

Курс:
«Язык сценариев JavaScript и библиотека jQuery»»

Модуль 9. ECMAScript

Модуль 7. Модульное тестирование

Задание 1

Реализовать страницу со списком продуктов и возможностью их сортировать.

В JS коде вам необходимо реализовать лишь одну (!) функцию сортировки, которая принимает 2 параметра: массив для сортировки и функцию-компаратор, которая указывает, как сравнивать объекты переданного массива.

Сигнатура функции: `mysort(arr, cmp)`, где `arr` – массив для сортировки, а `cmp` – компаратор.

Компаратор – это функция, которая принимает два объекта и возвращает логический результат: `true` – если первый объект меньше, чем второй, и `false` – если первый объект больше, чем второй.

При нажатии на кнопку **Sort** необходимо вызывать функцию сортировки, передавая в нее массив продуктов и необходимый компаратор (в зависимости от выбранного параметра в выпадающем списке).

Отдельно описывать функции-компараторы нет необходимости, передавайте их используя синтаксис стрелочных функций.

Products list

Type of sort:

Id

Sort

#1 APPLE 1.02\$	#2 PEAR 1.52\$	#3 BANANA 2.00\$	#4 MANGO 5.60\$	#5 ORANGE 2.35\$	#6 LIME 3.90\$
#7 APRICOT 4.10\$	#8 AVOCADO 5.90\$	#9 PAPAYA 7.00\$	#10 RASPBERRY 4.60\$	#11 LEMON 3.45\$	

Products list

Type of sort:

Price

Sort

#1 APPLE 1.02\$	#2 PEAR 1.52\$	#3 BANANA 2.00\$	#5 ORANGE 2.35\$	#11 LEMON 3.45\$	#6 LIME 3.90\$
#7 APRICOT 4.10\$	#10 RASPBERRY 4.60\$	#4 MANGO 5.60\$	#8 AVOCADO 5.90\$	#9 PAPAYA 7.00\$	

Задание 2

Реализовать класс **Fraction** и тесты для его методов.

Класс **Fraction** описывает дробь.

Поля класса:

- числитель (целое число);
- знаменатель (целое число).

Методы класса:

- `constructor(numerator, denominator)` – принимает числитель и знаменатель и сохраняет их;
- `reduction()` – изменяет объект, от которого была вызвана функция, таким образом, чтобы у числителя и знаменателя не осталось общих делителей;
- `isInteger()` – возвращает `true`, если при делении числителя на знаменатель, будет получено целое число, иначе возвращает `false`;
- `static addition(fraction1, fraction2)` – принимает две дроби, складывает их и возвращает результат в виде объекта типа **Fraction**;
- `static subtraction(fraction1, fraction2)` – принимает две дроби, вычитает вторую из первой и возвращает результат в виде объекта типа **Fraction**;
- `static multiplication(fraction1, fraction2)` – принимает две дроби, перемножает их и возвращает результат в виде объекта типа **Fraction**;
- `static division(fraction1, fraction2)` – принимает две дроби, делит первую на вторую и возвращает результат в виде объекта типа **Fraction**.

Данные для тестирования методов:

Методы	Тесты
addition	$2/5 + 1/5 = 3/5$ $3/5 + 4/5 = 7/5$ $2/5 + 3/7 = 29/35$ $1/2 + 1/8 = 5/8$
subtraction	$2/5 - 1/5 = 1/5$ $3/5 - 4/5 = -1/5$ $2/5 - 1/7 = 9/35$ $5/8 - 1/2 = 1/8$
multiplication	$1/4 * 3/4 = 3/16$ $2/3 * 1/4 = 1/6$
division	$1/4 : 3/4 = 1/3$ $2/3 : 1/4 = 8/3$
reduction	$2/5 = 2/5$ $24/75 = 8/25$
isInteger	$2/7 = \text{false}$ $4/4 = \text{true}$ $4/1 = \text{true}$ $10/3 = \text{false}$

Если при написании тестов вы будете использовать библиотеки **mocha** и **chai**, то результат может выглядеть так:

Fraction, addition()

- ✓ new Fraction(2, 5) + new Fraction(1, 5) = new Fraction(3, 5)
- ✓ new Fraction(3, 5) + new Fraction(4, 5) = new Fraction(7, 5)
- ✓ new Fraction(2, 5) + new Fraction(3, 7) = new Fraction(29, 35)
- ✓ new Fraction(1, 2) + new Fraction(1, 8) = new Fraction(5, 8)