day14

三.CSS hack

|  |
| --- |
| 由于不同浏览器对于css的解释和认知不同，导致同一份css在不同浏览器会生成不一样的效果  我们要做浏览器的兼容，就叫做写css hack  background:-webkit-linear-gradient();  -webkit-  -o-  -moz-  -ms-  tmooc视频第三个或者第四个，有专门讲解css hack。看20分钟 |

四.转换

1.什么是转换

|  |
| --- |
| 改变元素在页面中的大小，位置，角度以及形状  2D转换，只发生在x轴和y轴上的转换  3D转换，添加了z轴上的转换 |

2.转换属性

|  |
| --- |
| transform:转换函数1 转换函数2......; |

3.2D的转换函数

①位移

|  |
| --- |
| translate(x)和translatex(x)  在x轴位移  translatey(y)  在y轴的位移  translate(x,y)同时设置x轴和y轴的位移 |

②缩放

|  |
| --- |
| scale(n)  x轴和y轴同时缩放n倍  scale(x,y)  分别设置x轴和y轴的缩放倍数  scalex(x)  单独设置x轴的缩放  scaley(y)  单独设置y轴的缩放  n>1 放大  0<n<1 缩小  -1<n<0 缩小并反转  n<-1 放大并反转 |

3.旋转

|  |
| --- |
| rotate(ndeg)  n>0 顺时针  n<0 逆时针  注意：  1.旋转原点会影响旋转效果  transform-origin:x,y;  取值：1.px为单位的数字  2.%  3.关键字 x:left/center/right y:top/center/bottom  2.旋转是连同坐标轴一起旋转的，会影响旋转之后的位移效果 |

④倾斜

|  |
| --- |
| 1.skew(ndeg)和skewx(ndeg)    让y轴向着x轴倾斜ndeg，n>0逆时针，n<0顺时针  2.skewy(ndeg)    让x轴向着y轴倾斜ndeg,n>0顺时针，n<0逆时针  3.同时设置x轴y轴的倾斜  skew(xdeg,ydeg) |

练习

|  |
| --- |
| 01\_ex  一个div,200px\*200px; 背景色随意，鼠标悬停时  向右偏移100px，向下偏移100px,旋转135deg,向着x轴倾斜50deg,向着y轴倾斜-30deg，放大1.5 |

4.3D的转换函数

|  |
| --- |
| 3d都是模拟的  1.透视距离  模拟人的眼睛到3d转换物体之间的距离，距离不同，看到的效果不同  perspective:距离 一定要设置在3d转换元素的父元素上  2.3D旋转  1.transform:rotatex(xdeg)  以x轴为中心轴，旋转。老式爆米花机，烤羊腿  2.transform:rotatey(ydeg)  以y轴为中心轴，旋转。旋转门，土耳其烤肉，钢管舞  3.transform:rotatez(zdeg)  以z轴为中心轴，旋转。电风扇，摩天轮。  4.让3条轴同时转  transform:rotate3d(x,y,z,ndeg);  x,y,z取值为0，代表不参与旋转。取值为非0，代表该轴旋转的速率 |

五.过渡

|  |
| --- |
| 让css的值，在一段时间从一个状态平缓的变化为另一个状态 |

1.语法

①设置参与过渡的属性

|  |
| --- |
| transition-property：属性1 属性2.......;  取值 all 所有支持过渡的属性，都参与过渡  支持过渡的属性，有哪些  1.颜色属性  2.大多数取值为数值的属性  3.阴影  4.转换 |

②设置过渡时长

|  |
| --- |
| transition-duration  取值 s ms |

③设置时间曲线函数

|  |
| --- |
| transition-timing-function:  取值，1.ease 默认值，慢速开始，中间提速，慢速结束  2.linear 匀速  3.ease-in 慢速开始，一直加速  4.ease-out 快速开始，一直减速  5.ease-in-out 慢速开始，中间大幅度加速，再大幅度减速 |

④过渡代码的编写位置

|  |
| --- |
| 过渡代码写在原始样式中，过渡效果有去有回  写在激活过渡的伪类中，过渡效果，有去无回。 |

⑤过渡的简写方式

|  |
| --- |
| transition:property duration timing-fucntion delay;  最简方式 transition：duration; 属性默认是all |

练习

|  |
| --- |
| 翻滚吧，亮亮  亮亮的吻 |

六.动画

1.什么动画

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 是过渡高级操作，就是一个或者多个过渡效果拼接在一起  过渡和动画的区别   |  |  | | --- | --- | | 过渡 | 动画 | | 两个css效果之间的变化 | 两个或者多个css效果之间的变化 | | 伪类激活 | 伪类激活或者F5激活 | |

2.控制的动画的关键概念

|  |
| --- |
| 用关键帧来控制每一个时间点执行的css值  关键帧  1.执行动画的时间点  2.在这个时间点上的样式 |

3.动画的使用

①使用关键帧定义动画

|  |
| --- |
| @keyframes 动画名称{  0% | from{样式}  ...  100% | to{样式}  } |

②调用定义好的动画

|  |
| --- |
| /\* 调用动画的名称 \*/  animation-name: change;  /\* 动画持续时长 \*/  animation-duration: 5s;  /\* 时间曲线函数 \*/  animation-timing-function: linear;  /\* 动画的延迟 \*/  animation-delay: 1s; |

项目中的动画，90%都使用animate.css提供的动画css

4.动画的特有属性

①指定动画的播放次数

|  |
| --- |
| animation-iteration-count  取值，具体数值  infinite |

②动画的播放顺序

|  |
| --- |
| animation-direction:  normal 0~100% 默认值  reverse; 100%~0 反向  alternate 奇数次normal 偶数次 reverse |

③动画的简写方式

|  |
| --- |
| animation:name duration timing-function delay count direction;  最简方案 animation：name duration; |

④动画开始之前和结束之后的填充方式

|  |
| --- |
| animation-fill-mode:  backwards; 在delay的时间内，显示动画的第一帧  forwards 动画结束之后，停留在最后一帧  both |

5.动画低版本浏览器兼容性的问题

|  |
| --- |
| 要在定义动画的时候写内核  @keyframes 动画名称{}  @-webkit-keyframes 动画名称{}  @-o-keyframes 动画名称{}  @-moz-keyframes 动画名称{}  @-ms-keyframes 动画名称{} |

七.CSS优化

1.css优化的目的

|  |
| --- |
| 后台减少服务器端压力  提升用户体验 |

2.优化的原则

|  |
| --- |
| 尽量减少http的请求个数（使用精灵图、雪碧图）  页面的顶部，引入css文件  将css和js放到单独的文件中 |

3.css代码的优化

|  |
| --- |
| 合并样式(群组，简写方式，尽量减少代码量)  避免出现空的src和href  代码压缩 |

八.css reset 和 normalize.css

1.什么是样式重置

|  |
| --- |
| 不同浏览器，对于元素的默认样式解释不同  导致同一css代码，在不同浏览器运行会有不同效果  我们需要css reset 把浏览器默认样式进行统一  \*{margin:0;padding:0} |

2.css reset

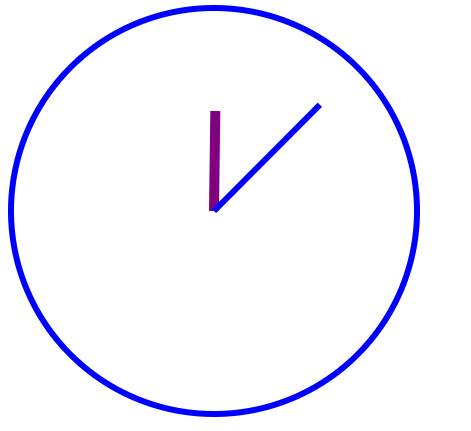
|  |
| --- |
| 没有规则，没有标准  css reset比较霸道，不允许元素拥有自己的特性，统一清空了  很多元素就失去了语义的特性 |

3.normalize.css

|  |
| --- |
| normalize.css是css reset一种替代方案  是一个非常小的css文件，保留很多标签语义特性，又同时排除了很多浏览器的bug  normalize.css有一个标准代码  没有强制的清除某一些样式，只是让大家进行统一 |

作业：

1.使用动画完成



2.使用弹性完成学子底部

