#第一天

1. 描边的时候会占用原图形的一部分（线宽的一半），所以在开发中，为了符合线宽的要求，应先填充后描边
2. Canvas在绘制线条的时候，会向左右各偏移线宽的一半，然后进行绘制：若线宽会奇数px，那么线条的边缘的颜色值会减掉一半（因为线条偏移后，奇数px会被分割成0.5px，而实际上0.5px又会被当成1px，所以线条的长度被扩充为原来线宽的两倍，但是为了还原成原来的线宽，线宽和颜色又被同时缩减为原来的一半）。---注意，以上变化仅影响线条的边缘，即先被扩充再被缩减的那0.5px
3. 线帽样式：butt(默认)，round(圆角，增长为线宽的一半)，square
4. 交点样式：mitter(默认，尖角)，round(圆角)，bevel（平角）。线帽样式不影响设置的交点样式
5. arr.forEach(function(val,index){ })
6. arr.map(function(val,index,){ return newVal})。其中newArr为经过处理过的数组
7. Math.max.apply(数组),求一个数组中的最大值。利用apply的平铺数组特性
8. Arr.map()会返回结果；arr.forEach()不会返回结果

#第二天

1. 如果网站中有少量的小图片，可以考虑直接使用base64格式的图片
2. 单独设置ctx.font不会生效：ctx.font=”20px”,”ctx.font=”700 20px”，以上方式都不行。这样才行：ctx.font =”20px 微软雅黑”
3. 浏览器对图片的加载是异步的
4. Window.onload 事件要加载完全部页面元素才能执行；img.onload直接加载完对应的这个img元素就可以执行

#第三天

1. 绘制新的图像时，一定要先清除画布
2. Canvas平移或旋转后不会影响已经绘制的图形，并且效果都会累加
3. 默认的，旋转的中心是坐标轴的中心，但是这样会出问题；应该让选中的中心为图形的中心，才能不出问题
4. 动态加载图片：方式一，只写img标签，不写src,然后通过js再动态赋值src; 方式2：通过var img =new Iamge()动态创建元素，然后通过js赋值src。----适用于ctx.drawImage()

#第4天

1. 状态的保存于回滚与路径无关
2. RequestAnimationFrame:请求动画帧函数；类似于setTimeout,区别在于不用指定时间，当要重现新绘制动画时，由浏览器自动触发