**DAY4**

**HTML5有哪些更新：**

（1）用于绘画的 canvas 元素

（2）用于媒介回放的 video 和 audio 元素

（3）对本地离线存储的更好的支持

（4）新的特殊内容元素，比如 article、footer、header、nav、section

（5）新的表单控件，比如 calendar、date、time、email、url、search

（6）可拖拽（draggable)

**Web存储：**

cookie存储 ： 大小限制在4kb，通过请求服务器返回一个cookieID存在浏览器缓存中，占用宽带，浏览器通用IE6限制每个与，名20个cookies。

SessionStroage： 会话级别的存储数据存储只会在当前会话中，关闭浏览器数据消失

localStroage ： 本地/跨会话存储 浏览器关闭数据依然存在

**src和href的区别：**

**href：**是指向网络资源所在位置，建立和当前元素（[锚点](https://so.csdn.net/so/search?q=%E9%94%9A%E7%82%B9&spm=1001.2101.3001.7020)）或当前文档（链接）之间的链接，用于超链接。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

src：是指向外部资源的位置，指向的内部会迁入到文档中当前标签所在的位置；在请求src资源时会将其指向的资源下载并应用到当前文档中，例如js脚本，img图片和frame等元素。

表示引用资源，表示替换当前元素，用在img，script，iframe上，src是页面内容不可缺少的一部分。

1. 当浏览器遇到href会并行下载资源并且不会停止对当前文档的处理。(同时也是为什么建议使用 link 方式加载 CSS，而不是使用 @import 方式)

2. 当浏览器解析到src ，会暂停其他资源的下载和处理，直到将该资源加载或执行完毕。(这也是script标签为什么放在底部而不是头部的原因)

**什么是重绘与回流：**

**重绘**简单来说就是重新绘画，当给一个元素更换颜色、更换背景，虽然不会影响[页面布局](https://so.csdn.net/so/search?q=%E9%A1%B5%E9%9D%A2%E5%B8%83%E5%B1%80&spm=1001.2101.3001.7020)，但是颜色或背景变了，就会重新渲染页面，这就是重绘。

**回流：** 当增加或删除dom节点，或者给元素修改宽高时，会改变页面布局，那么就会重新构造dom树然后再次进行渲染，这就是回流。

**如何减少重绘和回流:**

1.多个属性尽量使用简写，例如：boder可以代替boder-width、boder-color、boder-style

2.创建多个dom节点时，使用documentfragment创建

3.避免使用table布局

4.避免设置多层内联样式，避免节点层级过多

5.避免使用css表达式

6.将频繁重绘或回流的节点设置为图层，图层能够阻止该节点的渲染行为影响到别的节点（例：will-change\video\iframe等标签），浏览器会自动将该节点变为图层

**script标签中defer和async的区别：**

有了async属性，表示后续文档的加载和渲染与js脚本的加载和执行是并行进行的，即异步执行；

有了defer属性，加载后续文档的过程和js脚本的加载是并行进行的(异步)，此时的js脚本仅加载不

执行, js脚本的执行需要等到文档所有元素解析完成之后，DOMContentLoaded事件触发执行之

前。

**如何实现图片的懒加载：**

Chrome 和 Firefox 都支持通过**[loading](https://link.zhihu.com/?target=https://developer.mozilla.org/zh-CN/docs/Web/Performance/Lazy_loading) 属性**实现延迟加载。此属性可以添加到img元素中。 lazy 值会告诉浏览器，如果图像位于可视区，则立即加载图像，并在用户滚动到它们附近时获取其图像。如果浏览器不支持该属性，则该属性将被忽略，图像将像往常一样立即加载。

**<**img loading**=**"lazy"**>**

**使用 [Intersection Observer](https://link.zhihu.com/?target=https://developer.mozilla.org/zh-CN/docs/Web/API/IntersectionObserver)**

IntersectionObserver 接口(从属于Intersection Observer API)为开发者提供了一种可以异步监听目标元素与其祖先或视窗(viewport)交叉状态的手段。意思就是浏览器可以观察元素是否在可视区内。

*/\**

*初始化 Intersectionobserver 类*

*`callback`是当元素的可见性变化时候的回调函数，`options`是一些配置项*

*\*/***const** observer **=** **new** IntersectionObserver**(**callback**,** options**);***/\* 监听对应的 DOM 元素 \*/*observer**.**observe**(**DOM**);***/\* 取消对 DOM 元素的监听 \*/*observer**.**unobserve**(**DOM**);***/\**

*root：用于观察的根元素，默认是浏览器的视口*

*rootMargin：用来缩放视窗的的大小，用法和 CSS 类似，*

*`10px 10px 30px 20px`表示top、right、bottom 和 left 扩大的对应值，*

*`10px 20px 5px`表示top，left&right， bottom 扩大的对应值*

*thresholds: 当被监听的元素与根元素的交叉比例达到该值时触发回调函数，*

*默认为`[0]`, `0.1`表示当交叉比例达到`0.1`时触发回调，*

*`[0，0.5]`表示当交叉比例达到`0`和`0.5`时都会触发回调*

*\*/***const** options **=** **{**

root**:** **null,**

rootMargin**:** 0**,**

thresholds**:** 1**,};***/\**

*回调函数在监听的 DOM 元素可见性变化时触发*

*entries 中有多个参数，我们使用到以下两个参数*

*isIntersecting 表示观察的元素是否出现在 root 元素可视区，*

*intersectionRatio 表示观察的元素出现在 root 元素的比例*

*\*/*function **callback(**entries**)** **{**

entries**.**forEach**(**entry **=>** **{**

**if** **(**entry**?.**isIntersecting **||** entry**?.**intersectionRatio **>** 0**)** **{**

*// 在此处进行图片地址替换*

**}**

**});}**

**link和@import的区别：**

1、从属关系区别

@import是 CSS 提供的语法规则，只有导入样式表的作用；link是HTML提供的标签，不仅可以加载 CSS 文件，还可以定义 RSS、rel 连接属性等。

2、加载顺序区别

加载页面时，link标签引入的 CSS 被同时加载；@import引入的 CSS 将在页面加载完毕后被加载。

3、兼容性区别

@import是 CSS2.1 才有的语法，故只可在 IE5+ 才能识别；link标签作为 HTML 元素，不存在兼容性问题。

4、DOM可控性区别

可以通过 JS 操作 DOM ，插入link标签来改变样式；由于 DOM 方法是基于文档的，无法使用@import的方式插入样式。

**对CSSSprites的理解：**

CSS Sprites是一种性能优化技术，是一种将多个图像组合成单个图像文件以在网站上使用的方法，以提高性能，也被称为css精灵图

当需要特定图像(精灵图)时，一般会通过CSS background-images属性引用精灵表，在通过CSS background-position属性对其进行偏移定位得到所需的精灵图，然后以[像素](https://so.csdn.net/so/search?q=%E5%83%8F%E7%B4%A0&spm=1001.2101.3001.7020)为单位定义精灵图的大小。

**两栏布局的实现（左侧宽度固定，右侧宽度自适应）：**

**1.利用浮动：**

利用**浮动**，将左边元素宽度设置为200px，并且设置向左浮动。

将右边元素的margin-left设置为200px，宽度设置为auto（默认为auto，撑满整个父元素）。

2**.浮动+BFC：**

利用**浮动**，左侧元素设置固定大小，并左浮动。

右侧元素设置overflow: hidden; 这样右边就触发了**BFC**，**BFC**的区域不会与浮动元素发生重叠，所以两侧就不会发生重叠

**3.利用flex布局：**

    将左边元素设置为固定宽度200px，将右边的元素设置为flex:1

**4.利用绝对定位：**

利用**绝对定位**，将父级元素设置为**相对定位**。

左边元素设置为absolute定位，并且固定宽度设置为200px，left值设置为0。

将右边元素的left值设置为左边固定宽度200px，right值设置为0。

**5.利用网格布局：**

现在给类名为"container"的盒子添加"display: grid"属性，使该盒子成为容器。

再给该容器添加"grid-template-columns: 100px 1fr"属性，表示第一列宽度始终为100px，第二列的宽度为剩余的所有空间。

此时可以看到整个容器的高度因为首列的内容被撑开了，并且右边内容区实现了自适应。

**响应式设计的概念：**

也叫响应式布局，响应式开发

实现不同屏幕分辨率的终端上浏览网页的不同展示方式。

响应式布局是根据设备屏幕宽度不同适当调整标签显示的布局，从而在每种设备屏幕宽度下呈现的界面是不同的

**响应式设计的基本原理：**

响应式设计的基本原理是通过媒体查询检测不同的设备屏幕尺寸做处理，为了处理移动端，页面头部必须有meta声明viewport

**实现响应式布局的方式：**

媒体查询

百分比

rem

vw/vh

**CSS如何设置一行超出显示省略号：**

white-space:nowrap;

overflow:hidden;

text-overflow:ellipsis;

**CSS如何设置多行超出显示省略号：**

overflow: hidden; //超出的文本隐藏

display: -webkit-box;

-webkit-line-clamp: 2; // 超出多少行

-webkit-box-orient: vertical;

**设置小于12px的字体：**

transform: scale(）

使用zoom的方式如：td{font-size:12px;zoom:0.83}

**浏览器地址栏输入url到展示页面的流程：**

1、用户在浏览器中输入url地址

2、浏览器解析域名得到服务器ip地址

浏览器会首先从缓存中找是否存在域名，如果存在就直接取出对应的ip地址，如果没有就开启一个DNS域名解析器。DNS域名解析器会首先访问顶级域名服务器，将对应的ip发给客户端；然后访问根域名解析器，将对应的ip发给客户端；最后访问本地域名服务器，得到最终的ip地址。

3、TCP三次握手建立客户端和服务器的连接

因为HTTP是基于TCP的可靠传输，所以在发送http数据报之前，需要先进行TCP的三次握手建立连接。三次握手过程如下：

第一次握手：客户端--->服务端 ack=1,seq=x（x随机生成）

第二次握手：服务端--->客户端 ACK=1,ack=x+1,seq=y（y随机生成）

第三次握手：客户端--->服务端 ACK=1,ack=y+1,seq=x+1

完成第三次握手时，实际上客户端已经与服务器建立了连接，所以第三次握手的报文已经可以携带数据了。

4、客户端发送HTTP请求获取服务器端的静态资源

5、服务器发送HTTP响应报文给客户端，客户端获取到页面静态资源

6、TCP四次挥手关闭客户端和服务器的连接

数据传输完毕后，TCP会进行四次挥手断开连接，释放资源。四次挥手过程如下：

第一次挥手：客户端--->服务器 FIN=1,ack=1,seq=u 客户端状态变为FIN\_WAIT\_1

第二次挥手：服务器--->客户端 ACK=1,ack=u+1,seq=v 服务器状态变为CLOSE\_WAIT，TCP进入半关闭状态

第三次挥手：服务器--->客户端 FIN=1,ACK=1,ack=u+1,seq=w 服务器状态变为LAST\_ACK

第四次挥手：客户端--->服务器 ACK=1,ack=w+1,seq=u+1 客户端状态变为TIME\_WAIT，此时TCP未释放，需要等待计时器计时完成后，客户端状态变为CLOSED

7、浏览器解析文档资源并渲染页面

浏览器解析文档资源并渲染页面流程：

（1）解析html资源，构建DOM Tree

（2）解析css资源，构建CSS Rule Tree

（3）JS通过DOM API和CSS OM API来操作DOM Tree和CSS Tree

（4）解析完成后综合DOM Tree和CSS Tree会生成Render Tree，计算每个元素的位置，这个过程就是回流（layout or reflow）

（5）调用操作系统Native GUI的绘制

（6）页面绘制完成

背景图片设置：

1、使用background-image属性，语法“background-image:url(图片url);”；2、使用background属性，语法“background:url(图片url);”。