复习：

\*{}

div{}

#d1{}

.main{}

div.main{}

div span{}

div > span{}

div + span{}

div ~ span{}

[href]

[type=submit]

[class~=main]

:target

:enabled

:disabled

:checked

:first-child

:last-child

:only-child

:empty

:not()

:first-letter

:first-line

::selection

内容生成

p:before{

content: ‘’ / url() / counter(xx);

}

p:after{

content: ;

}

-----------

body{

counter-reset: myc1;

}

p:before{

counter-increment: myc1;

content: counter(myc1);

}

多列：

div{

column-count: 4;

column-gap: 10px;

column-rule: 1px solid #aaa;

}

1.变形/转换相关的CSS属性

transform: none / 2d-3d-func;

transform: func1() func2() func3();

transform-origin: left/center/right top/center/bottom; 指定变形的原点

注意：(1)所有的变形效果都不会释放或改变原始占用的空间！——与相对定位有点类似。(2)注意Safari的兼容性问题。

1)常见的2D变形函数：

·translate(x)/ translate(x,y) 位移函数

·translateX(x) / translateY(y)

·scale(x) / scale(x, y) 缩放函数，参数是一个表示百分比的小数，如1表100%、0.5表50%、2表200%

·scaleX(x) / scaleY(y)

·rotate(deg) 旋转变形，参数是角度值

·skew(x) / skew(x,y) 倾斜，参数是角度值

·skewX(x) / skewY(y)

2)常见的3D变形函数：

·rotateX(deg) / rotateY(deg) / rotateZ(deg) 沿轴旋转

·translateZ(z) Z轴位移

·scaleZ(z) Z轴上进行缩放，需配合X/Y旋转



2.过渡效果

(1)transition-property: none | all | p1 p2 ; 针对属性

(2)transition-duration：100ms | 3s; 持续时间

(3)transition-timing-function: linear | ease | ease-in | ease-out

(4)transition-delay：100ms | 3s; 延迟时间

(5)transition: property duration timing-function delay ;

变形：transform: 2d-3d-func();

2D变形函数：

translate(x,y)

scale(x,y)

rotate(deg)

skew(deg, deg)

3D变形函数：

rotateX(deg) / rotateY() / rotateZ()

translateZ()

scaleZ()

过渡：由一个状态慢慢的变化到第二个状态

transition: property duration timing-func delay;

3.帧动画

帧动画：在一段比较短的时间内，连续的展示一系列内容变化的图片，就可以实现一个动画的效果；其中的每张图片称为一个“帧(frame)”；电影播放时，1s会播放16帧相关的图片。

补间动画：只需要指定动画变化过程中的“关键帧(keyframe)”，两个关键帧中间的过渡帧由系统来自动填充若干个补间帧。

CSS可使用animation属性实现上述的补间动画效果。

<style>

/\*定义一个关键帧集合——一个动画\*/

@keyframes myAnim1{

0%{ /\*起始关键帧帧\*/

}

20%{ /\*一个关键帧\*/

}

30%{

}

80%{

}

100%{ /\*结束时的关键帧\*/

}  
}

@keysframes myAnim2{

from{/\*起始关键帧\*/

}

40%{ /\*一个关键帧\*/

}

to{ /\*结束关键帧\*/

}

}

div{

animation: myAnim2 3s;

}

</style>

与动画调用相关的CSS属性：

·animation-name：动画的名称，即某个@keyframes后声明的动画名。

·animation-duration：动画的持续时间

·animation-timing-function：动画播放速度函数 linear / ease-in / ease-out / ease-in-out

·animation-delay: 播放的延时时间

·animation-iteration-count: 动画播放次数，如3、10、infinite

·animation-direction: 播放方向 normal(第二次播放时从第一帧重新开始)；alternate(第偶数次播放时从最后一帧倒序播放)

·animation：集合属性，按顺序指定name duration timing-function delay count direction

·animation-fill-mode:

-backwards: 动画尚未开始时即处于第一帧的状态

-forwards: 动画完后后保持最后一帧的状态

-both: 上述二者的效果都要

·animation-play-state: paused(动画处于暂停状态) running(动画处于运行状态)

说明：过渡(transition)可以看做一种特殊的动画(animation)——只有开始和结束两个关键帧。

4.CSS Hack

CSS Hack：由于浏览器对于CSS属性有不同的理解或支持程度不同，为了屏蔽这种浏览器方面的不同实现统一的显示效果，或者专门利用这样的不同的显示效果，可以在编写CSS时，针对特定的浏览器或者特定的版本给予特别的代码。

CSS Hack的实现方式：

(1)IE条件语句：只在IE9-有效

<!--[if IE]> 小于IE10的浏览器会看到此句<![endif]-->

<!--[if IE 6]> IE6看到此句<![endif]-->

<!--[if lt IE 8]> 小于IE8的浏览器会看到此句 <![endif]-->

<!--[if lte IE 8]> 小于等于IE8的浏览器会看到此句 <![endif]-->

上述条件语句中可以放置任何CSS/HTML/JS语句。

(2)选择器前缀

<style>

.content{ } 所有浏览器都能理解的选择器

\*html .content{} 只有IE6能理解的选择器

\*+html .content{} 只有IE7能理解的选择器

</style>

(3)属性前缀

<style>

.content{

-webkit-animation: anim1 3s;

-moz-animation: anim1 3s;

-o-animation: anim1 3s;

background: red; /\*所有浏览器都能识别\*/

\*background:green; /\*IE6/IE7能识别\*/

\_background:blue; /\*IE6/IE7能识别\*/

+background:yellow; /\*IE能识别\*/

background: yellow\9\0; /\*IE9+能识别\*/

background: pink !important; /\*IE6无法识别\*/

}

</style>

5.页面访问速度优化

(0)硬件/网络优化

(1)数据库优化

(2)服务器优化

(3)前端优化: HTML优化、CSS优化、JS优化

CSS优化方案：

优化原则：尽可能减少HTTP请求数量；尽可能减少每次请求的数据大小

优化方法：

(1)CSS Sprites：背景图滑动门、把很多的小背景图拼接为一副大图——百度“CSS Sprites在线”可以找到很多这样的工具

(2)把CSS放到页面顶部，多用<link href=”x.css”/>代替@import url(x.css)

(3)避免使用CSS表达式

(4)避免空的src和href值

(5)巧用浏览器缓存，把CSS放在尽可能少的HTML外部文件

(6)首页中尽量不用外部CSS

(7)不要在HTML中缩放图像

(8)对JavaScript文件和CSS文件进行压缩(剔除空白/换行/注释等)，减小文件大小。可使用类似YUI Compressor等工具 Yahoo UI Libary

<a href=””></a>

<style>

body{...}

@import url(xxx.css);

</style>

面试题：

<link>和@import引入CSS文件的区别？