

## Задание 1.

Написать сервис userService (factory), который будет содержать информацию о пользователе (currentUser).

Сервис должен предоставлять методы для установки текущего пользователя (setCurrentUser) и для чтения текущего пользователя (getCurrentUser)

Сервис оформить в модуле services.

Создать модуль app с контроллером Simple. Подключить зависимый модуль services. Подключить приложение к HTML. Контроллер должен выводить в консоль созданный сервис. Установить пользователя Vitaliy, получить пользователя и вывести в консоль.

## Задание 2.

Создать сервис personService (service) на основе функции-конструктора

```
function Person() {}
Person.prototype.setName = function(name) {
    this.name = name;
};
Person.prototype.setAge = function(age) {
    this.age = age;
};
Person.prototype.getName = function() {
    return this.name;
};
Person.prototype.getAge = function() {
    return this.age;
};
```

В контроллере Simple вывести данный сервис в консоль. Установить возраст 16, получить возраст и вывести в консоль.

## Задание 3.

Создать конфигурационный сервис userGreetingService (provider) приветствия пользователя.

Сервис по умолчанию должен выводить с помощью метода greet(user) приветствие "Hi, user" в консоль.

Реализовать метод configGreeting для настройки приветственной фразы, например "Hello", "Good morning", .... Если фраза не задана, возвращать текущую.

Настроить сервис на использование "Hello".

В контроллере Simple поприветствовать пользователя Vitaliy.

## Задание 4.

Создать сервисы констант defaultGreeting = "Hi", officialGreeting = "Good morning" и использовать их для реализации и конфигурирования сервиса userGreetingService

### Задание 5.

Создать сервис значения `userName = "Vitaliy"`. Использовать его в контроллере `Simple`.

### Задание 6.

Используя сервис `$provide` декорировать сервис `userService`, функцией `userServiceLog`.

Декорированный сервис должен выводить в консоль время доступа к методам сервиса `userService`