

Задание №4:

1) Реализовать функцию `concatStrings`, которая может быть вызвана следующим образом: `concatStrings('first')('second')('third')()`.

Результатом вызова данной функции должна являться новая строка, содержащая все переданные таким образом строки.

Если одно из значений является невалидной строкой (пустая строка - это валидная строка), то возвращать результат, полученный до текущего момента (ошибок не бросать!)

Кроме этого добавить функции второй необязательный параметр - `separator`. Он также должен являться валидной строкой, однако в случаях, когда вместо валидной строки на его место передано что-то ещё - запускаем функцию как будто без него вообще (иными словами игнорируем, никаких ошибок кидать не нужно). Если же всё-таки параметр был валидной строкой, то результирующая строка должна содержать все переданные строки, разделённые значением `separator`.

> Примеры:

Вызываем функцию: `concatStrings('first')('second')('third')()`

Получаем результат: `'firstsecondthird'`

Вызываем функцию: `concatStrings('first', null)('second')()`

Получаем результат: `'firstsecond'`

Вызываем функцию: `concatStrings('first', '123')('second')('third')()`

Получаем результат: `'first123second123third'`

Вызываем функцию: `concatStrings('some-value')('')('')(null)`

Получаем результат: `'some-value'`

Вызываем функцию: `concatStrings('some-value')(2)`

Получаем результат: `'some-value'`

Вызываем функцию: `concatStrings('some-value')('333')(123n)`

Получаем результат: `'some-val333'`

2) Создать класс `Calculator`. Конструктор класса должен принимать два валидных числа, иначе (если параметра не два или хотя бы один из них невалидный `number`) бросать ошибку (`throw new Error("")`). Данный класс должен иметь методы `setX`, `setY`, `logSum`, `logMul`, `logSub`, `logDiv`.

- `setX(num)` - задаёт первому из чисел новое значение. Кидать ошибку если параметр не передан или является невалидным числом;

- `setY(num)` - задаёт второму из чисел новое значение. Кидать ошибку если параметр не передан или является невалидным числом;

- `logSum()` - выводит в консоль сумму двух наших чисел внутри класса;

- `logMul()` - выводит в консоль произведение двух наших чисел внутри класса;

- `logSub()` - выводит в консоль разность двух наших чисел внутри класса;

- `logDiv()` - выводит в консоль частное двух наших чисел внутри класса ИЛИ кидает ошибку, если второе число (`Y`) равняется нулю.

ВАЖНО! Все методы класса должны отрабатывать корректно **ДАЖЕ** в случае копирования функций в отдельные переменные.

> Пример:

```
const calculator = new Calculator(12, 3);
```

```
calculator.logSum(); // 15  
calculator.logDiv(); // 4  
calculator.setX(15);  
calculator.logDiv(); // 5  
const logCalculatorDiv = calculator.logDiv;  
logCalculatorDiv(); // 5  
calculator.setY(444n); // Ошибка!
```

Функцию и класс называем так, как написано в задании. Проверять буду тестами.

P.s. Infinity, -Infinity и NaN - это невалидные числа (делаю такую пометку в первый и последний раз).

Задание новое, так что там есть ошибки и нужно будет ещё дополнять условия