

```

int row = 20; // 行数
int grid_count_per_row = 20; // 每行格子的数量(其实就是列数)
float grid_width, grid_height;

void setup()
{
    size(500, 500); // 设置程序窗口大小为 500像素 x 500像素

    grid_width = width / grid_count_per_row; // 计算格子的宽度 窗口宽度除以每行格子数
    grid_height = height / row; // 计算格子的高度 窗口高度除以行数
}

void draw()
{
    color blue = color(0, 0, 128); // 蓝色
    color light_blue = color(0, 0, 255); // 浅蓝色
    color grid_color = light_blue;

    for (int i = 0; i < row; i++)
    {
        // 遍历所有行，就是第一层loop是循环处理每一行的逻辑
        // 这里每一行的逻辑是要确定每一行第一个格子的颜色
        if (i % 2 == 0)
        {
            // 如果是偶数行，第一个格子颜色是浅蓝色
            grid_color = light_blue;
        }
        else
        {
            // 否则第一个格子的颜色是蓝色
            grid_color = blue;
        }

        for (int j = 0; j < grid_count_per_row; ++j)
        {
            // 这里的循环是遍历行中的每一个格子，主要的逻辑就是设置格子的颜色，然后把格子画出来

            fill(grid_color); // 设置格子填充颜色
            stroke(grid_color); // 设置格子边框颜色

            // 画格子，rect函数的4个参数分别是 rect(矩形x坐标, 矩形y坐标, 矩形宽度, 矩形高度)
            // 窗口的坐标系如下：原点在左上角，沿右边为x正方向，沿下边为y正方向
            // .----- x
            // |-----
            // |-----
            // |-----
            // |-----

```

```

// |-----
// y

// 所以
// 每个格子的x坐标就是他处在每行中的第几个位置，乘以格子的宽度
// 每个格子的y坐标就是他所处的行数，乘以格子的高度
rect(j * grid_width, i * grid_height, grid_width, grid_height);

// 判断下一个格子的颜色
if (grid_color == blue)
{
    // 如果当前格子颜色是蓝色，那么下一个格子颜色就是浅蓝色
    grid_color = light_blue;
}
else
{
    // 如果当前格子颜色是浅蓝色，那么下一个格子颜色就是蓝色
    grid_color = blue;
}
}
}
}

```