

直播笔记

【布局】常见布局（PC）有固定宽度布局、弹性布局、响应式布局——多终端。

【文档流】浮动和绝对定位可以脱离文档流，相对定位不脱离文档流。static默认值。没有定位，元素出现在正常的流中（忽略 top, bottom, left, right 或者 z-index 声明）。

【css技巧】

```
{
position:absolute;
top:0;
bottom:0;
right:0;
left:0;
background: rgba(0,0,0,0.5);
}
```

相当于撑开全屏。

【布局】90%pc网站都是固定布局。好处就是简单，宽度可以写死，内容不会乱，缺点是小屏幕会出现滚动条。弹性布局：按百分比，缺点：复杂，因为会因为屏幕大小而变形；响应式布局。

【布局】设最大宽度/长度可避免滚动条。善用layout。

【css技巧】当你想到inline-block

你要想到

1.verdicle-align

2.间隙。

font-size 去缝隙

【兼容】*zoom 星号ie67才识别

【bug】body默认宽度等于浏览器比如说500，这时候你设置layout为600，而header和footer不在其中，这时候就有可能bug，即布局混乱。所以可以给body加最小宽度

【文档流】width:100%的原始意义是将其撑满其父元素的宽度，一般的块级元素就不用了，除非你是定位的，或者浮动的，脱离了文档流，就需要啦。

【布局】圣杯布局缺点：main宽度不能小于aside（因为aside是-100%上去的，当main宽度小于100%时，aside就不是-90%上去。它必须比自己大才上得去，不然都不够自己大，容纳下自己的坑都不够，也就是说上面的空间都不够自己上去）

【布局】双飞翼之所以不下去，因为content不会因为其内部设置了padding而变小。

【文档流】绝对定位父容器是看不到的，浮动还看得见。

【CSS技巧】

1、不要写height

2、div高度是有什么决定的，是文档流内容的高度，让内容撑外面的高度。

【字体】字体对齐不是中心对齐

【css属性】inline inline-block 水太深

CSS 2.0 inline / block

inline-block

CSS flex

【布局】响应式开发的入门 —— 不要写 width / height ,
但可以写最大宽度；在手机上不准用fixed，它造成的问题是无解的。

【字体】缩小的问题没法解决，因为半个像素无法解决，总有除不尽的情况。

【字体】老师，设计稿给的中文字体不是系统自带的。把字体和代码一起放在服务器上，用font-face引入，但是加载页面后总是先显示浏览器默认字体，然后才能变成引入的字体。有什么办法？是不是字体太大加载比较慢。要么要体验，要么要设计。前端别选，让设计师选。要么就图片。

【css技巧】display:block/inline，可以改变一个元素的展现形式；行内元素你给他设置宽高是无效的，垂直方向的padding和margin亦无效；此时可用line-height起到前两者想要达到的效果。(display:inline-block)

【兼容】ie6/7/8 不支持a以外的标签使用:hover

【布局】display:table-cell是可以居中，但是就不是块级元素了

【布局】元素之间没有依赖关系的就可以用绝对居中呀，像海报图的那种

【布局】为什么table布局被淘汰，因为加载慢，因为它不是一个一个渲染的，而是一整个表格，因为它要去计算宽度，而div是有什么渲染什么；table盖起来麻烦

【JS格式】

```
sum(1)(2)=  
(sum(1))(2);
```

为什么，因为是嵌套函数，前面的执行后就又是一个函数，所以可以这样写。

【时间函数】

```
new Date().getTime() = Date.now() = +new Date()  
getFullYear()  
getMonth()  
getDate()  
getDay()  
getHours()  
getMinutes()  
getSeconds()  
getMilliseconds()  
toString()  
  
var d2=new Date(Date.now()-365*24*60*60*1000)  
d2  
  
getUTCHours()  
  
Date.parse("2011-10-10T14:48:00")  
new Date(1318258080000)
```

【算法】

冒泡：教官双手算法

选择：教官指人算法

插入：插入一个比自己小的旁边

merge归并排序：领导算法，用到了递归

快速：自私算法，我前面的比我矮，我后面的比我高，那我的位置肯定就对了

随机化快速排序：就不会出现最1倒霉的情况（每次只降一两次，跟冒泡比优势很小）

选择排序

```
function sort(array){
  var i
  var j
  var indexOfMin
  for(i=0; i<array.length; i++){
    indexOfMin = i
    for(j=i+1; j<array.length; j++){
      if(array[j] < array[indexOfMin]){
        indexOfMin = j
      }
    }
    if(indexOfMin !== i){
      swap(array, i, indexOfMin)
    }
  }
  return array;
}
```

【命名】

程序员3大难题：

1.变量命名

2.循环边界

3.缓存失效

【JS抓取img地址】

```
var imgSrcs="";
var imgs = document.getElementsByTagName("img");
for(var i=0;i<imgs.length;i++){
  imgSrcs+=imgs[i].src;
  if(i+1<imgs.length)imgSrcs+="|";
}
console.log(imgSrcs);
```