Рекурсия

//Примепр по итератору

let students = {

    js: [{

        name: 'John',

        progress: 100

    }, {

        name: 'Ivan',

        progress: 60

    }],

    html: {

        basic: [{

            name: 'Peter',

            progress: 20

        }, {

            name: 'Ann',

            progress: 18

        }],

        pro: [{

        name: 'Sam',

        progress: 10

        }]

    }

};

// метод Object.values() - масив значений перечисляемых свойств

// метод Array.isArray() - возвращает true, если объэкт это масив и false, если не массив

function getTotalProgressByIteration (data) {

    let total = 0;

    let students = 0;

    for (let course of Object.values(data)){

        if (Array.isArray(course)) {

            students += course.length;

            for (let i = 0; i < course.length; i++) {

                total += course[i].progress;

            }

        } else {

            for (let subCourse of Object.values(course)) {

                students += subCourse.length;

                for (let i = 0; i < course.length; i++) {

                    total += subCourse[i].progress;

                }

            }

        }

    }

    return total / students;

    console.log(getTotalProgressByIteration(students);

}

//Примепр с использованием рекурсии

// база проверка на масив

function getTotalProgressByRecursion(data) {

    if (Array.isArray(data)) {

        let total = 0;

        for (let i = 0; i < data.length; i++) {

            total += data[i].progress;

        }

        return [total, data.length];

    } else {

        let total = [0, 0];

        for(let subData of Object.values(data)){

            const subDataArr = getTotalProgressByRecursion(subData);

            total[0] += subDataArr[0];

            total[1] += subDataArr[1];

        }

        return total;

    }

}

const result = getTotalProgressByRecursion(students);

console.log(result[0]/result[1]);