**1.模板区（template）**

1） canvas的宽高只能通过在行内设置width和height，不能使用css设置，否则会导致图形拉伸

2）当低版本的浏览器不支持canvas标记时，会把canvas展示为一个普通的标记，就会显示在canvas内部写的提示语，而如果浏览器支持canvas,内部的提示语则不会显示

<canvas width="700" height="700" ref="canvas">

您的浏览器版本过低,不支持canvas,请升级浏览器或使用chrome浏览器

</canvas>

<button @click="save">保存Canvas</button>

**2.样式区（style区）**

canvas实际上是一个内联元素，要设置它在页面水平居中，需要把它设置为块级元素，（再次声明，不可以在css里设置canvas的宽高）

canvas {

display: block;

margin: 0 auto;

border: solid 2px #999;

}

**3.脚本区(script区)**

**（1）获得canvas**

<script setup>

import { onMounted, ref } from 'vue'; // 从vue导入钩子函数和ref

// 找到canvas节点，可以使用js的document寻找，也可以使用vue的ref寻找

const canvas = document.querySelector('canvas')

const canvas = ref(null)

</script>

两个的区别就是，用ref找到的canvas使用时需要使用canvas.value,

而document.querySelector找到的canvas使用时不需要加.value

**（2）获取画笔**

const ctx = canvas.getContext('2d') // 使用document找到的

const ctx = canvas.value.getContext('2d') // 使用ref找到的

**（3）画线（两点确定一条直线）**

ctx.beginpath() // 开启一条路径

ctx.moveTo(x1, y1) // 将画笔移动到一个点

ctx.lineTo(x2, y2) // 将画笔移动到第二个点

ctx.srtokeStyle = '#333' // 设置描边的颜色

ctx.width = 3 // 设置线宽，取值为数值

ctx.stroke() // 开始画

ctx.closePath() // 关闭路径

**（4）画矩形**

ctx.beiginPath() // 如果不重新开启路径，会和上一次画的连上

ctx.fillStyle = '#0f0' // 填充样式

ctx.fillRect(x1, y1, x2, y2) // 填充

ctx.closePath()

**（5）擦除**

ctx.clearRect(x1, y1, x2, y2)

**(6）画圆**

ctx.beginPath() // 开启新路径

ctx.arc(x1, y1, 圆形半径, 角度值1, 角度值2,顺时针还是逆时针)

// 例如

ctx.arc(100, 100, 100, 0, Math.PI, true);

ctx.stroke()

**(7）画字（写字）**

ctx.font = '30px 微软雅黑'

ctx.fillText('文字内容', x1, y1)

ctx.fillText('canvas',100, 100) // 实心文字

ctx.strokeText('canvas', 100, 100) // 空心文字

**(8）canvas转换base64地址**

//用ref找到的元素，要加.value

let base64 = canvas.value.toDataURL('image/jpeg', 1);

**(9）下载canvas画的图**

// 模板区绑定的save方法

cosnt save = () => {

canvas.value.toBlob(blob => {

let blobURL = URL.createObjectURL(blob);

let aNode = document.createElement('a')

aNode.href = blobURL

aNode.download = 'Canvas图片'

aNode.click()

})

}

**（10）加载图片并加文字水印**