**CSS3**

1. CSS2.1-->CSS3（向后兼容），CSS3兼容性查询网站

http://www.runoob.com/cssref/css3browsersupport.html

1. 前缀 浏览器

-webkit- Chrome、Safari

-moz- firefox

-ms- IE

-o- opera

1. CSS3选择器：eg:<div data=’demo haha’>
2. 可以根据自定义属性data查找元素

div[data=’demo’]{}

1. ^:查找属性中以字母开头的元素

div[data^=’d’]

1. $:以什么做结尾
2. ~：元素中存在单词data
3. 伪类选择器：

:root{}（html根标签）

div:empty{}（选择内容为空的div，有空白本文元素的标签不满足）

div:target{background-color:green}点击目标拥有此样式

div:first-child{}第一个子元素

div:last-child{}最后一个元素

div:nth-child(n+1){} 2n, 2n+1,odd,even

div:nth-last-child(3) 倒数第三个子元素

div:first-of-type{} 找第一个或者被包含中的第一个

div:last-of-type{} 找最后一个或者被包含的最后一个

div:only-child{} 找只有一个子元素的div

div:only-of-type{} 找只包含一个此类型标签的元素

否定选择器：

div:not([data~=demo])选择不满足此条件的元素

1. CSS3边框与背景：
2. border-image:url(xx.png) 100(如何截取边框，距离上下左右各一百像素裁剪) stretch（默认参数，若content宽度超过border宽度会拉伸；repeat：重复平铺；round：border相应宽度/content宽度，四舍五入，填充张数）
3. border-radius：左上 右上 右下 左下（单位px，百分比：圆心contentWidth\*n% contentHeight\*n%，复合属性），可以传八个值，确定四个圆心，两个值，第一个代表左上和右下半径，第二个代表右上和左下半径

百分比为以圆心为坐标绘制椭圆

contentHeight\*n%

contentWidth\*n%

1. box-shadow:x偏移量（px），y偏移量（px），阴影模糊半径（px），阴影扩展半径（px），阴影颜色，投影方式（inset从外层阴影改变内层阴影）

x偏移量

y偏移量

1. CSS3新增背景属性：

background-repeat:repeat/no-repeat;

background-origin:content-box;(背景图片起始位置)

1. background-position：只作用于块级元素和替换元素（img,input,textArea,select,object）,要设置水平方向数值和纵轴方向数值

background-position：15px 20px;（不设置background-origin情况下，以content-box左上为顶点，可以使用百分数）

如果使用top/left/bottom/right单个值，另一个默认为center

1. CSS3：动画
2. transform:实现元素的形状、角度、位置变换
3. perspective：z方向拥有一定像素的深度，配合tansform-style:3d使用；
4. rotateX（正值：朝里），rotateY，rotateZ（正值：顺时针）
5. rotate3d:(1,0,0,45deg),x方向转动45°,（1,1,0,45deg）绕x,y矢量和方向旋转，注意元素坐标轴也会变化
6. scale：以元素几何中心点进行缩放，scale(2)默认x,y都是2，scale（x,y），scaleX，scaleY，scaleZ可以单独设置
7. skew():对元素进行倾斜扭曲，skew(xdeg,ydeg)，如果y值没有，默认为0，理解：只设置X，则X方向像素不变，Y方向倾斜一定度数；同理，只设置Y，Y方向不变，X方向倾斜一定度数，eg：skewX（45deg）

y方向

X方向

1. translateX,Y,Z（Z轴正方向朝外，X正方向朝右，Y正方向朝上）
2. transform-origin：设置旋转，位移中心点可以根据像素设置

中心点设置为top

默认中心点：50%，50%

中心点设置为left

1. tansition:linear,ease,ease-in,ease-out,ease-in-out

eg:

transition:all(所有可以拥有动画的属性) 2s（动画时长） linear （过渡函数，贝瑟尔曲线）cubic-bezier(n,n,n,n) 2s(延时时间)；主要用来设置过渡，display：block，inline-block可以用jQuery中的fadein，fadeout

1. animation 动画关键帧@keyframes，animation会根据关键帧中参数过渡执行

eg：element{animation:move 3s linear 3s reverse infinite}

//描述：利用move动画，过渡时间3s，线性过渡（贝瑟尔曲线），反向运动，无限次执行

@keyframes move{

%0{},100%{}

//from{attributes},to{attributes}

}

1. background:linear-gradient(45deg,#6fc7b5 0%,#13bdce 20%,…,#6fc7b5 100%);
2. CSS3盒模型：

1）w3c:element=width+padding+border

IE6混杂模式:element=width-padding-border

2）over-flow-x：hidden/visible/hidden/scroll,只要设置一个hidden/scroll，超出的话，另一个默认是scroll

3）resize:主要用来改变元素尺寸大小，增强用户体验（IE不兼容）

none/both/horizontal/vertical/inherit

<textarea>可伸缩文本框；resize：none，没法拖动

horizontal：只可x方向拖动。

vertical：只可y方向拖动。

inherit：继承父级属性，父级没有默认both

1. CSS3：弹性盒模型，display：flex
2. 以前方法，左右position：absolute，top/left,中间单独设置。小技巧—>width：calc(100%-20);进行两个单位不同的进行换算
3. .parent{display:flex},.son拥有flex-shrink，flex-grow，flex-basis，order属性
4. 主轴和侧轴，justify-content，默认水平，column表示侧轴自适应
5. 子级共同分担伸缩像素值，flex-shrink：number；最小的不损失，相同的平分，不同则根据和按比例分配压缩像素，默认值是一
6. 子级共同分担多出像素值，flex-grow：number；分配原则与flex-shrink规则相同；
7. flex-basis：200px；与width相同，权重大于width
8. flex：flex-grow flex-shrink flex-basis；复合属性，只写一个默认flex-grow为number值
9. order：number；根据大小决定出现顺序。
10. align-self：auto/flex-start（纵轴顶端）/flex-end（纵轴底端）/center（居中）/baseline（文字第一行的位置对齐）/stretch（默认值：根据父级默认对齐）
11. 设置在容器上的flex属性：flex-flow：flex-direction flex-wrap；
12. align-items设置纵轴方向对齐方式
13. justify-content设置主轴对齐方式，space-between代表每两个项目中间的空白相同，space-around每个项目两边空白相同
14. align-items：center；侧轴方向居中，justify-content:center；主轴居中
15. backface-visibility
16. CSS3性能优化：尽可能多用硬件能力，会开启GPU加速

1）注意：动画化过程有闪烁，考虑浏览器兼容

-webkit-:safari、 chrome

-ms-:IE

-moz-:firefox

-o-:opera

2）translate3d(500px,0,0)比left平移500px动画流畅度好

3）CSS3的3D变形会消耗更多的内存与功耗，确实有性能问题时再去使用它，重在权衡

4）极可能少用box-shadow，gradients，这两个属性最耗性能

5）尽可能的让动画元素不在文档流中，以减少重排

6）优化DOM flow

1. 元素水平垂直居中:（1）left:50%;top:50%;margin-left:-50%;margin-right:-50%;position：absolute；

（2）position：absolute；top：50%；left：50%；transform：translateX（-50%）translate（-50%）；

子元素居中：

parent-element-->display：flex; son-->justify-content:center;align-items:center;

水平居中：

margin:100px auto;

1. animation动画：

1）animation：function-name duration bezziar animation-iteration(执行次数) delay-time animation-direction；

eg：animation：move 5s linear 5(or infinite) 2s forwards;

2）animation-direction：reverse；关键帧反向操作，alternate：动画执行多次，每次执行顺序交替，normal：正常运动方向；forwards：结束位置为关键帧最后一帧的位置；backwards：0% 关键帧是否立即作用；both：backwards和forwards的合并。相当于设置动画初始和结束位置。

3）animation-play-state：通常通过JS添加pause来进行暂停，running：播放

4）animation-iteration：动画执行次数，number/infinite

iteration:反复，重复的意思

1. CSS3：响应式布局（适应各种浏览器，弹性盒子是一种响应式布局解决方案）

响应式布局：随着用户访问web页面终端的多样化，以前的设计目前无法适应这些多样化的终端设备，为满足用户需求，Ethan Marcotte在A List Part发表一篇文章，将弹性网格布局（flex），弹性图片，媒体和媒体查询整合起来，将其命名为响应式布局（Response Web Design）

响应式设计的三个条件：

1. 网站必须建立灵活的网络基础

设置网格宽高不能是固定的

1. 引用到网站的图片必须是可伸缩的

页面中用到图片不能固定宽高

1. 不同的显示风格，需要在MediaQuery上设置不同的样式

在不同终端上正常展示页面，让用户体验不变

1. 流体网格：可伸缩的网络（宽高大小都是可伸缩，利用flex）
2. 弹性图片：图片自适应（把图片放在背景图片位置，让div自适应）
3. 媒体查询（MediaQuery）：让网页在不同终端上面显示效果相同（用户体验相同）
4. 屏幕分辨率：终端屏幕分辨率
5. 主要断点：设备宽度的零界点
6. meta标签：content=”width=device-width’,initial-scale=1.0,minium-scale=1.0,user-scalable=no”,含义：宽度为设备宽度，缩放为比为1(不缩放),maxium-scale最大缩放比，minium-scale最小缩放比，user-scalable是否允许用户缩放

content内的内容就是媒体特性

1. rem:css3新增的一个相对单位，相对的只是HTML根元素，eg：根据HTML的font-size为14px，则子元素3rem为42px。利用媒体查询获得设备类型。

em:相对于父级的大小，父级font-size为14px，则1em为14px

1. 媒体查询MediaQuery方法：

常用all、screen、print

1. 外部引用，link标签：

<link rel=”styleseet”media=”screen and (max-width:500px)” href=”demo.css”’>

1. 内部引用，demo.css下：\*{margin:0;padding:0;}

@media screen and (max-width:700px;) and (min-width:500px){

div{background-color:green;}

}只有宽度满足500-700时内部CSS才能作用上

3）@import url(demo.css) and screen and (max-width:100px) and (min-width:10px)IE6、7不支持

1. 响应式布局技巧
2. 尽量少用无关紧要的div
3. 不要使用内联元素
4. 尽量少用JS和flash
5. 丢弃没用的绝对定位和浮动样式
6. 摒弃任何冗余的结构和不适用100%设置
7. 使用H5的Doctype和相关指南
8. 重置好样式
9. 给重要的网页元素使用简单的技巧，比如导航菜单之类的元素

如何判断结构的好坏：将页面样式全部注释掉，内容排列有序，方便阅读，那么这个结构就不会差。

1. not指定否定条件：

@media not screen and （monochrome）

否定整个声明，所以上面的示例会应用于除单色屏幕外的所有设备（monochrome：单色设备）

1. 物理像素和设备独立像素
2. 分辨率：最大分辨率，由电脑厂商定义（通常为1920\*1080）
3. 实际分辨率：实际自己设置的电脑分辨率（自定义的）
4. CSS像素：感官像素，由浏览器定义的（一个变化的值）
5. 设备独立像素：操作系统定义的
6. 参考像素：根据设备独立像素做了调节，CSS像素根据参考像素做了一层调节
7. 设备像素比：实际分辨率/CSS像素

实际分辨率：物理像素=1:1

CSS像素：设备独立像素=1：1=物理像素：设备独立像素

可以通过devicePixelRatio改变CSS像素与设备独立像素的比值

1. 位图像素：1位图像素
2. 300\*500与600\*1000，一个像素对应四个像素

|  |
| --- |
|  |

1. CSS权重：!important>ID选择器>类选择器，伪类选择器，属性选择器>标签选择器>通配符选择器

**CSS试卷讲解**

1. 利用box-sizing：border-box（IE盒模型），content-box（标准盒模型），可以切换盒模型
2. $(‘element’).on(‘transitionend’,function(){})在元素完成过渡后迅速执行函数
3. 不定宽高元素的水平居中
4. div{position:absolute;top:50%;left:50%;transform:taanslateX(-50%) translateY(-50%);}
5. div{display:flex;justify-content:center;align-items:center;}
6. css3中还可以用hls来设置颜色：色相（H），饱和度（S），明度（L）
7. animation：move 3s ease-in-out 4s infinite;
8. 去掉图片之间的空隙：font-size:0;/float:left;
9. 超出部分用省略号表示：div{width:500px;white-space:nowrap(不换行);overflow:hidden;text-overflow:ellipsis;}
10. 实现文字环绕图片效果：img{float:left;width:150px;}浮动会脱离文档流，但是不脱离文本流。
11. 前端页面由哪三层构成：结构（HTML），样式（CSS），行为（JS）
12. 实现三角形：div{width:0;height:0;border:transparent;border-top:5px solid red;}