

Assignment 5

选中列后显示对应列

使用两个变量维护按动状态，一个表示当前是否按下，另一个表示按动列的序号。在按动表格列项时，更准确来说是在触发`onmousedown()`动作时，调用`touch`函数。

1. 如果当前按动序号与已按动的不符合，无论当前状态如何，都需要按下对应列。因此需要先恢复原列，再按动；
2. 如果再次按动同一序号，则需要更新状态，检查当前应该按下还是抬起；

female	local
26	69

触发与非触发状态的对比

```
// 维护按动状态
let column_index = 0;
let down = 0;

// 如果按动新的，先取消原本的再按，否则直接按
function touch(table, index) {
  if (index !== column_index) {
    down = 1;
    unchoose(table, column_index);
    choose(table, index);
  } else {
    down = 1 - down;
    if (down == 1) choose(table, index);
    else unchoose(table, index);
  }
}

// 选中列改变背景颜色，并作图
function choose(table, index) {
  column_index = index;
  let unit = table.rows[0].cells[index];
```

```

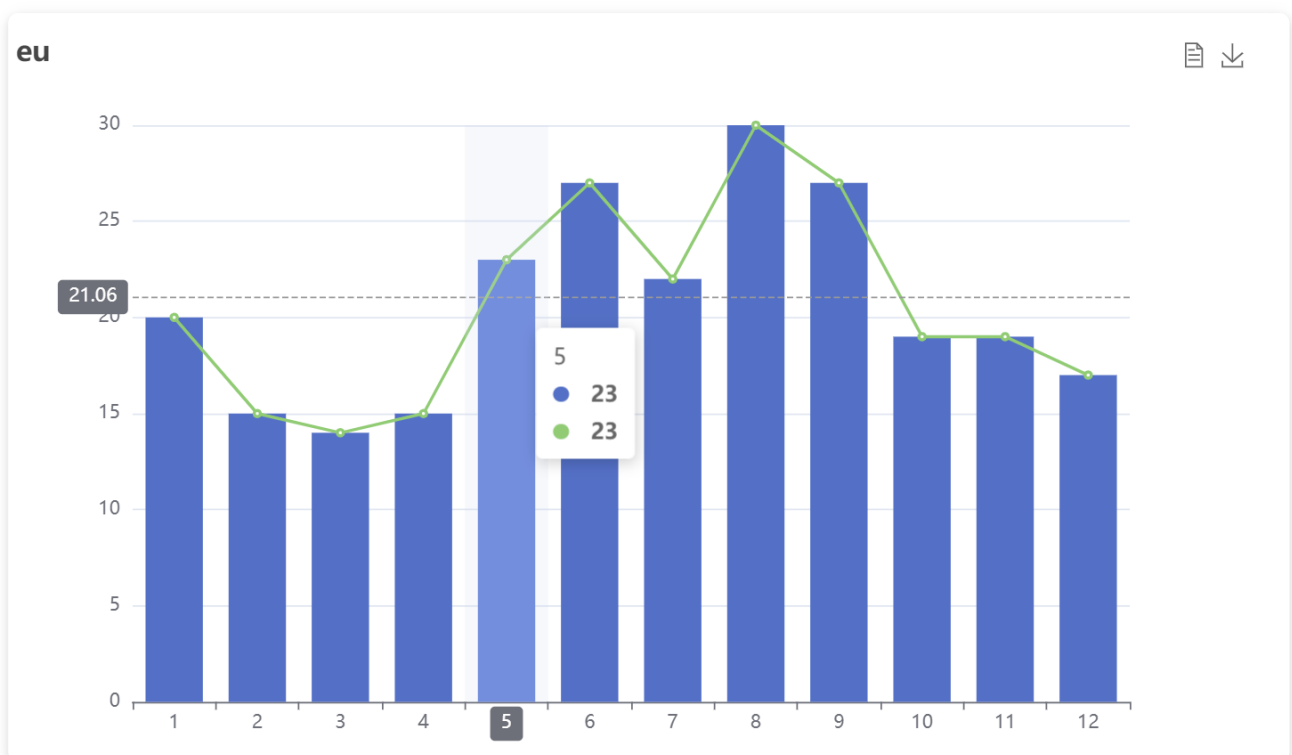
unit.style.backgroundColor = "grey";
charts(table);
}

// 取消选中，回归背景色，并清除原图像
function unchoose(table, index) {
    let unit = table.rows[0].cells[index];
    unit.style.backgroundColor = "whitesmoke";
    var myChart = echarts.init(document.getElementById('main'));
    myChart.clear();
    var option = {};
    myChart.setOption(option);
}

```

绘制柱状图与折线图

在绘图之前，需要获得对应列的数据，可以参考此前保存时使用的获取表格数据方法，挑选对应列的数据即可。在取得数据后，参考echarts中模板表格的写法，只设定一个y轴，添加line与bar两种类型的数据，即可完成。



绘图样例

```

function charts(table) {
    var myChart = echarts.init(document.getElementById('main'));
    let y_data = get_table_data(table)[column_index - 1];
    let x_data = [];
    let Max = 0;

```

```

for (let i = 1; i <= y_data.length; i++) x_data[i-1] = i;
for (let i = 0; i < y_data.length; i++) {
    if (y_data[i] !== "") y_data[i] = parseFloat(y_data[i]);
    else y_data[i] = 0;
    if(y_data[i]>Max) Max = y_data[i];
}
let chart_title = table.rows[0].cells[column_index].innerHTML;
var option = {
    title: {
        text: chart_title
    },
    tooltip: {
        trigger: 'axis',
        axisPointer: {
            type: 'cross',
            crossStyle: {
                color: '#999'
            }
        }
    },
    toolbox: {
        feature: {
            dataView: {show: true, readOnly: true},
            saveAsImage: {show: true}
        }
    },
    xAxis: [
        {
            type: 'category',
            data: x_data,
            axisPointer: {
                type: 'shadow'
            }
        }
    ],
    yAxis: [
        {
            type: 'value',
            min: 0,
            max: Max
        }
    ],
    series: [

```

```
        {
            type: 'bar',
            data: y_data
        },
        {
            type: 'line',
            data: y_data
        }
    ]
};
myChart.setOption(option);
}
```