

# Задания для знакомства с typescript

---

1. Создайте интерфейс `User` с полями:

- `id` (number)
- `name` (string)
- `email` (string, опционально)
- `isActive` (boolean, по умолчанию `true`).

Напишите функцию `createUser`, которая принимает эти поля и возвращает объект пользователя.

2. Создайте интерфейс `Book` с полями:

- `title` (обязательная строка)
- `author` (обязательная строка)
- `year?` (опциональное число)
- `genre` (строка, но которая может быть только `'fiction'` или `'non-fiction'`). Используйте Union Type для `genre`.

Напишите функцию `createBook(book: Book): Book`, которая возвращает объект книги.

Продемонстрируйте вызов функции с передачей объекта и без опционального поля `year`.

3. Напишите функцию `calculateArea`, которая:

- Принимает `shape` (тип: `'circle' | 'square'`) и параметры (`radius` для круга, `side` для квадрата).
- Возвращает площадь фигуры.
- Используйте перегрузку функций.

4. Создайте тип `Status`, который может принимать одно из трех строковых значений: `'active'`, `'inactive'`, или `'new'`. Напишите функцию `getStatusColor(status: Status): string`, которая возвращает цвет (строку) для каждого статуса (например, `'green'` для `'active'`).

5. Объявите тип функции `StringFormatter`.

- Он должен описывать функцию, которая принимает строку и необязательный параметр `uppercase: boolean` (по умолчанию `false`).
- Функция возвращает строку.
- Напишите две функции под этот тип: одна делает первую букву заглавной, а вторая обрезает пробелы по краям и, если `uppercase` is `true`, приводит всю строку к верхнему регистру.

6. Напишите функцию `getFirstElement<T>(arr: T[]): T | undefined`, которая возвращает первый элемент массива любого типа или `undefined`, если массив пустой. Протестируйте ее на массиве чисел и массиве строк.

7. Создайте интерфейс `HasId` с полем `id: number`. Напишите универсальную функцию `findById<T extends HasId>(items: T[], id: number): T | undefined`, которая ищет объект по `id` в массиве объектов, у которых есть это поле.