# JS高级阶段第一次笔试题

**笔试目的**：锻炼大家的笔试能力，不会让你在笔试中出现提笔忘字的情况出现。同时也是对自己学习成果的一次检验，掌握本阶段最核心最精华的知识点。

1. 问题：什么是面向对象？(10分)

本题目的：检测对面向对象概念，面向对象特性的理解

答案：

# 面向对象是一种编程思想. 在使用对象的过程中, 不需要去管对象内部是怎么实现的, 只需要知道可以实现什么功能就可以了.（3分）

# 面向对象具有封装性，继承性，多态性（3分）。封装隐蔽了对象内部不需要暴露的细节，使得内部细节的变动跟外界脱离,只依靠接口进行通信.封装性降低了编程的复杂性.。通过继承,使得新建一个类变得容易,一个类从派生类那里获得其非私有的方法和公用属性的繁琐工作交给了编译器。而继承和实现接口和运行时的类型绑定机制所产生的多态,使得不同的类所产生的对象能够对相同的消息作出不同的反应,极大地提高了代码的通用性（4分）.

1. 问题：什么是原型链？(10分)

本题目的：检测原型链的理解和通过原型链的结构的分析

答案：

当从一个对象那里调取属性或方法时，如果该对象自身不存在这样的属性或方法，就会去自己关联的prototype对象那里寻找（4分），如果prototype没有，就会去prototype关联的前辈prototype那里寻找（2分），如果再没有则继续查找Prototype.Prototype引用的对象（2分），依次类推，直到Prototype.….Prototype为undefined（2分）（Object的Prototype就是undefined）从而形成了所谓的“原型链”。

1. 问题：什么是Canvas？(10分)·

本题目的：检测Canvas的理解和用途

答案：

# canvas是H5的新标签，称为画布（5分）

# 可以通过js来绘制文字、图像、图形、动画以及图片的特殊效果（5分）。

1. 问题：.代码实现，canvas画一个任意扇形(10分)

本题目的：检测canvas画线和画弧形的结合使用

答案：

# var ctx = canv.getContext("2d");

# ctx.beginPath();（2分）

# ctx.lineWidth = 1;

# ctx.moveTo(300,300);

# ctx.arc(300,300,100,0,90\*Math.PI/180,false);（6分）

# ctx.closePath();

# ctx.stroke();（2分）

1. 问题：canvas实现文字hello canvas，阴影效果：text-shadow：5px 10px 15px ‘#ccc’;放在100px 100px位置。(10分)

本题目的：检测canvas绘制文字和阴影相关API的使用

答案：

# var ctx = canv.getContext(“2d");

# ctx.beginPath();（2分）

# //设置文字阴影

# //阴影颜色

# ctx.shadowColor = "#ccc";（2分）

# //阴影x偏移

# ctx.shadowOffsetX = 5;（2分）

# //阴影y偏移

# ctx.shadowOffsetY = 5;（2分）

# //阴影模糊度

# ctx.shadowBlur = 5;

# ctx.fillText("hello canvas”,100,100);（2分）

1. 问题：代码实现，canvas图片反色处理(10分)

本题目的：检测对于图片像素处理和画布绘制图片的掌握情况

答案：

# var ctx = canvas.getContext("2d");

# ctx.beginPath()；

# var img = new Image();

# img.src = “xxx.png";（2分）

# img.onload = function(){

# var bili = img.width/img.height;

# ctx.drawImage(img,100,100,300,300/bili);（2分）

# var myData = myPixel.data;

# for(var i = 0; i < myData.length; i+=4){

# var r = myData[i];

# var g = myData[i+1];

# var b = myData[i+2];（2分）

# myData[i] = 256 - r;

# myData[i+1] = 256 - g;

# myData[i+2] = 256 - b;（2分）

# ctx.clearRect(0,0,canv.width,canv.height);

# ctx.putImageData(myPixel,100,100);（2分）

# }

# }

1. 问题：代码实现，canvas图片黑白处理(10分)

本题目的：检测对于图片像素处理和画布绘制图片的掌握情况

答案：

# var ctx = canvas.getContext("2d");

# ctx.beginPath()

# var img = new Image();

# img.src = “xxx.png";（2分）

# img.onload = function(){

# var bili = img.width/img.height;

# ctx.drawImage(img,100,100,300,300/bili);（2分）

# var myData = myPixel.data;

# for(var i = 0; i < myData.length; i+=4){

# var r = myData[i];

# var g = myData[i+1];

# var b = myData[i+2];（2分）

# myData[i] = (r + g + b)/3;

# myData[i+1] = (r + g + b)/3;

# myData[i+2] = (r + g + b)/3;（2分）

# ctx.clearRect(0,0,canv.width,canv.height);

# ctx.putImageData(myPixel,100,100);（2分）

# }

# }

1. 问题：canvas图像发生重叠时，设置显示重叠样式的属性是什么？属性的值destination-out、destination-over分别是什么意思？(10分)

本题目的：对于重叠样式的理解

答案：

# 设置显示重叠样式的属性是：globalCompositeOperation（4分）

# （1）destination-out：显示源图像之外的目标图像，源图像透明（2分）

# （2）destination-over：让目标图像显示在源图像之上（2分）

1. 问题：代码实现，canvas拖拽功能。(10分)

本题目的：检测画布与js事件的配合情况

答案：

# var ctx = canv.getContext(“2d");

# var x = 20, y = 20, r = 20;

# ctx.beginPath();

# ctx.fillStyle = "red";

# ctx.arc(x,y,r,0,Math.PI\*2);

# ctx.fill();（2分）

# document.onmousedown = function(ev){（2分）

# var e = ev || window.event;

# var mouseX = e.clientX;

# var mouseY = e.clientY;

# var circleX = x;

# var circleY = y;

# if(ctx.isPointInPath(mouseX,mouseY)){

# document.onmousemove = function(ev){

# var e = ev || window.event;

# var nowMouseX = e.clientX;

# var nowMouseY = e.clientY;

# var pianyiX = nowMouseX - mouseX;

# var pianyiY = nowMouseY - mouseY;

# x = circleX + pianyiX;

# y = circleY + pianyiY;

# ctx.clearRect(0,0,canv.width,canv.height);

# ctx.beginPath();

# ctx.fillStyle = "blue";

# ctx.arc(x,y,r,0,Math.PI\*2);

# ctx.fill();

# }（3分）

# document.onmouseup = function(){

# document.onmousemove = null;

# }（3分）

# }

# }

1. 问题：.canvas.isPointInPath是什么意思？作用是什么？

本题目的：检测对于.isPointInPath的理解

答案：

# isPointInPath: 检查某点坐标,是否在当前canvas路径下。参数为坐标。（5分）

# 返回值为true是在当前canvas路径下，为false是不在当前canvas路径下。（5分）