

SERIE JAVASCRIPT : OBJECTS

EXERCICE1 :

Given an object containing counts of both upvotes and downvotes, return what vote count should be displayed. This is calculated by subtracting the number of downvotes from upvotes.

Examples

```
getVoteCount({ upvotes: 13, downvotes: 0 }) → 13
```

```
getVoteCount({ upvotes: 2, downvotes: 33 }) → -31
```

```
getVoteCount({ upvotes: 132, downvotes: 132 }) → 0
```

EXERCICE2 :

Create a function that takes an object argument `sizes` (contains *width*, *length*, *height* keys) and returns the volume of the box.

Examples

```
volumeOfBox({ width: 2, length: 5, height: 1 }) → 10
```

```
volumeOfBox({ width: 4, length: 2, height: 2 }) → 16
```

```
volumeOfBox({ width: 2, length: 3, height: 5 }) → 30
```

Notes

- Don't forget to `return` the result.
- Remember that the values are in an object.
- Volume is `length * width * height`.

EXERCICE3 :

Given a number and an object with `min` and `max` properties, return `true` if the number lies within the given range (inclusive).

Examples

```
isInRange(4, { min: 0, max: 5 }) → true
isInRange(4, { min: 4, max: 5 }) → true
isInRange(4, { min: 6, max: 10 }) → false
isInRange(5, { min: 5, max: 5 }) → true
```

Notes

- Numbers can be positive or negative, and they may not be integers.
- You can assume `min <= max` is always true.

EXERCISE4 :

Given an array of scrabble tiles, create a function that outputs the maximum possible score a player can achieve by summing up the total number of points for all the tiles in their hand. Each hand contains 7 scrabble tiles.

Here's an example hand:

```
[
  { tile: "N", score: 1 },
  { tile: "K", score: 5 },
  { tile: "Z", score: 10 },
  { tile: "X", score: 8 },
  { tile: "D", score: 2 },
  { tile: "A", score: 1 },
  { tile: "E", score: 1 }
]
```

The players maximumScore from playing all these tiles would be 1 + 5 + 10 + 8 + 2 + 1 + 1, or 28.

Examples

```
maximumScore([
  { tile: "N", score: 1 },
  { tile: "K", score: 5 },
  { tile: "Z", score: 10 },
  { tile: "X", score: 8 },
  { tile: "D", score: 2 },
  { tile: "A", score: 1 },
  { tile: "E", score: 1 }
]) → 28
```

```
maximumScore([
  { tile: "B", score: 2 },
  { tile: "V", score: 4 },
  { tile: "F", score: 4 },
  { tile: "U", score: 1 },
  { tile: "D", score: 2 },
  { tile: "O", score: 1 },
  { tile: "U", score: 1 }
])
```

]) → 15

Notes

Here, each tile is represented as an object with two keys: tile and score.

EXERCISE5 :

Given three arguments — an object `obj` of the stolen items, the pet's `name` and a `value` — return an object with that name and value in it (as key-value pairs).

Examples

```
addName({}, "Brutus", 300) → { Brutus: 300 }
```

```
addName({ piano: 500 }, "Brutus", 400) → { piano: 500, Brutus: 400 }
```

```
addName({ piano: 500, stereo: 300 }, "Caligula", 440) → { piano: 500, stereo: 300, Caligula: 440 }
```

Notes

The `value` argument will be a number.