day10

五.框模型（盒子模型）

|  |
| --- |
| 所有元素皆为框------所有元素，默认情况下，在页面上都占据空间  默认情况，这个空间不能被其它元素占用  盒子模型：元素在页面上占据空间的计算方式（公式）    margin:外边距  padding：内边距 |

1.外边距

|  |
| --- |
| 元素边框以外的距离，外边距在页面上是透明的，f12中为橘黄色  改变外边，元素有位移效果  margin:v; 同时设置上右下左4个方向外边距  margin-top:  margin-right:  margin-bottom:  margin-left:  左右冲突以左为准，上下冲突以上为准  取值：1.以px为单位的数字  2.% 4个方向都是父元素宽度的百分比  3.auto auto对上下外边距无效  让块级元素本身水平居中  外边距的使用场景：元素位置的微调  元素之间的距离 |
| 简写方式  margin:v1; 同时设置4个方向外边距 margin:auto;  margin:v1 v2; v1:上下 v2:左右 margin:0 auto; margin:20px auto;  margin:v1 v2 v3; v1:上 v2:左右 v3：下 margin:10px auto 20px;  margin:v1 v2 v3 v4; 上右下左 |

3.外边距的特殊情况

①外边距的合并

|  |
| --- |
| 当两个垂直外边距相遇时，会合并成一个  最后的取值以大的为准  解决方案：  1.只写一个上外边距或者下外边距  2.在设计页面的时候规避这种设计 |

②块级元素，行内元素，行内块的完整总结

|  |  |  |
| --- | --- | --- |
| 块级元素 | 行内元素（span a） | 行内块（input,button） |
| 单独成行 | 与其他行内元素和行内块共用一行，一行放不下自动换行 | 与其他行内元素和行内块共用一行，一行放不下自动换行 |
| 可以设置宽高  默认宽，父元素100%  默认高，内容撑开 | 设置宽高无效  宽高都靠内容撑开 | 设置宽高有效  浏览器自动分配宽高 |
| 4个方向外边距有效 | 上下外边距无效 | 4个方向外边距都有效  改变行内块的上下外边距，  这个行内块会带着同一行其它行内元素和行内块一起移动 |

③自带外边距的元素

|  |
| --- |
| body 8px  ul 上下外边距16px 左内边距40px  body,button,dd,dl,dt,fieldset,form,h1,h2,h3,h4,h5,h6,hr,input,legend,  li,ol,p,pre,td,textarea,th,ul |

④外边距的溢出

|  |
| --- |
| 在特殊情况下，子元素的上外边距会作用到父元素上  特殊情况：1.父元素没有上边框  2.子元素的内容区域上沿与父元素内容区域上沿重合  解决方案：1.给父元素添加上边框，弊端，增加了父元素实际占地高度  2.给父元素添加上内边距，弊端，增加了父元素实际占地高度  3.overflow:hidden/auto,如果父元素想要溢出显示，就不行了  4.利用table标签的特殊性，把子元素和父元素分开。  给父元素第一个儿子的位置处添加空的table元素 |

4.内边距

|  |
| --- |
| 边框到内容区域的距离  改变内边距，感觉上是元素变大了，其实内容区大小没变，内边距大小变了  内边距在页面就是元素背景色，f12中是绿色  padding:v; 设置4个方向内边距  padding-top:  padding-right:  padding-bottom:  padding-left:  padding:v1; 同时设置4个方向  padding:v1 v2; v1:上下，v2:左右  padding:v1 v2 v3; v1:上 v2:左右 v3:下  padding：v1 v2 v3 v4; 上右下左  取值：1.px为单位的数字  2.% 4个方向都根据父元素宽度的百分比  3.auto无效 |

5.改变盒子模型的计算方式

|  |
| --- |
| box-sizing:content-box; 我们设置的width就是内容区域的宽度  左外+左边框+左内+ width +右内+右边框+右外  border-box;我们设置的width包含border,padding,内容区域宽度  左外+ width +右外  特点，只要不改变margin的值，width的值也不变  border,padding,内容区域宽度的设置值在不超过范围的情况，自动协调  结果：border,padding可以改变，并不会影响元素实际占地大小  border-box的使用场合：  一般width在设置为百分比的时候，都会使用border-box来计算  width:25% |

六.背景

1.背景颜色

|  |
| --- |
| background-color:合法颜色值 transparent |

2.背景图片

|  |
| --- |
| background-image: url(路径); |

3.背景图片的平铺

|  |
| --- |
| background-repeat: repeat; 默认值，平铺  no-repeat 不平铺  repeat-x x轴平铺  repeat-y y轴平铺 |

4.背景图片的定位

|  |
| --- |
| background-position:  取两个值，分别设置x y; background-position: 150px 200px;  取一个值，y轴默认居中，只设置x； background-position: 150px;  取值：1.px为单位的数字  2.%  3.关键字 x:left/center/right y:top/center/bottom |

5.背景图片的大小

|  |
| --- |
| background-size:  取两个值，分别设置宽高  取一个值，同时设置宽高  取值：1.px为单位的数字  2.%  3.cover 图片等比缩放，把容器完全充满。图片可能显示不全！  4. contain 图片等比缩放，在容器显示完整图片。容器可能会出现空白区域！ |

6.背景的简写方式

|  |
| --- |
| background:color img repeat option; 顺序可以改变  没有background-size:  最简方式 background:color;  background:img; |

七.渐变

|  |
| --- |
| 渐变是指多种颜色，平缓变化的一种显示效果  渐变的关键因素---色标  色标：一种颜色，这种颜色出现的位置  一个渐变可以有无数个色标，最少两个色标 |

1.线性渐变

|  |
| --- |
| background-image:linear-gradient(方向,色标1,色标2,.........);  方向：to right/to left/to bottom/to top 以终点为表现形式  45deg 以角度为表现形式  色标：颜色 位置；  1.#ff0 0%; 在方向上，0%这个点显示的颜色  2.#ff0 100px;  3.#ff0 只写颜色，不写位置。在方向，把颜色平均分配  background-image: linear-gradient(45deg,#000,#ff0); |

2.径向渐变

|  |
| --- |
| 以圆的方式填充渐变色  background-image: radial-gradient(半径 at 圆心x 圆心y,色标1,色标2......);  半径：以px为单位的数字  圆心：1.以px为单位的数字  2.%  3.关键字 left/center/right top/center/bottom  色标：1.颜色 %； %是半径的百分比  2.颜色 px; px是绝对位置，此时的色标位置与半径就没有关系了  3.颜色； 不写位置，在半径上平均分配色标  background-image: radial-gradient(100px at 100px 100px, #000, #ff0, #000); |

3.重复渐变

|  |
| --- |
| 把线性渐变和径向渐变重复的完成  background-image: repeating-linear-gradient(90deg,#000 0%,#ff0 1%);  background-image: repeating-radial-gradient(100px at center center, #000 0%,#00f 1%); |

4.渐变在低版本浏览器上的兼容

|  |
| --- |
| 刚才所有的语法，只对高版本浏览器生效  低版本浏览器渐变解决方案，我们只关注线性渐变  线性渐变的低版本浏览器兼容，要做到两件事  1.添加前缀  -webkit- chrome/safari  -o- opera  -moz- firefox  -ms- IE  2.如果使用兼容性的写法，线性渐变的方向必须写起点  to right--->left  to top---->bottom  to bottom--->top  to left--->right  background-image: -webkit-linear-gradient(left,#000,#fff);  background-image: -o-linear-gradient(left,#000,#fff);  background-image: -moz-linear-gradient(left,#000,#fff);  background-image: -ms-linear-gradient(left,#000,#fff); |

作业：

1.单独完成6个div的布局和内容



http://101.96.128.94:9999/mfresh/ 拓展练习的网页

http://101.96.128.94:9999/mfresh /mfresh.zip 拓展练习网页资源

https://www.youzhan.org/ 拓展练习网页