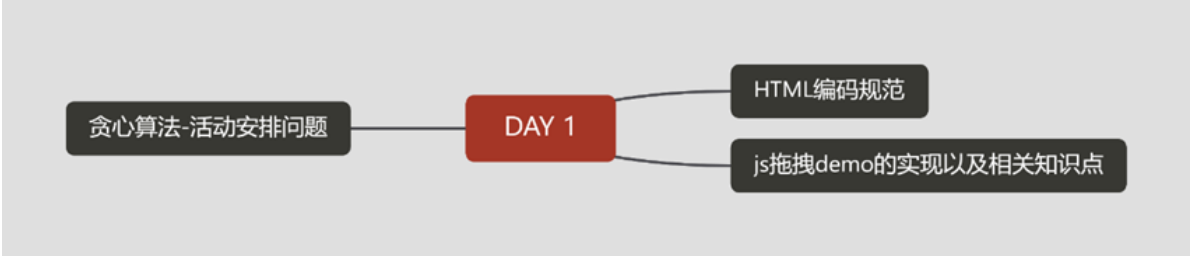
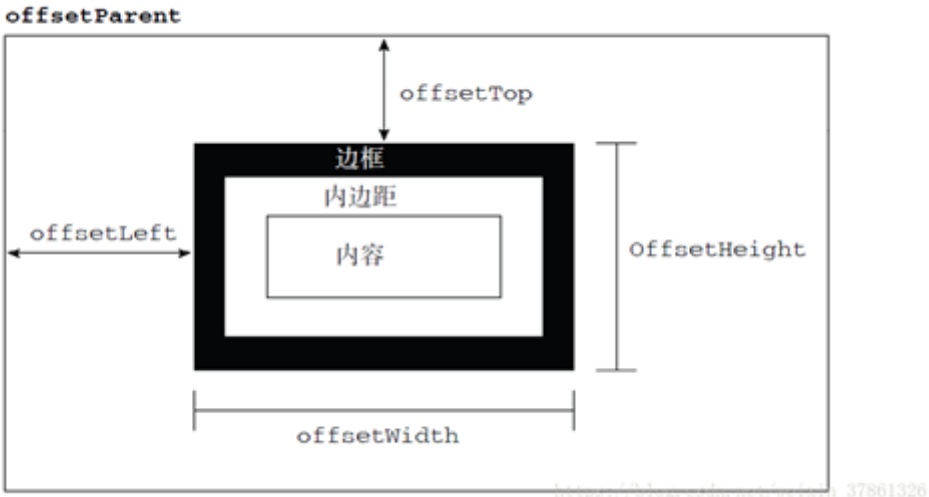
今日大纲：



学习心得

js中的offset、scroll、client的区别

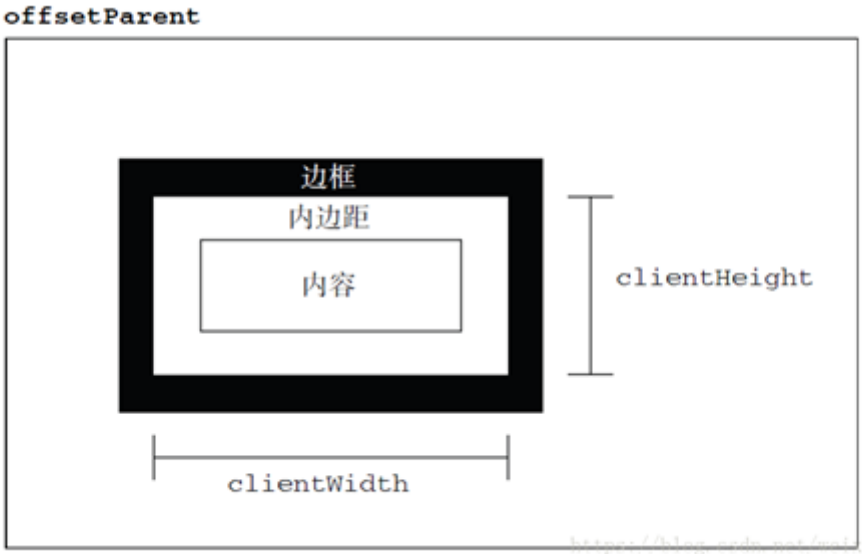


offsetLeft: 是外边框向左到父级元素的距离

offsetTop: 是外边框向上到父级元素的距离

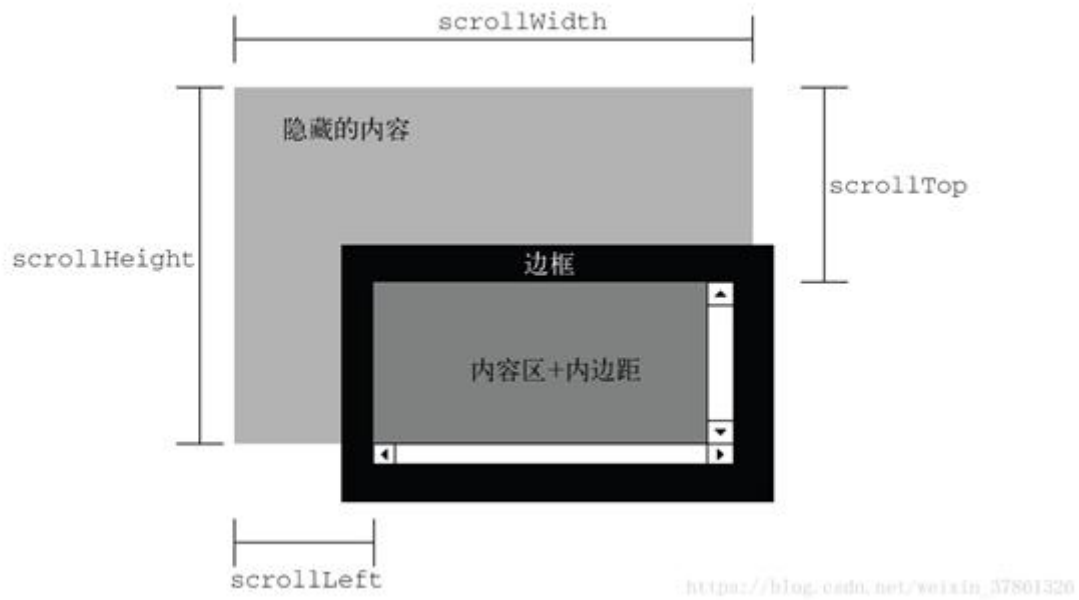
offsetHeight: 元素高

offsetWidth: 元素宽



clientWidth：元素除去外边框之外的宽

clientHeight：元素除去外边框之外的高



scrollHeight：元素隐藏内容的高

scrollWidth：元素隐藏内容的宽

pageX,pageY：鼠标在页面上的位置,从页面左上角开始,以页面为参考点,不随滑动条移动而变化，其实就是clientY+scrollTop，参考点是页面

clientX ,clientY: 鼠标在页面上可视区域的位置,从浏览器可视区域左上角开始,即是以浏览器滑动条此刻的滑动到的位置为参考点,随滑动条移动而变化. 这是一个变化的，是以浏览器左上角为参考点

screenX, screenY: 鼠标相对于屏幕的位置，跟浏览器没有关系，不管浏览器多大，是以计算机屏幕为参考点的

offsetX,offsetY: FF不识别的,鼠标相比较于触发事件的元素的位置,以元素盒子模型的内容区域的左上角为参考点,如果有boder,可能出现负值,是以event对象为参考点

layerX,layerY:如果event元素没有相对定位和绝对定位的话，相当于pageX, pageY,如果有定位属性的话，相当于border+offsetX。

DOM事件中的鼠标事件

属性	描述
onclick	当用户点击某个对象时调用的事件句柄。
oncontextmenu	在用户点击鼠标右键打开上下文菜单时触发
ondblclick	当用户双击某个对象时调用的事件句柄。
onmousedown	鼠标按钮被按下。
onmouseenter	当鼠标指针移动到元素上时触发。不支持事件冒泡
onmouseleave	当鼠标指针移出元素时触发，不支持事件冒泡
onmousemove	鼠标被移动。
onmouseover	鼠标移到某元素之上。
onmouseout	鼠标从某元素移开。
onmouseup	鼠标按键被松开。

拖拽功能主要用到的是onmousedown、onmousemove、onmouseup三大属性，判断不行鼠标事件下，要处理不同的逻辑。

拖拽实现的三个事件

按下时需要监听的是被拖拽的元素，当移动和停止时要监听整个document。

HTML编码规范

1. img要写alt属性，为了防止图片加载不出来的时候，显示该文字
2. 单标签不要写闭合标签
3. 自定义属性要用data开头
4. td要在tr里面，li要在ul/ol里面
5. ul/ol的直接子元素只能是li
6. section里面要有标题标签

   如果用section/aside/article/nav标签，需要在里面写一个h1/h2/h3之类的标题标签，因为这四个标签可以划分章节，它们都是独立的章节，需要有标题。
7. 行内元素里面不可使用块级元素

   可能会导致行内标签无法正常点击,每个页面要写

   ```
   <!DOCType html>
   ```

   写了是标准模式，没写是怪异模式，怪异模式下很多东西渲染会有所不同，怪异模式下input/textarea的默认盒模型会变成border-box，文档高度会变成可视窗口的高度，获取window的高度时就不是期望的文档高度，父容器行高在怪异模式下将不会影响子元素。
8. html要保持简洁，不要套太多层
9. 特殊情况下才在html里面写script和style

但是有时候为了避免闪屏的问题，可能会直接在相应的html后面跟上调整的script

10. 样式要写在head标签里
11. html要加上lang的属性

    有利于SEO和屏幕阅读器使用者，他可以快速地知道这个网页是什么语言的
12. 要在head标签靠前位置写上charset的meta标签

    避免网页显示unicode符号时乱码
13. 特殊符号使用html实体

    | 符号 | 实体编码 |
14. img空src的问题

    会导致浏览器认为src就是当前页面链接，然后会再一次请求当前页面。第一种是把src写成about:blank，第二种办法是写一个1px的透明像素的base64
15. 关于行内元素空格和换行的影响

    换行可能会引入空格
16. 类的命名使用小写字母加中划线连接
17. 不推荐使用自定义标签
18. 重复id和重复属性

    第二个class将会被忽略
19. 不推荐使用属性设置样式
20. 不要在https的链接里写http的图片。

    会导致浏览器地址栏左边的小锁没有了，一般不要写死。
21. 使用适合的标签

1. 如果内容是表格就使用table，table有自适应的优点；如果是一个列表就使用ol/ul标签，扩展性比较好
2. 如果是输入框就使用input，而不是写一个p标签，然后设置contenteditable=true，因为这个在IOS Safari上光标定位容易出现问题。如果需要做特殊效果除外。
3. 如果是粗体就使用b/strong，而不是自己设置font-weight
4. 如果是表单就使用form标签，注意form里面不能套form
5. 如果是跳链就使用a标签，而不是自己写onclick跳转。a标签里面不能套a标签
6. 使用html5语义化标签，如导航使用nav，侧边栏aside，顶部和尾部使用header/footer，页面比较独立的部分可以使用article，如用户的评论。
7. 如果是按钮就应该写一个button或者  ，而不是写一个a标签设置样式，因为使用button可以设置disabled，然后使用CSS的:disabled，还有:active等伪类使用，例如在:active的时候设置按钮被按下去的感觉
8. 如果是标题就应该使用标题标签h1/h2/h3，而不是自己写一个

   ，相反如果内容不是标题就不要使用标题标签了
9. 在手机上使用select标签，会有原生的下拉控件，手机上原生select的下拉效果体验往往比较好，不管是IOS还是android，而使用  在手机上会弹一个电话号码的键盘，   都会弹相应的键盘
10. 如果是分隔线就使用hr标签，而不是自己写一个border-bottom的样式，使用hr容易进行检查
11. 如果是换行文本就应该使用p标签，而不是写br，因为p标签可以用margin设置行间距，但是如果是长文本的话使用div，因为p标签里面不能有p标签，特别是当数据是后端给的，可能会带有p标签，所以这时容器不能使用p标签。

贪心算法

1. 主要思想：

   总是考虑局部最优解。

2. 分配问题 (leetcode 455)

   1. 题目描述

      有一群孩子和一堆饼干，每个孩子有一个饥饿度，每个饼干都有一个大小。每个孩子只能吃一个饼干，且只有饼干的大小不小于孩子的饥饿度时，这个孩子才能吃饱。求解最多有多少孩子可以吃饱。

   2. 输入输出样例

      Input: [1,2], [1,2,3]

      Output: 2

   3. 思路：通过优先考虑饥饿度最小的孩子。给剩余孩子里最小饥饿度的孩子分配最小的能饱腹的饼干。

3. candy (leetcode 135)

   1. 题目描述：一群孩子站成一排，每一个孩子有自己的评分。现在需要给这些孩子发糖果，规则是如果一 个孩子的评分比自己身旁的一个孩子要高，那么这个孩子就必须得到比身旁孩子更多的糖果；所 有孩子至少要有一个糖果。求解最少需要多少个糖果。

   2. input: [1,0,2] Output: 5

   3. 思路：

      两次遍历：把所有孩子的糖果数初始化为 1；先从左往右遍历一遍，如果右边孩子的评分比左边的高，则右边孩子的糖果数更新为左边孩子的 糖果数加1；再从右往左遍历一遍，如果左边孩子的评分比右边的高，且左边孩子当前的糖果数 不大于右边孩子的糖果数，则左边孩子的糖果数更新为右边孩子的糖果数加1。

4. 区间问题

   1. 题目描述：给定多个区间，计算让这些区间互不重叠所需要移除区间的最少个数。起止相连不算重叠

   2. 输入输出样例

      Input: [[1,2], [2,4], [1,3]]

      Output: 1

   3. 思路：

      选择的区间结尾越小，余留给其它区间的空间 就越大，就越能保留更多的区间。因此，采取的贪心策略为，优先保留结尾小且不相交的区间。

5. 活动安排问题

   1. 题目描述：设有n个活动的集合E={1,2,...,n}，其中每个活动都要求使用同一资源，如演讲会场等，而在同一时间内只有一个活动能使用这一资源。每个活动i都有一个要求使用该资源的起始时间si和一个结束时间fi,且si <fi 。如果选择了活动i，则它在半开时间区间[si, fi)内占用资源。若区间[si, fi)与区间[sj, fj)不相交，则称活动i与活动j是相容的。也就是说，当si≥fj或sj≥fi时，活动i与活动j相容。活动安排问题就是要在所给的活动集合中选出最大的相容活动子集合。

i	1	2	3	4	5	6	7	8	9	10	11
S[i]	1	3	0	5	3	5	6	8	8	2	12
f[i]	4	5	6	7	8	9	10	11	12	13	14

2. 思路：

   根据最晚结束时间，选择最晚结束事件晓得集合归并到结果集A中，尽可能加入多的结果。将剩余可安排的时间最大化。

明天计划

1. 写完拖拽demo
2. 实现贪心算法
3. 学习CSS的代码规范
4. 学习浏览器运行原理整理
5. 看js高级编程第九章——代理