可读性和可访问性。
    2. 增强型表单控件：HTML5提供了一些新的表单控件，如**<input type="date">、<input type="email">、<input type="number">**等，使得表单输入更加方便和准确。
    3. 新增表单属性：placeholder、multiple、required、pattern
    4. 新增表单类型：date、time、datetime-loca、week、month、numbe、range、color、search、email、url、tel、file、textarea的maxlength属性、datalist
    5. 视频和音频支持：HTML5引入了**<video>和<audio>标签**，使得在网页上嵌入和播放视频和音频内容更加简单和灵活。
    6. 画布（Canvas）：HTML5的**<canvas>**标签允许通过JavaScript在网页上绘制图形、动画和游戏等交互式内容，为开发者提供了强大的绘图功能。
    7. 本地存储：HTML5引入了本地存储机制，包括**Web Storage和IndexedDB**，使得网页可以在客户端存储和访问数据，减少对服务器的依赖，提高性能和用户体验。
    8. 地理位置定位：HTML5提供了**Geolocation API**，可以通过浏览器获取用户的地理位置信息，为基于位置的应用和服务提供支持。
    9. **Web Workers**：HTML5引入了Web Workers，允许在后台运行独立的JavaScript线程，提高网页的响应性能和并行处理能力。
    10. **WebSockets**：HTML5的WebSockets提供了一种实时通信的机制，使得浏览器和服务器之间可以进行双向的实时数据传输。
    11. **响应式设计**：HTML5鼓励使用响应式设计来创建适应不同设备和屏幕大小的网页，使得网页在不同设备上展现良好的用户体验。
12. ES6、ES7、ES8、ES9、ES10、ES11分别引入了什么
    1. ES6（ECMAScript 2015）：也称为ES2015，是JavaScript的重要更新版本。它引入了许多新的语言特性，如箭头函数、模板字面量、解构赋值、类和模块等。
    2. ES7（ECMAScript 2016）：也称为ES2016，是对ES6的一些小规范补充。它引入了一些新的特性，如指数运算符（\*\*）和Array.prototype.includes()方法。
    3. ES8（ECMAScript 2017）：也称为ES2017，引入了一些有用的新功能，如异步函数（async/await）、共享内存和原子操作等。
    4. ES9（ECMAScript 2018）：也称为ES2018，引入了一些新的功能，如正则表达式的命名捕获组、Promise.finally()方法、Object.fromEntries()方法等。
    5. ES10（ECMAScript 2019）：也称为ES2019，引入了一些新的功能，如数组的扁平化方法（Array.prototype.flat()和Array.prototype.flatMap()）、可选链操作符（?.）等。
    6. ES11（ECMAScript 2020）：也称为ES2020，引入了一些新的功能，如可选链操作符的升级（?.()）、全局对象的新方法（globalThis）、动态导入（import()）等。
13. SVG是H5引入的吗
    1. SVG（可缩放矢量图形）并不是H5引入的，它是一种基于**XML**的图形格式，用于在Web上绘制矢量图形。SVG最初于2001年发布，它独立于HTML和CSS，并且可以在任何支持XML的环境中使用。
    2. 虽然SVG可以与HTML5一起使用，并且HTML5在处理SVG方面提供了更好的支持，但SVG本身并不是HTML5引入的新特性。HTML5引入了一些新的元素（如<svg>）和API（如SVG DOM API）来更好地集成和操作SVG图形，以实现在HTML页面中绘制矢量图形的能力。
    3. 因此，可以说SVG是HTML5中使用的一种标准的图形格式，但SVG本身并不是HTML5引入的。它在Web开发中独立存在，并且可以在各种环境中使用。
    4. SVG使用XML格式定义图形，是万维网联盟的标准，与DOM和XSLzhilei的W3C标准是一个整体，是HTML下的一个分支
14. XML、JSON和HTML5
    1. JSON比XML在数据编码上JSON更具有效率，更节约空间
    2. XML对数据的类型描述表达比JSON更丰富
    3. 存在一些数据库，支持XML或JSON数据的操作，如mongodb，postgresql
    4. XML并不比JSON更能表达结构化数据
15. 继承属性
    1. 一、**无继承性**的属性（display、margin、border、padding、background、height、width、position）
       1. **display**：规定元素应该生成的框的类型
       2. **文本属性**：
          1. vertical-align：垂直文本对齐
          2. text-decoration：规定添加到文本的装饰
          3. text-shadow：文本阴影效果
          4. white-space：空白符的处理
          5. unicode-bidi：设置文本的方向
       3. **盒子模型的属性**：width、height、margin 、margin-top、margin-right、margin-bottom、margin-left、border、border-style、border-top-style、border-right-style、border-bottom-style、border-left-style、border-width、border-top-width、border-right-right、border-bottom-width、border-left-width、border-color、border-top-color、border-right-color、border-bottom-color、border-left-color、border-top、border-right、border-bottom、border-left、padding、padding-top、padding-right、padding-bottom、padding-left
       4. **背景属性**：background、background-color、background-image、background-repeat、background-position、background-attachment
       5. **定位属性**：float、clear、position、top、right、bottom、left、min-width、min-height、max-width、max-height、overflow、clip、z-index
       6. **生成内容属性**：content、counter-reset、counter-increment
       7. **轮廓样式属性**：outline-style、outline-width、outline-color、outline
       8. **页面样式属性**：size、page-break-before、page-break-after
       9. **声音样式属性**：pause-before、pause-after、pause、cue-before、cue-after、cue、play-during
    2. 二、有继承性的属性
       1. 字体系列属性：font、font-family、font-weight、font-size、font-style、font-variant、font-stretch、font-size-adjust
          1. font-variant：设置小型大写字母的字体显示文本，这意味着所有的小写字母均会被转换为大写，但是所有使用小型大写字体的字母与其余文本相比，其字体尺寸更小。
          2. font-stretch：对当前的 font-family 进行伸缩变形。所有主流浏览器都不支持。
          3. font-size-adjust：为某个元素规定一个 aspect 值，这样就可以保持首选字体的 x-height。
       2. 文本系列属性：text-indent、text-align、text-height、word-spacing、letter-spacing、text-transform、direction、color
          1. word-spacing：增加或减少单词间的空白（即字间隔）
          2. text-transform：控制文本大小写
       3. 元素可见性：visibility
       4. 表格布局属性：caption-side、border-collapse、border-spacing、empty-cells、table-layout
       5. 列表布局属性：list-style-type、list-style-image、list-style-position、list-style
       6. 生成内容属性：quotes
       7. 光标属性：cursor
       8. 页面样式属性：page、page-break-inside、windows、orphans
       9. 声音样式属性：speak、speak-punctuation、speak-numeral、speak-header、speech-rate、volume、voice-family、pitch、pitch-range、stress、richness、、azimuth、elevation
    3. 三、所有元素可以继承的属性
       1. 元素可见性：visibility
       2. 光标属性：cursor
    4. 四、内联元素可以继承的属性
       1. 字体系列属性
       2. 除text-indent、text-align之外的文本系列属性
    5. 五、块级元素可以继承的属性
       1. text-indent、text-align
16. date类别和datetime-local类别
    1. date类型用于选择日期，它允许用户选择年、月、日的值。用户可以通过日历选择器或手动输入日期。
    2. 兼容性较好
    3. datetime-local类型用于选择日期和时间，它允许用户选择年、月、日、小时和分钟的值。用户可以通过日历选择器或手动输入日期和时间。
17. blockquote标签、marquee标签
    1. **<blockquote>** 标签用于引用文本块，表示被引用的内容来自于其他来源。通常情况下，**<blockquote>** 内的文本会有一定的缩进或者其他样式来与其他内容区分开来。
    2. **<marquee>** 标签用于创建滚动的文本或图像效果。然而，需要注意的是，**<marquee>** 标签已经被 HTML5 废弃，不再是标准的一部分。使用该标签可能会导致兼容性和可访问性问题。建议使用 CSS 或 JavaScript 来实现滚动效果。
18. 自关闭的斜线（H5）
    1. 指的是在空元素上使用斜线 **/** 来表示该元素是自关闭的，而不需要闭合标签。
19. 提高dom元素效率
    1. 减少 DOM 操作的次数：每次进行 DOM 操作都会触发浏览器的重新渲染，因此减少 DOM 操作的次数可以提高效率。可以通过将多个操作合并为一个操作、使用文档片段（DocumentFragment）进行**离线操作**等方式来减少 DOM 操作。
    2. 缓存 DOM 查询结果：多次查询相同的 DOM 元素会带来性能损耗，可以将查询结果缓存起来以避免重复查询。例如，将查询结果存储在变量中，而不是每次需要时都重新查询。
    3. 批量更新样式：修改元素样式时，最好将多个样式的修改合并为一个操作，或者使用 CSS 类来进行样式的切换。这样可以减少样式的计算和应用过程，提高效率。
    4. 使用事件委托：对于大量的子元素需要绑定相同的事件，可以将事件绑定到父元素上，利用事件冒泡机制捕获子元素的事件。这样可以减少事件绑定的数量，提高性能。
    5. 使用文档碎片（DocumentFragment）：如果需要在 DOM 中插入大量的元素，可以先将这些元素添加到文档碎片中，然后再一次性将文档碎片插入到 DOM 中。这样可以减少多次的渲染和回流操作，提高效率。
    6. 避免强制同步布局（reflow）：当读取某些属性（如 offsetWidth、offsetHeight、scrollTop 等）时，会触发强制同步布局操作，影响性能。尽量避免在循环中频繁使用这些属性，或者在使用之前先进行缓存。
    7. 使用 CSS3 动画和过渡：利用 CSS3 的动画和过渡效果可以在 GPU 加速下进行，比 JavaScript 操作 DOM 具有更好的性能。尽量使用 CSS3 动画和过渡来实现页面效果，减少对 JavaScript 操作 DOM 的需求。
20. 事件代理
    1. 事件代理（Event delegation）是一种常见的事件处理模式，它通过将事件处理程序绑定到**父元素或祖先元素**上，来管理子元素上的事件。当子元素触发事件时，事件会冒泡到父元素或祖先元素，然后在那里被捕获并处理。
    2. 通过事件代理，可以避免给每个子元素都绑定事件处理程序，从而提高性能和代码的简洁性。相反，只需在父元素上绑定一个事件处理程序，然后利用事件冒泡机制来监听子元素的事件。
    3. 实际上，事件代理的原理是利用事件冒泡和事件捕获的机制。当子元素触发事件时，事件会从子元素一直冒泡到父元素，期间会经过每一级祖先元素。因此，通过在父元素上绑定事件处理程序，可以捕获并处理子元素触发的事件。
    4. 事件代理的优点包括：
       1. 动态添加的子元素也可以自动绑定事件，无需额外处理。
       2. 减少了事件处理程序的数量，提高了性能和内存利用率。
       3. 简化了代码结构，减少了重复的事件绑定操作。
    5. 常见的使用场景包括：
       1. 列表或表格中的多个项或行需要相同的事件处理。
       2. 动态添加的元素需要绑定事件。
       3. 大量相似元素需要监听相同的事件。
    6. 总之，事件代理是一种利用事件冒泡机制来管理子元素事件的技术，通过将事件处理程序绑定到父元素或祖先元素上，可以简化代码、提高性能，并解决动态添加元素的事件绑定问题。
21. DocumentFragment
    1. **DocumentFragment** 是一个轻量级的文档片段，可以用来临时存储一系列的 DOM 元素，然后一次性添加到文档中，以提高性能。使用 **DocumentFragment** 替代多次调用 **appendChild** 可以减少对 DOM 的多次操作，从而提高效率。
    2. 通过使用 **DocumentFragment**，我们将一系列 DOM 操作合并为一次操作，减少了对 DOM 树的多次修改，提高了性能。
22. input标签的range类别是滑块
    1. 在HTML5中，**input** 标签可以使用 **type** 属性来定义不同的输入类型
23. HTML标签是否受跨域影响
    1. HTML标签本身并不受跨域限制。跨域限制主要适用于浏览器中的 JavaScript 执行环境，涉及到跨域请求数据、修改跨域页面内容等操作。
    2. 当浏览器加载包含跨域资源（例如**图片、脚本、样式表**等）的 HTML 页面时，浏览器会根据资源的跨域策略来判断是否允许加载和显示这些资源。如果资源的服务器设置了正确的跨域响应头（例如 Access-Control-Allow-Origin），则浏览器会允许加载和显示这些跨域资源。
    3. 但是，一些 HTML 标签（例如 **<img>、<script>、<link>**）加载的跨域资源可能会受到跨域限制的影响。例如，如果 <img> 标签的 src 属性指向一个跨域的图片资源，浏览器可能会拒绝加载该图片资源，除非服务器设置了适当的跨域响应头。
    4. 总结起来，HTML 标签本身不会受到跨域限制，但是其中加载的跨域资源可能会受到限制。跨域问题主要涉及到浏览器的安全策略和跨域资源共享（CORS）规范。
    5. CSS文件的加载不受跨域限制
    6. window.onerror方法默认情况下无法获取跨域脚本的报错请求
    7. canvas中使用drawImage贴图回收跨域限制
24. window.onerror方法默认情况下无法获取跨域脚本的报错请求
    1. 在浏览器中，默认情况下，通过 <script> 标签加载的跨域脚本的报错信息是不可访问的，这是由于浏览器的安全策略所限制的。
    2. 跨域脚本加载时，浏览器为了保护用户的安全，会阻止脚本中的错误信息暴露给跨域的页面。这意味着，无法通过常规的方式捕获跨域脚本的报错信息，例如无法通过 window.onerror 或者 try-catch 来获取。
    3. 这种限制是由浏览器的同源策略引起的。同源策略要求脚本只能访问同一域名下的资源，不能直接访问跨域的资源。浏览器会将跨域脚本的错误信息视为来自不同域的资源，因此无法访问报错信息。
    4. 如果你需要获取跨域脚本的报错信息，可以考虑以下方案：
       1. 在跨域脚本中，使用 window.postMessage() 方法向主页面发送报错信息，然后在主页面中接收并处理报错信息。
       2. 在服务器端配置合适的 CORS 头部，允许其他域名的页面访问报错信息。
       3. 通过使用错误监控工具或服务，例如 Sentry、Bugsnag 等，这些工具专门用于收集和监控 JavaScript 错误，可以跨域获取报错信息。
    5. 需要注意的是，以上方案仅在你有控制权并能够修改跨域脚本的情况下有效。如果无法修改跨域脚本或无法配置服务器响应头，那么无法直接获取跨域脚本的报错信息。
25. HTML文档开始部分没有使用doctype是什么意思
    1. 在HTML文件的第一行，类似<!DOCTYPE html>，这代表我们用的HTML的文档生命
    2. 如果没有使用，则怪异模式，浏览器使用自己的怪异模式解析渲染页面
26. 真正区分link标签和@import
    1. <link>标签是HTML标签，@import是CSS规则。
    2. <link>标签可以在HTML文档的任何位置使用，@import语句必须在CSS文件开头使用。
    3. <link>标签的加载是并行的，@import语句的加载是串行的。
    4. 实际开发中，通常推荐使用**<link>**标签来引入样式表，因为它具有更大的灵活性和性能优势。**@import**语句在一些特定的场景下可能会有一些限制，并且对于页面加载性能也有一定的影响。
27. 什么叫做反语义化
    1. 首先理解语义化，是指合理利用HTML标记yiji 其特有的属性去格式化文档内容
    2. font标签已经不支持了
28. area标签、map标签和track标签
    1. <area> 标签用于定义 <map> 元素中的可点击区域，通常用于图像映射（image maps）。
    2. area标签的coords属性
       1. 对于矩形区域recy，使用四个数值表示左上角和右下角的坐标，如 coords="x1,y1,x2,y2"。
       2. 对于圆形区域circle，使用三个数值表示圆心坐标和半径，如 coords="x,y,r"。
       3. 对于多边形区域poly，使用一系列的坐标点，以逗号分隔，如 coords="x1,y1,x2,y2,x3,y3..."。
       4. 对于椭圆形区域ellipse，使用四个数值表示外接矩形的左上角和右下角的坐标，如 coords="x,y,rx,ry"。
    3. <map> 标签用于定义图像映射，即将图像分割为多个可点击区域，并与各个区域关联特定的链接或操作。
    4. <track> 标签用于为 <audio> 或 <video> 元素定义外部的文本轨道，用于提供音频或视频的字幕、描述或翻译。
29. 标签选择器看清楚
    1. 类名是否选对
    2. 是否是直接子元素
    3. 标签种类是否选对
    4. 是否有\*对其进行设置
30. 常见元素属于块级元素还是行内元素
    1. img是行内元素，a是行内元素
    2. blockquote是块级元素，pre是块级元素，fildset是块级元素，canvas是块级元素
31. H5可以向后兼容旧浏览器
    1. 向后表示过去，向前表示未来
32. 一些宽高的计算公式
    1. clientWidth = width + padding\*2
    2. offsetWidth = width + padding\*2 + border\*2
    3. scrollTop = 被卷上去的上侧距离
    4. scrollHeight = height

# HTML20230615

第二次50题，正确率68%

1. HTML中的属性名对大小写不敏感：无论大小写，为了规范，推荐小写
2. 行内元素如何将宽度等分成三部分，-webkit-box-flex和-webkit-flex属性不了解
   1. 目前没有浏览器支持box-flex属性，safari、opera和Chrome支持-webkit-box-flex，firefox支持-moz-box-flex
   2. box-flex属性规定子元素是否可伸缩其尺寸，flex都是相对的
   3. -webkit-flex：webkit内核的浏览器，bi需加上-webkit前缀
   4. 如果元素不是弹性盒模型兑现个资源是，则flex属性不起作用
3. height:100%并不会占满整个页面
   1. 默认情况下，div 标签的高度是由其内容的高度来撑开的。如果 div 标签没有设置固定的高度或通过 CSS 指定了其他高度属性（如 height），它会根据其内部的文本、元素和盒模型属性来自动调整高度。
4. 关于XML
   1. 每个合格的XML都有唯一的根元素
   2. XML的格式上是要求严格的，每个元素的开闭必须完整，不允许交叉开闭
   3. XML常用于WebService中用来做数据交换的标准
      1. XML（可扩展标记语言）在 Web 服务中常用作数据交换的标准。Web 服务是一种通过网络进行通信和交互的软件系统，XML提供了一种通用的结构化数据格式，使不同平台和编程语言之间的系统能够互相交换数据。
   4. XML中元素是大小写敏感的
   5. XML中数据可以通过XPATH检索查询
      1. 在XML中，可以使用XPath（XML路径语言）来检索和定位数据。XPath是一种基于XML文档结构的查询语言，通过使用路径表达式来选择XML文档中的节点。
      2. **/root/child[@id='123'][@type='abc']**
5. XML语言详解
   1. XML（可扩展标记语言）在 Web 服务中常用作数据交换的标准。Web 服务是一种通过**网络**进行通信和交互的软件系统，XML提供了一种通用的**结构化数据格式**，使不同平台和编程语言之间的系统能够互相交换数据。
   2. 在 Web 服务中，XML通常用作数据的载体，通过将数据以标记的形式进行描述和封装，使数据能够在不同的系统之间进行传输和解析。XML具有以下特点使其成为数据交换的理想标准：
      1. 可扩展性：XML使用标签来描述数据结构，可以自定义标签名称和属性，从而满足不同应用的需求。
      2. 结构化：XML数据具有明确定义的结构，可以按照层次关系和父子节点进行组织，使数据具有良好的可读性和可维护性。
      3. 平台无关性：XML是一种与平台和编程语言无关的标准，可以在不同的系统和环境中使用，实现跨平台的数据交换。
      4. 可解析性：XML数据可以使用解析器进行解析和处理，解析器能够识别标签和属性，并将其转换为程序可操作的数据结构，方便数据处理和操作。
   3. 在 Web 服务中，常用的技术如SOAP（简单对象访问协议）和REST（表述性状态转移）使用XML作为数据传输的格式。SOAP使用XML来封装和传递消息，而REST通过HTTP协议直接传输XML数据或其他格式的数据。
   4. 总之，XML作为一种通用的结构化数据格式，在Web服务中广泛应用于数据交换的标准，通过定义明确的数据结构和格式，实现不同系统之间的数据交流和互操作性。
6. date、week、year
   1. 日月年：date，周年week，月年：month，年：year
7. body标签有一个属性是bgcolor，用于设置背景颜色，但是H5已弃用
   1. 不存在background标签
8. H5新增标签
   1. em不是H5新特性，head不是，header是，embed是，datalist是，audio、video是
   2. sessionStorage、localStorage和WebSocket是新增的，cookie、websql不是新增的
9. meter标签，progress标签
   1. **<meter>**元素用于表示一种度量或比例值，并提供了一个可视化的方式来展示该值在指定范围内的进度。它可以用于显示**任务的完成度**、文件**上传**的进度等。
      1. value：表示当前值的属性。
      2. min：表示最小值的属性。
      3. max：表示最大值的属性。
      4. low：表示低阈值的属性。
      5. high：表示高阈值的属性。
      6. optimum：表示最佳值的属性。
   2. **<progress>**元素用于表示一个**任务的完成度或进度**，并提供了一个可视化的进度条。它常用于显示表单提交、文件**下载**等操作的进度。
      1. value：表示当前值的属性。
      2. max：表示最大值的属性。
   3. meter不能作为进度条来使用，progress显示任务的完成进度
10. a标签的target属性值
    1. \_blank：在新的浏览器窗口或选项卡中打开链接。
    2. \_self：在当前窗口或选项卡中打开链接（默认行为）。
    3. \_parent：在父级窗口或框架中打开链接。
    4. \_top：在最顶层的窗口中打开链接，忽略任何嵌套的框架。
    5. framename：在指定名称的窗口或框架中打开链接。如果不存在具有该名称的窗口或框架，则会创建一个新的窗口或框架。
11. tabindex属性
    1. 某个元素使用tabindex属性后，该元素可以通过tab键来获取焦点
12. stong和b标签对字体加粗，而不是bold
13. mark标签突出显示元素指定文本
14. H5废除的元素
    1. font，basefont，big，center，s，strike，tt，u；能用css代替
    2. frameset，frame，noframes；不支持frame框架，只支持iframe框架；或者用服务器放拆功能键的由多个页面组成的符合页面的形式
    3. 部分浏览器支持applet，bgsound，blink，marquee
15. 空元素
    1. 空元素可能是HTML，SVG或者MathML里的一个不可能存在子节点的element
    2. HTML中的空元素：也就是不需要闭合的标签
    3. area\base\br\col\colgroup\command\embed\hr\img\input\keygen\link\meta\param\source\track\wbr
16. 不熟的东西汇总
    1. websocket：WebSocket是一种在Web浏览器和服务器之间进行**双向**通信的协议。它提供了一种**实时**的、**持久**的连接，允许服务器主动向客户端推送数据，同时也允许客户端向服务器发送数据。
    2. WebSQL: Web浏览器提供的一种**客户端数据库**解决方案，允许在**浏览器中**使用SQL语句操作数据库，但**已经被废弃**，推荐使用**IndexedDB**作为替代方案。
    3. sessionStorage: Web浏览器提供的一种**客户端存储**方案，可以在浏览器**会话期间**存储数据，数据保存在用户关闭标签页或浏览器后被清除。
    4. embed: HTML标签，用于**嵌入外部内容**，如嵌入音频、视频、Flash等多媒体内容。
    5. datalist: HTML标签，用于在**输入框**中提供**预定义的选项列表**，用户可以从列表中选择或输入。
    6. iframe: HTML标签，用于在当前网页中**嵌入另一个网页**，实现页面内嵌套和加载外部内容。
    7. bgsound: HTML标签，用于在网页中嵌入**背景音乐**，但已经被废弃，不再推荐使用。
    8. marquee: HTML标签，用于创建**滚动文本**效果，但已经被废弃，不再推荐使用。
    9. blink: HTML标签属性，用于让**文字闪烁**，但已经被废弃，不再推荐使用。
    10. applet: HTML标签，用于**嵌入Java小程序**，但由于安全和性能等问题，已经不再推荐使用。
    11. indexedDB：IndexedDB是一种在Web浏览器中提供的**客户端数据库**解决方案，它允许在浏览器中存储和检索大量结构化数据。与传统的Cookie和LocalStorage相比，IndexedDB具有更强大的功能和性能。
17. position定位
    1. 元素position属性设置为relative时，偏移值是相对于自己
    2. 元素position属性设置为absolute时，偏移值是相对于上一级有定位的祖先元素
    3. 网页中rem作为元素尺寸单位时，是相对文档根节点的font-size进行计算的
    4. 浏览器默认盒模型时content-box
18. 子元素都为浮动，如何解决父元素高度塌陷的问题？举例
    1. 给父元素末尾添加空的块级元素对其使用clearfix类清除浮动
    2. 对父元素使用overflow属性清除浮动，hidden属性设置为auto护着hidden
    3. 对父元素使用clearfix类清除浮动
    4. ab两种方法都是添加伪元素，设置content为空、display为table、clear为both
19. BFC块级格式化上下文
    1. 给父元素添加**overflow: hidden**样式，它会触发 BFC（块级格式化上下文），从而使父元素能够正确计算高度并包含浮动子元素
    2. BFC（Block Formatting Context）是 CSS 中的一种布局机制，用于控制块级元素的渲染规则和布局行为。BFC 在页面上创建了一个独立的渲染环境，其中的元素按照一定的规则进行布局，并且不会影响到其他的元素。
    3. 作用
       1. 清除浮动：当一个元素的子元素都浮动时，父元素会发生高度塌陷，设置父元素的overflow: hidden或display: flow-root会创建一个 BFC，使父元素能够正确计算高度并包含浮动元素。
       2. 阻止外边距重叠：同属于一个 BFC 的相邻块级元素的外边距会发生折叠，而不同的 BFC 之间的外边距不会折叠。
       3. 自适应两栏布局：可以使用 BFC 来实现常见的两栏布局，其中一栏浮动，另一栏形成 BFC，防止内容重叠。
    4. 形成条件
       1. 根元素（<html>）
       2. 浮动元素（float不为none）
       3. 绝对定位元素（position为absolute或fixed）
       4. 行内块元素（display为inline-block）
       5. 表格单元格（display为table-cell）
       6. overflow值不为visible的块级元素
    5. 创建方式
       1. 设置元素的overflow属性值为hidden、auto、scroll等。
       2. 使用float属性进行浮动，不为none。
       3. 设置元素的position属性值为absolute或fixed。
       4. 使用display属性值为inline-block、table-cell等。
20. caption标签不会默认加粗文本，h标签、th标签都会，em标签是倾斜
21. Canvas通过JS的API进行绘图，SVG通过XML进行绘图，W3C参考
22. HTML嵌入PHP代码
    1. <?php 和 ?>
    2. <? 和 ?> 有些服务器环境中可能会将 short\_open\_tag 设置为关闭状态，这意味着使用 <? ?> 的形式可能无法正常解析 PHP 代码
    3. <script language = “php”></script>
23. input元素tel类型定义输入电话号码的控件
24. 绝对URL、相对URL和锚URL
    1. 锚URL通常用于在长页面中直接跳转到页面的特定部分，例如标题、章节或其他标记位置。它以 **#** 符号后面跟随一个标识符来表示
    2. 相对URL（Relative URL）是相对于当前文档所在位置的URL，它不包含完整的协议和域名信息，而是基于当前文档的相对路径进行解析
    3. 绝对URL（Absolute URL）是指包含了完整路径和协议的URL，可以独立地定位资源的位置。它包括协议、域名、端口（可选）、路径、查询参数和片段标识符（可选）
25. 地理定位是H5新引入的特性
26. H5的跨文档消息传输
27. canvas是不是行内元素
    1. canvas不是行内元素而是块级元素，canvas是一个画布
28. 媒体元素audio和video可以触发哪些事件？
    1. play、seeked、abort、playing、seeking、pause
29. 媒体元素audio和video有哪些方法？
    1. play()、load()、pause()
30. H5跨文档消息传输
    1. 可以实现同域内的web页面之间互相通信
    2. 可以相互传递字符串和对象
    3. 使用这个功能，需要获取到网页所在窗口对象的实例
    4. H5（HTML5）中提供了跨文档消息传输的功能，允许不同窗口、标签页或框架之间进行安全的通信。这种通信方式可以用于同源或不同源的文档之间进行消息传递，无论这些文档是来自同一域名下的不同页面，还是来自不同域名的页面。
    5. 可以实现跨域通信
31. meta标签的一些属性理解
32. css样式的等级（从高到低
    1. 内联样式
    2. ID选择器
    3. class|伪类|属性选择器
    4. 标签|伪元素选择器
    5. 通用选择器，子选择器，相邻同胞选择器
33. onsearch事件，用户向搜索域或文本搜索时触发，只有input才有
34. 屏幕的尺寸标准
    1. <768，>=768，>=992，>=1200
    2. 超小， 小 ， 中等， 大屏
35. filter:alpha(opacity=80)
    1. 将对象的背景色透明度改为80%
    2. ie兼容性写法：filter:alpha(opacity:80)
36. 区分fadeIn、fadeOut、fadeToggle和fadeTo
37. p标签虽然是块级元素，但是不能嵌套块级元素，其他块级元素可以
    1. p标签遇到下一个块级元素标签，会自动结束
    2. 所以p标签如果需要换行，可以直接使用p标签，而不需要使用br换行标签
38. table标签中如果要设置表格中文字和边框距离，通过cellpadding属性进行设置
    1. 表格其他常用标签：cellpadding、cellspacing、border、w、h、bgcolor、frame、rules、align
39. 字体颜色是通过color设置，没有text-color属性
40. 各大浏览器的内核
    1. Chrome (Google Chrome, Microsoft Edge Chromium): 使用 **Blink** 内核，它是基于 WebKit 内核的一个分支。
    2. Firefox (Mozilla Firefox): 使用 **Gecko** 内核，它是由 Mozilla 开发的。
    3. Safari (Apple Safari): 使用 **WebKit** 内核，它是苹果自家开发的。
    4. Microsoft Edge (旧版): 使用 **Trident** 内核，这是旧版 Edge 使用的内核，类似于 Internet Explorer。
    5. Microsoft Edge (新版, Chromium 内核): 使用 **Chromium** 内核，与 Google Chrome 共享相同的渲染引擎。
    6. Internet Explorer: 使用 **Trident** 内核，这是旧版的 Microsoft Edge 和 Internet Explorer 使用的内核。

# 网络基础

1. ABC类地址
   1. A、B、C 类地址分别用于大型网络、中型网络和小型网络
   2. A类地址1.0.0.1-126.255.255.254
   3. B类地址128.1.0.1-191.255.255.254
   4. C类地址192.0.1.1-223.255.255.254
2. 在传统的 IP 地址分类中，只有 A、B、C、D 和 E 五个类别
   1. D 类地址范围：224.0.0.0 到 239.255.255.255（用于多播）D 类地址用于多播，多播地址是一种用于将数据报文同时传输给多个主机的通信方式。
   2. E 类地址范围：240.0.0.0 到 255.255.255.255（保留，未分配给特定用途）E 类地址是保留地址，未分配给特定用途。在实际应用中，E 类地址并不常见，一般不用于一般的网络通信。
   3. IP 地址的分类在现代网络中已经不再严格遵循，而是采用了更为灵活的子网划分和无类别域间路由（CIDR）的方式。因此，在实际应用中，我们更多地使用 CIDR 表示法来描述 IP 地址的范围和子网划分。
3. 其他地址
   1. 保留地址
      1. 这些地址是为了特定目的保留的，并未分配给实际的网络通信使用。保留地址的范围是 0.0.0.0 到 0.255.255.255，以及 240.0.0.0 到 255.255.255.254。
   2. 私有地址，私有地址用于内部网络之间的通信，**不被路由器转发**到公共互联网
      1. A 类私有地址范围：10.0.0.0 到 10.255.255.255
      2. B 类私有地址范围：172.16.0.0 到 172.31.255.255
      3. C 类私有地址范围：192.168.0.0 到 192.168.255.255
4. 各种层
   1. 物理层
   2. 数据链路层
   3. 网络层
   4. 传输层
5. 网络设备
   1. 网桥
   2. 交换机
   3. 集线器
   4. 网卡
   5. 路由器
6. 网络设备都工作在哪个层
   1. 集线器、网卡工作在物理层
   2. 网桥、交换机工作在数据链路层，只能识别MAC地址
   3. 路由器在网络层，能识别IP地址
7. OSI
   1. OSI（Open Systems Interconnection）是国际标准化组织（ISO）制定的一个用于计算机网络体系结构的参考模型。它定义了在网络通信中不同层次的功能和协议，并将其分为七个不同的层级，每个层级负责处理特定的任务和功能
   2. 从下到上分为：
   3. 物理层（Physical Layer）：负责传输原始的**比特流**，包括**物理介质、电压和电流**等。
   4. 数据链路层（Data Link Layer）：提供对物理层传输的**比特流进行分组和管理**，包括数据帧的传输和错误检测。
   5. 网络层（Network Layer）：处理数据在不同网络之间的**路由和转发**，包括**寻址和路由选择**。
   6. 传输层（Transport Layer）：负责**端到端**的数据传输，提供可靠的**数据传输和错误恢复**机制。
   7. 会话层（Session Layer）：建立、管理和终止**应用程序之间**的会话。
   8. 表示层（Presentation Layer）：负责**数据**的格式化、加密和压缩，以确保**不同系统**之间的数据能够正确解释。
   9. 应用层（Application Layer）：提供应用程序之间的**通信服务和协议**，包括电子邮件、文件传输、远程登录等。
8. 编码技术
9. 下一跳IP地址？路由器？路由表？目的网络？
10. 交换机？目标MAC？MAC地址表？

# HTML20230616

1. css选择器里面的+符号是什么
   1. 在CSS选择器中，加号（+）符号表示**相邻兄弟选择器**（Adjacent Sibling Selector）。它选择紧接在指定元素后面的同级元素
   2. element + targetElement{ } 这个选择器会选中紧接在**element**元素后面的**targetElement**元素，并且它们是同级元素（具有相同的父元素）。只有第一个满足条件的同级元素会被选择。
2. H5有的元素和H5没有的元素，主要看哪些是删除了的
   1. foot不是！是footer
   2. main标签是H5里面的，main标签规定文档的主要内容
   3. main标签不能是article、aside等新增语义标签的子元素
3. 什么叫做锚伪类
   1. 在支持 css 的浏览器中，链接的不同状态都可以不同的方式显示，这些状态包括：**活动状态**，**已被访问状态**，**未被访问状态和鼠标悬停状态**。用来表示链接不同状态的伪类就是锚伪类。
   2. 有哪些：a:link a:visted a:hover a:active
   3. 分别表示未访问的链接、已访问的链接、鼠标悬停的链接、选定的链接
   4. 几个伪类必须按照一定的顺序(l-v-h-a)，a:hover必须被置于a:link和a:visited之后，才是有效的；a:active必须被置于a:hover之后，才是有效的。
   5. 书写顺序很重要：a:link - a:visited - a:hover - a:actived
4. HTML中不同的编码格式
   1. UTF-8（8-bit Unicode Transformation Format）是一种针对Unicode的可变长度字符编码，又称**万国码**。UTF-8用1到6个字节编码UNICODE字符。用在网页上可以同一页面显示中文简体繁体及其它语言（如英文，日文，韩文）。万国码，所以阿拉伯文可以。其他各国语言都可以。
5. 理解>符号和+符号
   1. div > p：选择父元素是 <div> 的所有 <p> 元素
   2. div + p：选择紧跟 <div> 元素的首个 <p> 元素
6. H5中的新标签，记一下
   1. <!--...--> 定义注释。
   2. <!DOCTYPE> 定义文档类型。
   3. <a> 定义超链接。
   4. <abbr> 定义缩写。
   5. <acronym> HTML 5 中不支持。定义首字母缩写。
   6. <address> 定义地址元素。
   7. <applet> HTML 5 中不支持。定义 applet。
   8. <area> 定义图像映射中的区域。
   9. <article> 定义 article。
   10. <aside> 定义页面内容之外的内容。
   11. <audio> 定义声音内容。
   12. <b> 定义粗体文本。
   13. <base> 定义页面中所有链接的基准 URL。
   14. <basefont> HTML 5 中不支持。请使用 CSS 代替。
   15. <bdi> 定义文本的文本方向，使其脱离其周围文本的方向设置。
   16. <bdo> 定义文本显示的方向。
   17. <big> HTML 5 中不支持。定义大号文本。
   18. <blockquote> 定义长的引用。
   19. <body> 定义 body 元素。
   20. <br> 插入换行符。
   21. <button> 定义按钮。
   22. <canvas> 定义图形。
   23. <caption> 定义表格标题。
   24. <center> HTML 5 中不支持。定义居中的文本。
   25. <cite> 定义引用。
   26. <code> 定义计算机代码文本。
   27. <col> 定义表格列的属性。
       1. **<col>**元素不能直接放在**<table>**标签内部。它通常被放置在**<colgroup>**元素中，作为**<table>**的子元素
   28. <colgroup> 定义表格列的分组。
   29. <command> 定义命令按钮。
   30. <datalist> 定义下拉列表。
   31. <dd> 定义定义的描述。
   32. <del> 定义删除文本。
   33. <details> 定义元素的细节。
   34. <dfn> 定义定义项目。
   35. <dir> HTML 5 中不支持。定义目录列表。
   36. <div> 定义文档中的一个部分。
   37. <dl> 定义定义列表。
   38. <dt> 定义定义的项目。
   39. <em> 定义强调文本。
   40. <embed> 定义外部交互内容或插件。
   41. <fieldset> 定义 fieldset。
   42. <figcaption> 定义 figure 元素的标题。
   43. <figure> 定义媒介内容的分组，以及它们的标题。
   44. <font> HTML 5 中不支持。
   45. <footer> 定义 section 或 page 的页脚。
   46. <form> 定义表单。
   47. <frame> HTML 5 中不支持。定义子窗口（框架）。
   48. <frameset> HTML 5 中不支持。定义框架的集。
   49. <h1> to <h6> 定义标题 1 到标题 6。
   50. <head> 定义关于文档的信息。
   51. <header> 定义 section 或 page 的页眉。
   52. <hgroup> 定义有关文档中的 section 的信息，用于对网页或区段（section）的标题进行组合
   53. <hr> 定义水平线。
   54. <html> 定义 html 文档。
   55. <i> 定义斜体文本。
   56. <iframe> 定义行内的子窗口（框架）。
   57. <img> 定义图像。
   58. <input> 定义输入域。
   59. <ins> 定义插入文本。
   60. <keygen> 定义生成密钥。
   61. <isindex> HTML 5 中不支持。定义单行的输入域。
   62. <kbd> 定义键盘文本。
   63. <label> 定义表单控件的标注。
   64. <legend> 定义 fieldset 中的标题。
   65. <li> 定义列表的项目。
   66. <link> 定义资源引用。
   67. <map> 定义图像映射。
   68. <mark> 定义有记号的文本。
   69. <menu> 定义菜单列表。
   70. <meta> 定义元信息。
   71. <meter> 定义预定义范围内的度量。
   72. <nav> 定义导航链接。
   73. <noframes> HTML 5 中不支持。定义 noframe 部分。
   74. <noscript> 定义 noscript 部分。
   75. <object> 定义嵌入对象。
   76. <ol> 定义有序列表。
   77. <optgroup> 定义选项组。
   78. <option> 定义下拉列表中的选项。
   79. <output> 定义输出的一些类型。
   80. <p> 定义段落。
   81. <param> 为对象定义参数。
   82. <pre> 定义预格式化文本。
   83. <progress> 定义任何类型的任务的进度。
   84. <q> 定义短的引用。
   85. <rp> 定义若浏览器不支持 ruby 元素显示的内容。
   86. <rt> 定义 ruby 注释的解释。
   87. <ruby> 定义 ruby 注释。
   88. <s> HTML 5 中不支持。定义加删除线的文本。
   89. <samp> 定义样本计算机代码。
   90. <script> 定义脚本。
   91. <section> 定义 section。
   92. <select> 定义可选列表。
   93. <small> 将旁注 (side comments) 呈现为小型文本。
   94. <source> 定义媒介源。
   95. <span> 定义文档中的 section。
   96. <strike> HTML 5 中不支持。定义加删除线的文本。
   97. <strong> 定义强调文本。
   98. <style> 定义样式定义。
   99. <sub> 定义下标文本。
   100. <summary> 定义 details 元素的标题。
   101. <sup> 定义上标文本。
   102. <table> 定义表格。
   103. <tbody> 定义表格的主体。
   104. <td> 定义表格单元。
   105. <textarea> 定义 textarea。
   106. <tfoot> 定义表格的脚注。
   107. <th> 定义表头。
   108. <thead> 定义表头。
   109. <time> 定义日期/时间。
   110. <title> 定义文档的标题。
   111. <tr> 定义表格行。
   112. <track> 定义用在媒体播放器中的文本轨道。
   113. <tt> HTML 5 中不支持。定义打字机文本。
   114. <u> HTML 5 中不支持。定义下划线文本。
   115. <ul> 定义无序列表。
   116. <var> 定义变量。
   117. <video> 定义视频。
   118. <xmp> HTML 5 中不支持。定义预格式文本。
7. 关于HTTP请求
   1. 浏览器在一次 HTTP 请求中，需要传输一个 4097 字节的文本数据给服务端，可以通过使用POST
   2. 把数据存入IndexdDB\COOKIE\URL参数\Session和LocalSotrage都不能实现传输给服务端
   3. IndexdDB 是 HTML5 的**本地**存储，把一些数据存储到**浏览器（客户端）**中，当与**网络断开**时，可以从浏览器中读取数据，用来做一些**离线应用**。 Cookie 通过在**客户端 ( 浏览器 )** 记录信息**确定用户身份**，最大为 4 kb 。 url 参数用的是 **get 方法**，从**服务器上获取数据**，大小不能大于 2 kb 。 Session 是**服务器端**使用的一种**记录客户端状态**的机制 。 post 是向服务器传送数据，**数据量较大**。 local Storage 也是 HTML5 的**本地存储**，将数据保存在客户端中（一般是**永久**的）。
8. 区分客户端（浏览器）和服务端（服务器）
9. 记住一些数据大小的限制，比如Cookie和URL参数
10. 理解：H5支持H4中所有的表单控件
    1. 向后兼容
    2. 向上兼容
11. XML和JSON比较
    1. JSON相对于XML来讲，数据的体积小，传递的速度更快些
    2. JSON与JavaScript的交互更加方便，更容易解析处理，更好的数据交互
    3. JSON的速度一般要快于XML
    4. JSON对数据的描述性比XML较差
    5. 【XML】
       1. 优点： （1）格式统一，符合标准； （2）容易与其他系统进行远程交互，数据传输比较方便。
       2. 缺点： （1）XML文件庞大，文件格式复杂，传输占带宽； （2）服务器端和客户端都需要花费大量代码来解析XML，导致服务器端和客户端代码变得异常复杂且不易维护； （3）客户端不同浏览器之间解析XML的方式不一致，需要重复编写很多代码； （4）服务器端和客户端解析XML花费较多的资源和时间。 那么除了XML格式，还有一种叫JSON
    6. 【JSON】
       1. 优点： （1）数据格式比较简单，易于读写，格式都是压缩的，占用带宽小； （2）易于解析，客户端JavaScript可以简单的通过eval\_r()进行JSON数据的读取； （3）支持多种语言，包括ActionScript, C, C#, ColdFusion, Java, JavaScript, Perl, PHP, Python, Ruby等服务器端语言，便于服务器端的解析； （4）在PHP世界，已经有PHP-JSON和JSON-PHP出现了，偏于PHP序列化后的程序直接调用，PHP服务器端的对象、数组等能直接生成JSON格式，便于客户端的访问提取； （5）因为JSON格式能直接为服务器端代码使用，大大简化了服务器端和客户端的代码开发量，且完成任务不变，并且易于维护。
       2. 缺点： （1） 没有XML格式这么推广的深入人心和喜用广泛，没有XML那么通用性； （2）JSON格式目前在Web Service中推广还属于初级阶段 收起
12. XML类似HTML，语法规则如下
    1. 所有的XML元素必须有一个开标签和一个闭标签，并且标签名称要匹配。
    2. XML元素可以包含文本内容、子元素和属性。
    3. XML标签对大小写敏感。
    4. XML文档必须有一个根元素，其他元素必须嵌套在根元素内部。
    5. XML可以使用实体引用来表示特殊字符（如<、>、&等）。
    6. XML可以使用注释（<!-- ... -->）来添加注释信息。
13. 关于页面加载过程不理解
    1. css加载不会阻塞DOM树的解析
    2. css加载会阻塞DOM树的渲染
    3. css加载会阻塞后面js语句的执行
14. 完整的网页加载流程
    1. (1)解析HTML结构
    2. (2)加载外部脚本script和样式表文件link
    3. //页面中的多个脚本文件采用同步的方式加载 会产生“线程等待”
    4. (3)解析并执行脚本代码
    5. (4)构建HTML DOM模型
    6. (5)加载图片等外部资源文件
    7. //图片等文件以异步线程的方式加载 不会产生等待
    8. (6)页面加载完毕
15. 修改元素的颜色会触发重绘，不会触发重排，不会引起回流
16. 重绘、重排、回流
    1. 重绘、重排和回流是浏览器渲染过程中的重要概念，它们与页面元素的**布局和样式变化**有关。
    2. 重绘（Repaint）：当元素的**样式属性**（例如颜色、背景色、边框等）发生改变，但**不影响其布局**时，浏览器会**重新绘制（重绘）**该元素。重绘不会影响其他元素的位置和大小，只会重新绘制元素的可视外观。
    3. 重排（Reflow）：当元素的**布局属性**（例如宽度、高度、位置等）发生改变，或者**页面的结构**发生变化时，浏览器会重新计算元素的布局信息，然后**重新排列（重排）**所有受影响的元素。重排会引起其他元素的位置和大小发生变化，可能涉及整个页面的布局重新计算。
    4. 回流（Layout）：**回流是重排的一种特殊情况**，指浏览器重新计算并**渲染整个页面**布局的过程。回流发生时，浏览器会**从根节点开始遍历整个DOM树**，计算每个元素的位置和大小，并确定它们在页面中的最终布局。回流的开销较大，因为它会导致页面重新渲染。
17. 为了优化页面性能，可以采取一些措施，减少重排和回流的次数，例如：
    1. 使用 CSS3 的 **transform** 和 **opacity** 属性进行动画效果，因为它们通常**只触发重绘而不引起重排和回流**。
    2. **将需要多次操作的 DOM 元素离线处理（Document Fragment）**，先进行修改后再将其添加回文档流，以减少回流次数。
       1. 使用 **document.createDocumentFragment()** 方法创建文档片段
       2. 将需要多次操作的 DOM 元素在内存中进行处理，而不直接插入到文档中。可以使用 **document.createElement()** 方法创建元素，在内存中对元素进行操作，然后再将最终结果插入到文档中。
    3. 使用文档片段（Document Fragment）进行多个 DOM 操作，然后一次性将其添加到文档中，减少回流次数。
    4. 避免频繁访问布局信息，如 **offsetTop** 和 **offsetWidth**，因为它们会强制浏览器进行回流操作。
18. H5全局属性contenteditable
    1. 规定元素内容是否可编辑
19. 一些其他的属性
    1. accesskey class contenteditable contextmenu dir draggable dropzone hidden id lang spellcheck style tabindex title
    2. 分别是：访问元素的键盘快捷键 元素类名 用户是否可编辑(继承父元素) 元素的上下文菜单(主流浏览器不支持)
    3. 元素内容文本方向(ltr rtl auto) 用户是否可拖动(firefox chrome safari支持) 被拖动元素拖到元素上时是否被复制移动和链接 元素是否无关 唯一id 元素内容的语言代码 是否拼写或语法检查 行内样式 tab键控制次序 元素额外信息
20. img标签的图片链接属性名是src还是href
    1. 是src
21. body标签的属性名
    1. bgcolor设置背景颜色
    2. background设置背景图片地址
22. sessionStorage本地存储会话数据，非持久，sessionStorage.setItem(“名”:”值”)
23. 有onmousemove，没有onMove事件
24. 获取Element对象的方法：
    1. 1. getElementById()：根据id属性值获取元素对象。id属性值一般**唯一**
    2. 2. getElement**s**ByTagName()：根据元素名称获取元素对象们。返回值是一个数组
    3. 3. getElement**s**ByClassName()：根据Class属性值获取元素对象们。返回值是一个数组
    4. 4. getElement**s**ByName(): 根据**name属性**值获取元素对象们。返回值是一个数组
    5. 除了**ID选择器的唯一性和querySelector返回单个节点**，其余返回节点列表
25. jQuery方法
    1. slideUp()和slideDown都是jQuery函数，slideUp()通过使用滑动效果，隐藏被选元素，如果元素已显示出来的话。slideDown() 方法通过使用滑动效果，显示隐藏的被选元素。Up是隐藏，Down是显示
    2. 在 jQuery 中，没有内置的 **mergeUp** 和 **mergeDown** 方法。这两个方法属于自定义的函数，用于特定的表格合并单元格操作。如果你想在 jQuery 中实现类似的功能，可以使用 jQuery 的 DOM 操作方法和选择器来编写自己的逻辑。
26. input标签的mutiple属性
    1. 用于select的多选
    2. 用于input的file上传多个文件
    3. 而不是是他可以输入多行文字
27. <a href="#" onmouseover="this.style.fontSize='30px'">注册</a>
    1. 在标签的属性里面写事件，赋值的右边，要用引号
28. DOCTYPE标签
    1. 严格模式的排版和 JS 运作模式是 以该浏览器支持的最高标准运行。在混杂模式中，页面以宽松的向后兼容的方式显示。模拟老式浏览器的行为以防止站点无法工作。
    2. DOCTYPE不存在或格式不正确会导致文档以混杂模式呈现。
29. dp单位？rem单位？px单位
    1. dp是虚拟像素，在不同的像素密度的设备上会自动适配，比如：
       1. 在320x480分辨率，像素密度为160，1dp=1px
       2. 在480x800分辨率，像素密度为240，1dp=1.5px
       3. 所以不可以使用dp
    2. em 自动适应用户所使用的字体
30. JS中的Window对象
    1. Window 对象表示浏览器中打开的窗口，通过Window对象的属性，比如closed属性检索窗口是否已被关闭。 window对象的属性包含内容。 如果文档包含框架（<frame> 或 <iframe> 标签），浏览器会为 HTML 文档创建一个 window 对象，并为每个框架创建一个额外的 window 对象。
    2. Window对象是浏览器所有内容的主容器
    3. Window对象表示浏览器的窗口，可用于检索有关窗口状态的信息
    4. 浏览器打开HTML文档时，通常会创建一个Window对象
31. 哪些HTML元素可以获得焦点？
    1. a标签没有设置href属性时是不能获取到焦点的
    2. <div contenteditable="true">我可以</div>这个也可以获取到焦点
32. canvas绘制rec?circle?
    1. Ctx.arc(x,y,r,0,Math.PI,true);是画半圆
    2. Ctx.arc(x,y,r,0,2Math.PI,true);画整个圆
33. 预格式文本
    1. <pre> 标签可定义预格式化的文本。被包围在 <pre> 标签 元素中的文本通常**会保留空格和换行符**。而文本也会呈现为**等宽字体**
    2. 可导致段落断开的标签如p和address标签，不能包含在pre所定义的块里面
    3. 常见用于表示计算机的源代码
34. 看样式服从哪个要求，先看样式等级，再看综合样式权重
35. 表格的colspan和rowspan
    1. colspan为几，就表示占几列
    2. rowspan为几，就表示占几行
36. object元素和param参数
    1. param 元素允许您为插入 **XHTML** 文档的对象规定 run-time 设置，也就是说，此标签可为包含它的 <object> 标签提供参数。
37. HTML版本
    1. HTML5 是 HTML 最新的修订版本，由万维网联盟（W3C）于 2014 年 10 月完成标准制定。目标是取代 1999 年所制定的 HTML 4.01 和 XHTML 1.0 标准。
    2. HTML 4.01：HTML 4.01 是 HTML 4 的一个规范版本，于1999年发布。它定义了一系列的标记语言元素和属性，并支持脚本和样式表。
    3. XHTML 1.0：XHTML 1.0 是 HTML 4.01 的一个扩展版本，于2000年发布。它基于 XML 格式，要求标签和属性都必须被正确地嵌套和闭合。
    4. HTML5：HTML5 是 HTML 的最新版本，于2014年发布。它引入了许多新的元素、属性和 API，以支持更丰富的多媒体内容、语义化标记和交互功能。
    5. HTML5.1、HTML5.2：HTML5.1 和 HTML5.2 是 HTML5 的后续版本，它们对 HTML5 进行了一些修订和扩展，以提供更好的功能和兼容性。
38. 正确的引入CSS样式的方式
    1. <link rel="stylesheet" type="text/css" **href**="test.css">get
    2. <style type=”text/css”>body{color:red}</style>
    3. src用于**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**
39. <meta name="viewport" content="width=device-width, initial-scale=1">这段代码的作用
40. 针对滚屏适配
    1. 针对滚动屏幕进行适配是一种响应式设计的方法，以确保在不同设备和屏幕尺寸下提供更好的用户体验。下面是一些常用的适配方法：
    2. 使用媒体查询（Media Queries）：通过在 CSS 中使用媒体查询，可以根据不同的屏幕尺寸应用不同的样式和布局。例如，可以使用 @media 媒体查询来设置不同的宽度、高度、字体大小等样式。
    3. 弹性布局（Flexbox）：Flexbox 是一种弹性盒模型布局，可以方便地实现自适应布局。通过设置父容器为 display: flex，并使用弹性盒子属性来定义子元素的布局方式，可以自动调整元素在不同屏幕尺寸下的位置和大小。
    4. 栅格系统（Grid System）：栅格系统是一种将页面分为等宽的列的布局系统。通过将页面划分为多个列，并使用响应式的 CSS 类来定义每个列在不同屏幕尺寸下的宽度，可以实现灵活的布局适配。
    5. 隐藏或显示特定元素：可以使用 CSS 的 display 属性或 JavaScript 来控制在特定屏幕尺寸下隐藏或显示某些元素。这可以使页面在不同设备上呈现不同的内容或布局。
    6. 使用 CSS3 动画和过渡：可以使用 CSS3 的动画和过渡效果来增强滚动屏幕的交互体验。通过使用 CSS3 的 transition 和 animation 属性，可以实现平滑的过渡和动画效果。

# 数据结构20230617