**Упражнения**

**Идентификаторы**

1. Объявите переменную и запишите в нее свое имя как литерал строки.
2. Объявите константу и запишите в нее год своего рождения как литерал числа.
3. Создайте функцию, которая печатает приветствие и имеет один аргумент: name.

**Циклы**

1. Реализуйте функцию range(start: number, end: number): array которая отдает массив чисел из диапазона [15, 30] включая крайние числа.
2. Реализуйте функцию rangeOdd(start: number, end: number): array которая отдает массив нечетных чисел из диапазона [15, 30] включая крайние числа.

**Функции**

1. Вложенные вызовы функций в цикле

* Реализуйте функцию average с сигнатурой average(a: number, b: number): number вычисляющую среднее арифметическое своих аргументов.
* Реализуйте функцию square с сигнатурой square(x: number): number вычисляющую квадрат своего аргумента.
* Реализуйте функцию cube с сигнатурой cube(x: number): number вычисляющую куб своего аргумента.
* Вызовите функции square и cube в цикле от 0 до 9, вычисляя, соответственно квадрат и куб от переменной цикла. Передайте квадрат и куб на каждой итерации в функцию average. Результаты сложите в массив и возвратите из функции calculate.

**Объекты**

1. Выполнить следующие пункты внутри функции fn (см. заготовку: 7-objects.js)

* Создайте объект с одним полем name и присвойте его в константу.
* Создайте объект с одним полем name и присвойте его в переменную.
* Попробуйте поменять поле name у обоих объектов.
* Попробуйте присвоить другой объект в оба идентификатора.
* Объясните поведение кода и оставьте только рабочий код.

1. Реализуйте функцию createUser с сигнатурой createUser(name: string, city: string): object. Пример вызова: createUser('Marcus Aurelius', 'Roma') функция должна вернуть объект { name: 'Marcus Aurelius', city: 'Roma' }

**Коллекции**

1. Реализуйте телефонную книгу на массиве объектов.

* Объявите массив объектов с двумя полями: name и phone для хранения телефонной книги. Пример объекта: { name: 'Marcus Aurelius', phone: '+380445554433' } и добавьте несколько объектов в массив, чтоб было на чем проверять.
* Реализуйте функцию findPhoneByName с сигнатурой findPhoneByName(name: string): string. При вызове функция должна находить объект, где поле name равно аргументу name и возвращать номер телефона из объекта. A. Реализуйте телефонную книгу на хеш-таблицах, т.е. справочниках (объектах).
* Задайте справочник (объект) с ключами равными значениям поля name (из предыдущего примера) и значениями равными полю phone.
* Реализуйте функцию findPhoneByName с сигнатурой findPhoneByName(name: string): string которая находит телефон в хеше по имени и возвращает номер телефона. Используйте hash[key] для поиска телефона.