

1) Написать свою функцию для глубокого копирования объектов. При этом алгоритм должен осуществляться вручную (никаких вариантов с использованием готовых библиотек или `JSON.stringify() + JSON.parse()`, за это сразу 0 баллов за первую задачу). Функция должна называться `makeObjectDeepCopy`. Это важно. Принимать функция должна один параметр - объект, копию которого нужно сделать.

2) Написать функцию, которая принимает три аргумента: массив чисел, первое значение интервала, второе значение интервала. Результатом вызова функции должен быть массив из значений переданного массива (в качестве первого параметра), которые входят в этот интервал (концы включая). Если в качестве первого параметра передан не массив или массив, содержащий не только числа - кидать ошибку (`throw new Error(...)`). Если хотя бы одно из значений интервала (имеются ввиду параметры функции `selectFromInterval`) - невалидное число!, то также кидать ошибку. Если первое значение интервала < второго (имеются ввиду параметры функции `selectFromInterval`), то считать за интервал значения от первого до второго, в ином случае - от второго до первого.

Функция должна называться `selectFromInterval`.

> Примеры:

Вызываем функцию: `selectFromInterval([1,3,5], 5, 2)`

Получаем результат: `[3,5]`

Вызываем функцию: `selectFromInterval([-2, -15, 0, 4], -13, -5)`

Получаем результат: `[]`

Вызываем функцию: `selectFromInterval(['aaa'], 2, 3)`

Получаем результат: Ошибка!

3) Создать обычный объект и сделать его итерируемым. При этом итерация должна происходить следующим образом:

Должны поочерёдно проходиться все значения от свойства объекта `from`, до свойства `to`. (в случае если `to < from` - должна возникать ошибка).

Если `to` или `from` не указаны ИЛИ `to` или `from` не являются числами, должна возникать ошибка. Объект должен называться `myIterable`.

> Примеры:

```
const myIterable = { from: 1, to: 4 };
```

```
for (let item of myIterable) {  
  console.log(item); // 1, 2, 3, 4  
}
```

```
const myIterable = { from: 'aaa', to: 4 };
```

```
for (let item of myIterable) { // Ошибка!  
  console.log(item);  
}
```

В новом репозитории хочу видеть ОДИН файл, в котором должны лежать (как минимум) две функции (названные правильно) и объект с реализованной итерацией (также названный правильно).

Никаких логов, алертов, комментариев и т.п.