연습문제

#문제 1 ~ 3

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>
        .subplot {
            float: left;
            width: 33.3%;
            padding: 50px;
            box-sizing: border-box;
        }

        .subplot-item {
            width: auto;
            height: 320px;
        }
    </style>
</head>
<body>
    <!--학과별 학생수 세로 막대 그래프-->
    <div class="subplot">
        <h2>학과별 학생수</h2>
        <div class="subplot-item">
            <canvas id="mychart1"></canvas>
        </div>
    </div>

    <!--학년에 따른 평균 나이 변화 선 그래프-->
    <div class="subplot">
        <h2>학년에 따른 평균 나이 변화</h2>
        <div class="subplot-item">
            <canvas id="mychart2"></canvas>
        </div>
    </div>

    <!--학년별 평균키와 평균 몸무게 다중 막대 그래프-->
    <div class="subplot">
        <h2>학년별 평균키와 평균 몸무게</h2>
        <div class="subplot-item">
            <canvas id="mychart3"></canvas>
        </div>
    </div>
    <script src="dataset.js"></script>
    <script
```

```
src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/3.7.1/chart.min.js"></script>
    <script>
        /** 각 항목을 저장할 배열 선언*/
        const dept = [] // 학과
        const g = [] // 학년
        const birth = [] //생년월일

        /** 각 항목의 배열에 값 입력*/
        // 문제 1번의 값들
        for (let i=0; i < student.length; i++) {
            dept[i] = student[i].deptno;
        }
        const deptResult = {};
        dept.forEach((x) => {
            // deptResult[x]가 '컴퓨터과'라면 처음에는 값이 없으므로 0 + 1 = 1 이 된
다.
            // deptResult[x]가 다시 '컴퓨터과'가 되면 deptResult[x]에 저장되어 있던 1
과 + 1이 되어 2가 됨. 이런식으로 증가한다.
            // deptResult[x]가 '컴퓨터과'가 아니라 '정보통신과'라면 값이 없으므로 0 +
1 이 되어 1이된다
            // 이 과정이 끝날때까지 반복한다.
            deptResult[x] = (deptResult[x] || 0) +1;
        });
        // Object.getOwnPropertyNames() 를 이용하여 deptResult의 키값 가져오기
        const department = Object.getOwnPropertyNames(deptResult);
        // 반복문을 통해 studentCount 배열에 deptResult의 value값 넣기
        const studentCount = []
        for (let i in deptResult) {
            studentCount.push(deptResult[i])
        }

        //문제 2번의 값들
        //학년값 및 태어난 년도 가져오기
        for (let i = 0; i < student.length; i++) {
            g[i] = student[i].grade;
            birth[i] = (student[i].birthdate + "").substring(0,4);
        };

        //학년 중복값 없애기
        const grade = g.filter((v, i) => {
            return g.indexOf(v) === i
        })
        for (let i in grade) {
            grade[i] = grade[i] + "학년"
        }

        //나이 계산
        const now = new Date()
        const age = []
        const ageinfo = {}
        for (let i = 0; i < birth.length; i++) {
            //한국인은 태어날때부터 1살
            age[i] = now.getFullYear() - parseInt(birth[i]) + 1
        }
```

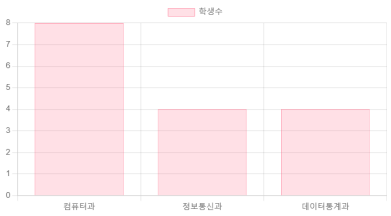
```
        //학년별로 나이값 넣기
        const ageArr1 = []
        const ageArr2 = []
        const ageArr3 = []
        const ageArr4 = []
        for (let i = 0; i < student.length; i++) {
            g[i] = student[i].grade;
            if (g[i] == 4) {
                ageArr4.push(age[i])
                ageinfo[grade[0]] = ageArr4
            } else if (g[i] == 1) {
                ageArr1.push(age[i])
                ageinfo[grade[1]] = ageArr1
            } else if (g[i] == 3) {
                ageArr3.push(age[i])
                ageinfo[grade[2]] = ageArr3
            } else if (g[i] == 2) {
                ageArr2.push(age[i])
                ageinfo[grade[3]] = ageArr2
            }
        }

        //학년별 평균 나이값 계산
        for (let i in ageinfo) {
            let sum = 0;
            for (let j = 0; j < ageinfo[i].length; j++){
                sum += ageinfo[i][j];
            }
            ageinfo[i] = (sum / ageinfo[i].length);
        }
        //학년별 평균 정렬하기
        const orderedAgeInfo = {}
        Object.keys(ageinfo).sort().forEach(function(key) {
            orderedAgeInfo[key] = ageinfo[key];
        })
        //학년별 JSON의 키값과 value값 가져오기
        const ageInfoKeys = Object.getOwnPropertyNames(orderedAgeInfo);
        const ageInfoValue = []
        for (let i in orderedAgeInfo) {
            ageInfoValue.push(orderedAgeInfo[i])
        }

        //문제3 값
        // heigth값과 weight 값 넣기
        const h1 = []
        const h2 = []
        const h3 = []
        const h4 = []
        const w1 = []
        const w2 = []
        const w3 = []
        const w4 = []
        for (let i = 0; i < student.length; i++) {
            if (student[i].grade == 4) {
```

```
            h4.push(student[i].height)
            w4.push(student[i].weight)
        } else if (student[i].grade == 3) {
            h3.push(student[i].height)
            w3.push(student[i].weight)
        } else if (student[i].grade == 2) {
            h2.push(student[i].height)
            w2.push(student[i].weight)
        } else if (student[i].grade == 1) {
            h1.push(student[i].height)
            w1.push(student[i].weight)
        }
    }
    // bodyInfo값 넣기
    const bodyInfo = {}
    bodyInfo[grade[0]] = {"height": h4, "weight": w4}
    bodyInfo[grade[1]] = {"height": h1, "weight": w1}
    bodyInfo[grade[2]] = {"height": h3, "weight": w3}
    bodyInfo[grade[3]] = {"height": h2, "weight": w2}


    const ordereBodyInfo = {}
    Object.keys(bodyInfo).sort().forEach(function(key) {
        ordereBodyInfo[key] = bodyInfo[key];
    })
    const avgHeight = []
    const avgWeight = []
    for (let i in ordereBodyInfo) {
        let sum = 0;
        let avg = 0;
        for (let j = 0; j < ordereBodyInfo[i].height.length; j++) {
            sum += ordereBodyInfo[i].height[j]
        }
        avg = (sum / ordereBodyInfo[i].height.length)
        avgHeight.push(avg)
        sum = 0;
        avg = 0;
        for (let k = 0; k < ordereBodyInfo[i].weight.length; k++) {
            sum += ordereBodyInfo[i].weight[k]
        }
        avg = (sum / ordereBodyInfo[i].weight.length)
        avgWeight.push(avg)
    }
    console.log(avgWeight)

    //그래프를 담을 html코드 불러오기
    const mychart1 = document.querySelector('#mychart1')
    const mychart2 = document.querySelector('#mychart2')
    const mychart3 = document.querySelector('#mychart3')

    //그래프 만들기

    new Chart(mychart1, {
        //그래프 종류
```

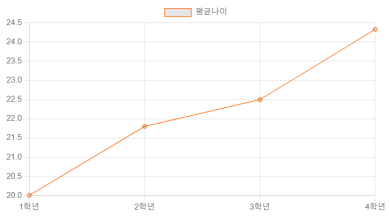
```
            type: 'bar',

            //데이터 종류
            data: {
                labels: department,

                datasets: [
                    {
                        label: "학생수",
                        data: studentCount,
                        borderWidth: 0.5,
                        borderColor: 'rgba(255,99,132,1)',
                        backgroundColor:"rgba(255,99,132,0.2)",
                    },
                ],
            },
            options: {
                maintainAspectRatio: false,
            },
        })

        new Chart(mychart2, {
            type:"line",

            data: {
                labels: ageInfoKeys,

                datasets: [
                    {
                        label: "평균나이",
                        data: ageInfoValue,
                        borderWidth: 1,
                        borderColor: "#ff6600"
                    },
                ],
            },

            options: {
                //그래프의 가로, 세로 비율 유지(기본값=true) -> 부모 div의 넓이에만 맞
추고 높이는 무시함
                //false로 설정할 경우 부모의 widht, height에 크기를 맞춤
                maintainAspectRatio: false,
            },
        })

        new Chart(mychart3, {
            type: "bar",

            data: {
                labels: ageInfoKeys,

                datasets: [
                    {
                        label: "키",
```

```
                        data: avgHeight,
                        borderWidth: 0.5,
                        borderColor: 'rgba(54,162,235,1)',
                        backgroundColor: 'rgba(54,162,235,0.2)'
                    },

                    {
                        label: '몸무게',
                        data: avgWeight,
                        borderColor: 'rgba(255,99,132,1)',
                        backgroundColor:"rgba(255,99,132,0.2)",
                    },

                ],
            },

            options: {
                //그래프의 가로, 세로 비율 유지(기본값=true) -> 부모 div의 넓이에만 맞추고 높이는 무시함
                //false로 설정할 경우 부모의 widht, height에 크기를 맞춤
                maintainAspectRatio: false,
            },

        })
    </script>
</body>
</html>
```

학과별 학생수



학년에 따른 평균 나이 변화



학년별 평균키와 평균 몸무게