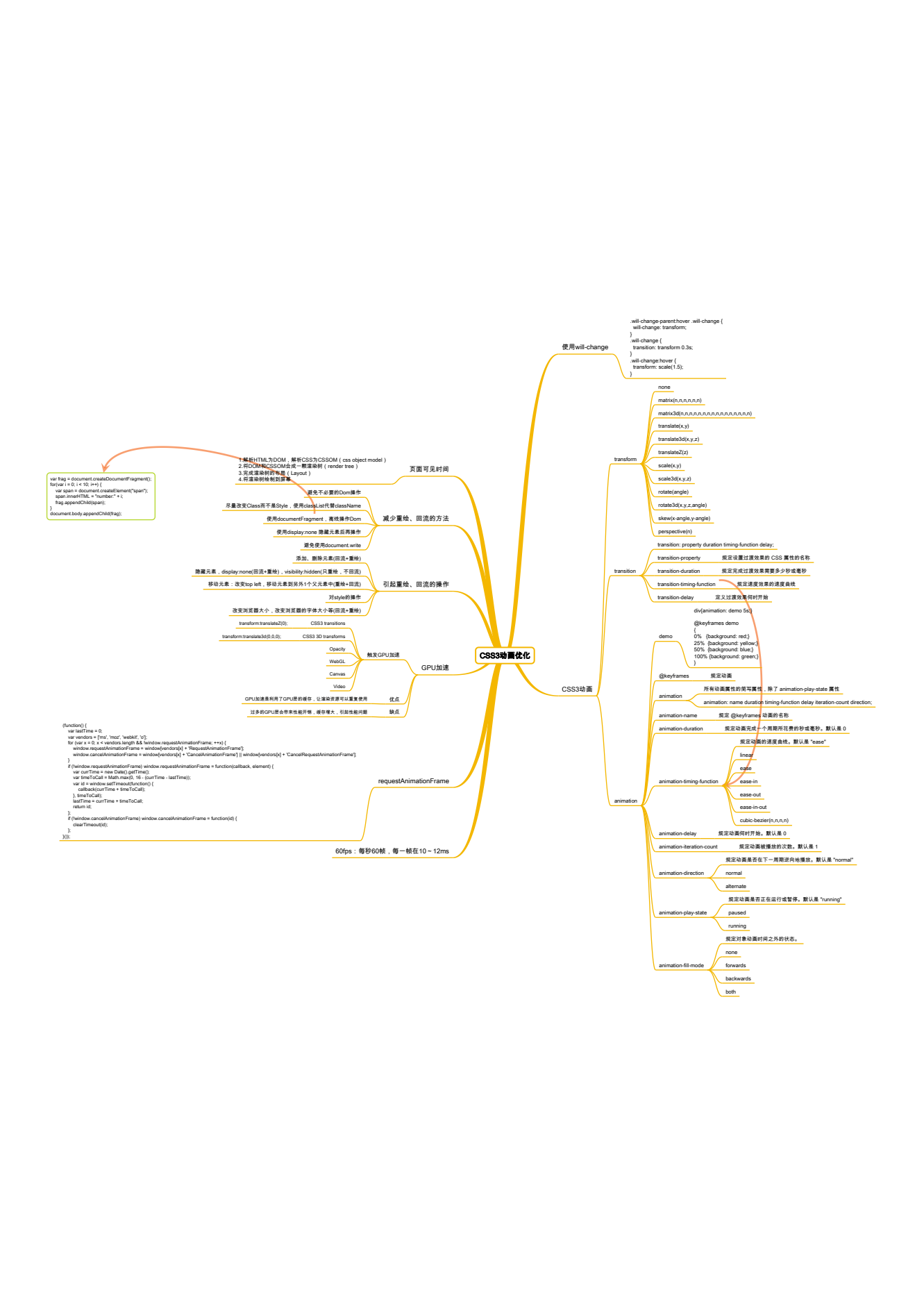
CSS3动画优化
CSS3动画
使用will-change
.will-change-parent:hover .will-change {
will-change: transform;
}
.will-change {
transition: transform 0.3s;
}
.will-change:hover {
transform: scale(1.5);
}
transform
none
matrix(n,n,n,n,n,n)
matrix3d(n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n)
translate(x,y)
translate3d(x,y,z)
translateZ(z)
scale(x,y)
scale3d(x,y,z)
rotate(angle)
rotate3d(x,y,z,angle)
skew(x-angle,y-angle)
perspective(n)
transition
transition: property duration timing-function delay;
transition-property 规定设置过渡效果的 CSS 属性的名称
transition-duration 规定完成过渡效果需要多少秒或毫秒
transition-timing-function 规定速度效果的速度曲线
transition-delay 定义过渡效果何时开始
div{animation: demo 5s;}
@keyframes demo
{
0% {background: red;}
25% {background: yellow;}
50% {background: blue;}
100% {background: green;}
}
animation
demo
@keyframes 规定动画
animation 所有动画属性的简写属性，除了 animation-play-state 属性
animation: name duration timing-function delay iteration-count direction;
animation-name 规定 @keyframes 动画的名称
animation-duration 规定动画完成一个周期所花费的秒或毫秒。默认是 0
animation-timing-function 规定动画的速度曲线。默认是 "ease"
linear
ease
ease-in
ease-out
ease-in-out
cubic-bezier(n,n,n,n)
animation-delay 规定动画何时开始。默认是 0
animation-iteration-count 规定动画被播放的次数。默认是 1
animation-direction 规定动画是否在下一周期逆向地播放。默认是 "normal"
normal
alternate
animation-play-state 规定动画是否正在运行或暂停。默认是 "running"
paused
running
animation-fill-mode 规定对象动画时间之外的状态。
none
forwards
backwards
both
页面可见时间
1.解析HTML为DOM，解析CSS为CSSOM（ css object model ）
2.将DOM和CSSOM合成一颗渲染树（ render tree ）
3.完成渲染树的布局（ Layout ）
4.将渲染树绘制到屏幕
减少重绘、回流的方法
避免不必要的Dom操作
var frag = document.createDocumentFragment();
for(var i = 0; i < 10; i++) {
var span = document.createElement("span");
span.innerHTML = "number:" + i;
frag.appendChild(span);
}
document.body.appendChild(frag);
尽量改变Class而不是Style，使用classList代替className
使用documentFragment，离线操作Dom
使用display:none 隐藏元素后再操作
避免使用document.write
引起重绘、回流的操作
添加、删除元素(回流+重绘)
隐藏元素，display:none(回流+重绘)，visibility:hidden(只重绘，不回流)
移动元素：改变top left，移动元素到另外1个父元素中(重绘+回流)
对style的操作
改变浏览器大小，改变浏览器的字体大小等(回流+重绘)
GPU加速
触发GPU加速
transform:translateZ(0); CSS3 transitions
transform:translate3d(0,0,0); CSS3 3D transforms
Opacity
WebGL
Canvas
Video
优点 GPU加速是利用了GPU层的缓存，让渲染资源可以重复使用
缺点 过多的GPU层会带来性能开销，缓存增大，引起性能问题
requestAnimationFrame
(function() {
var lastTime = 0;
var vendors = ['ms', 'moz', 'webkit', 'o'];
for (var x = 0; x < vendors.length && !window.requestAnimationFrame; ++x) {
window.requestAnimationFrame = window[vendors[x] + 'RequestAnimationFrame'];
window.cancelAnimationFrame = window[vendors[x] + 'CancelAnimationFrame'] || window[vendors[x] + 'CancelRequestAnimationFrame'];
}
if (!window.requestAnimationFrame) window.requestAnimationFrame = function(callback, element) {
var currTime = new Date().getTime();
var timeToCall = Math.max(0, 16 - (currTime - lastTime));
var id = window.setTimeout(function() {
callback(currTime + timeToCall);
}, timeToCall);
lastTime = currTime + timeToCall;
return id;
};
if (!window.cancelAnimationFrame) window.cancelAnimationFrame = function(id) {
clearTimeout(id);
};
}());
60fps：每秒60帧，每一帧在10 ~ 12ms