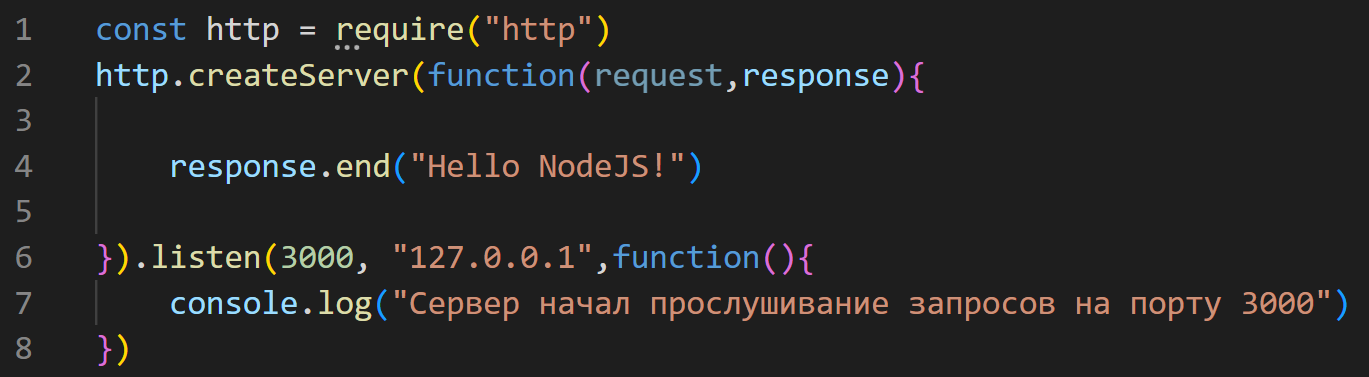
**КТ № 2. Модули. Глобальные объекты.**

**Время на выполнение практической работы:** 4 часа.

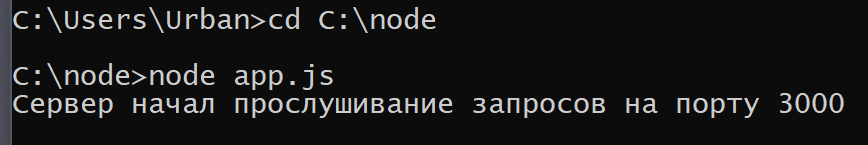
**Теоретические сведения к выполнению практической работы:**

Напишем первое простейшее приложение для NodeJS. Для создания приложений можно использовать практически все стандартные конструкции языка JavaScript. Исключением является работа с DOM, так как приложение будет запускаться на сервере, а не в браузере, поэтому DOM и такие объекты как window или document в данном случае нам будут недоступны.

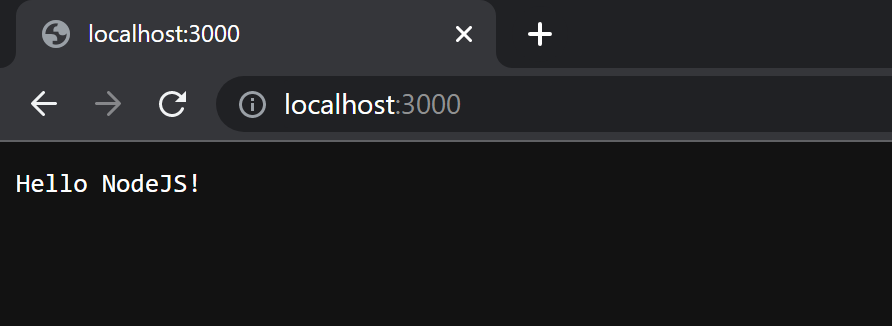
Создаем каталог на жестком диске. В каталоге создаем файл app.js, содержащий в себе следующий код, который создает сервер для прослушивания входящих подключений и обработки запросов.



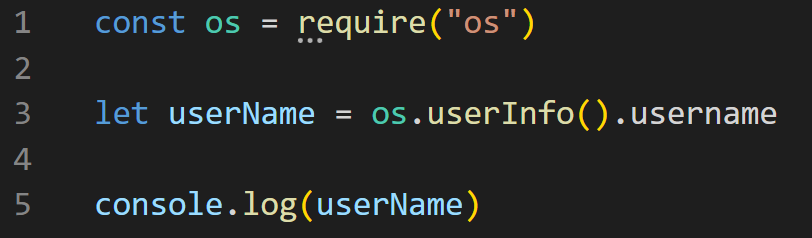
Запускаем сервер.



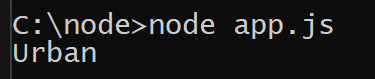
Делаем проверку в браузере.



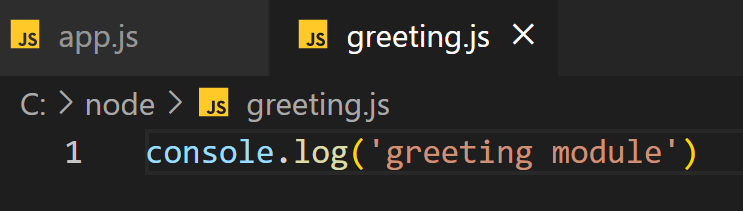
Подключаем модуль os, запрашиваем информацию о пользователе, затем выводим.



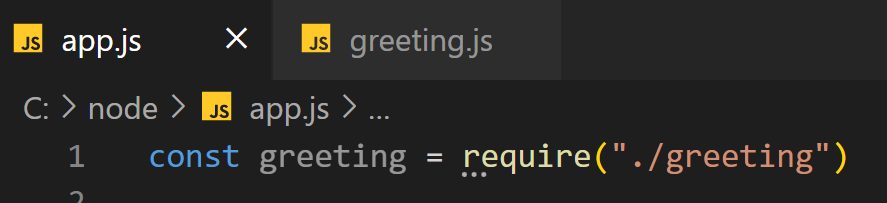
Проверка правильности работы кода.



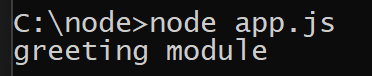
Создаем собственный модуль.



Подключаем собственный модуль.



Проверка через командную строку.

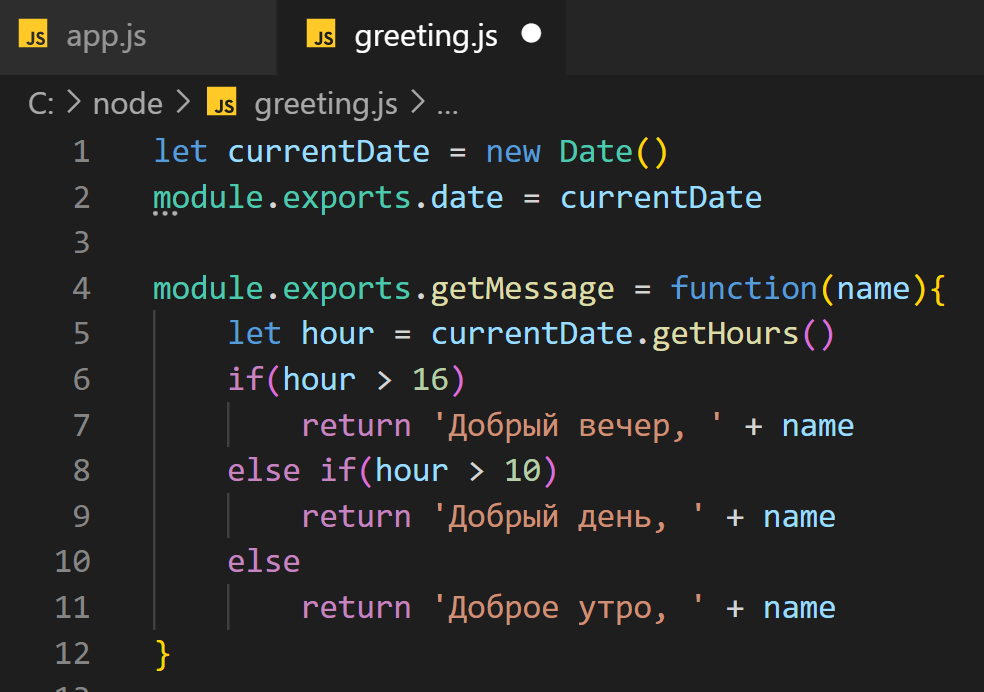


Теперь изменим файл greeting.js.

Здесь определена переменная currentDate. Однако из вне она недоступна. Она доступна только в пределах данного модуля. Чтобы такие переменные или функции модуля были доступны, необходимо определить их в объекте module.exports. Объект module.exports - это то, что возвращает функция require() при получении модуля.

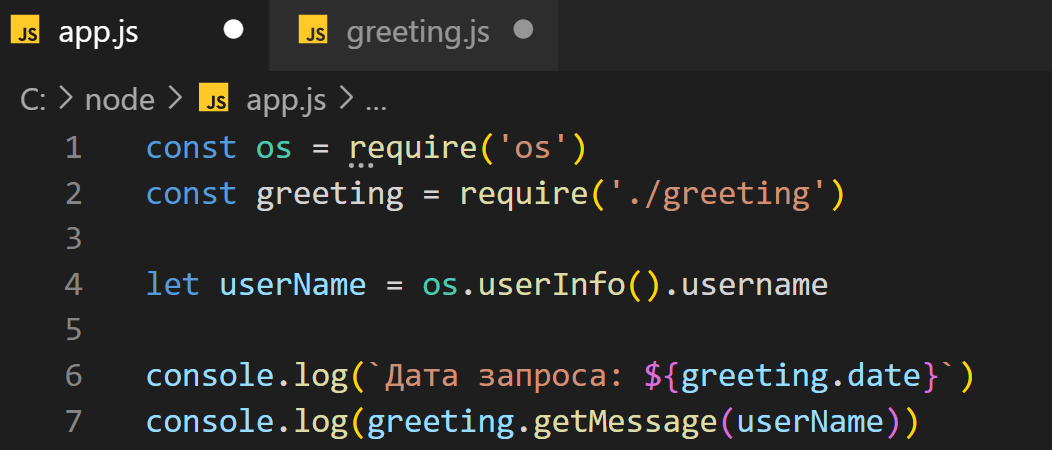
Вообще объект module представляет ссылку на текущий модуль, а его свойство exports определяет все свойства и методы модуля, которые могут быть экспортированы и использованы в других модулях.

В частности, здесь определяется свойство date и метод getMessage, который принимает некоторый параметр.

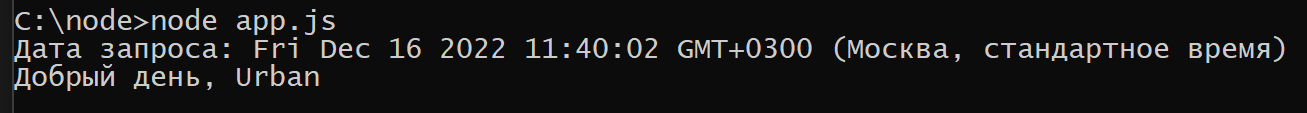


Далее изменим файл app.js.

Все экспортированные методы и свойства модуля доступны по имени: greeting.date и greeting.getMessage().

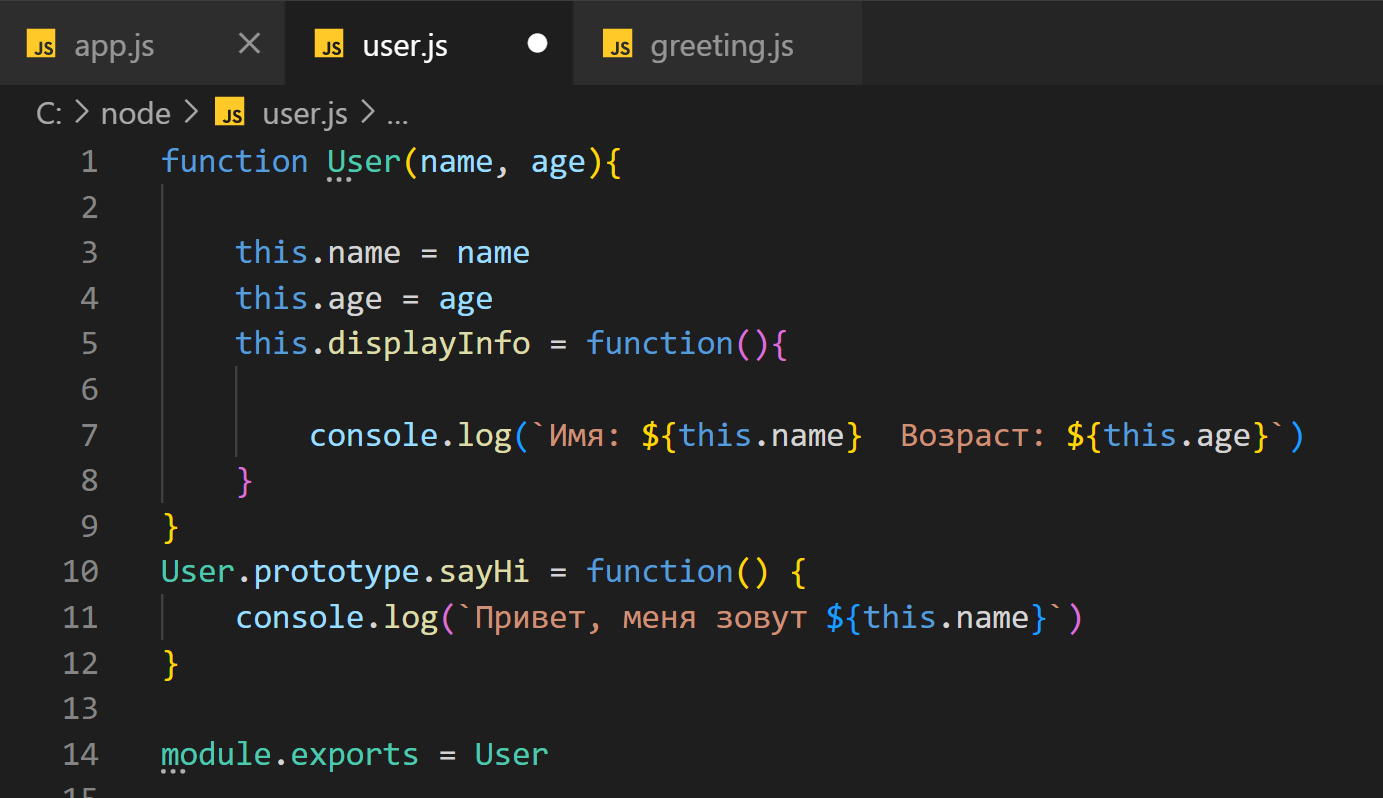


Перезапустим приложение.

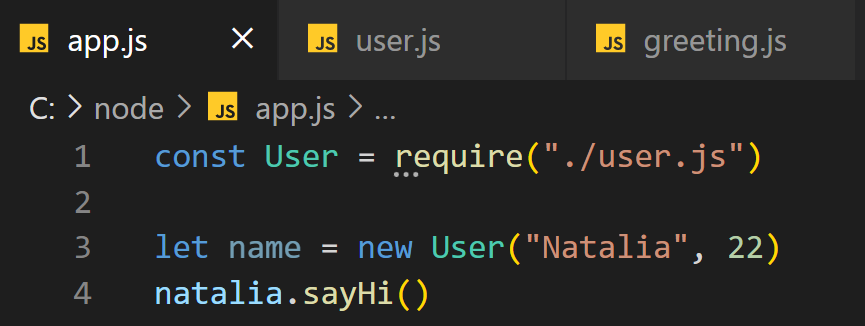


**Определение конструкторов и объектов в модуле**

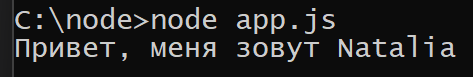
Кроме определения простейших функций или свойств в модуле могут определяться сложные объекты или функции конструкторов, которые затем используются для создания объектов. Так, добавим в папку проекта новый файл user.js.



Подключим и используем этот модуль в файле app.js.

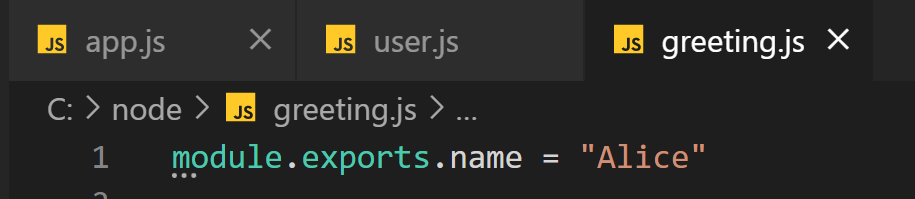


Запустим приложение.



Рассмотрим некоторые аспекты работы с модулями в Node.js. Прежде всего надо отметить, что подключаемые модули кэшируются.

Это, с одной стороны, увеличивает производительность, а с другой, может создать некоторые проблемы, если мы не будем учитывать этот аспект. Например, изменим файл greeting.js следующим образом:

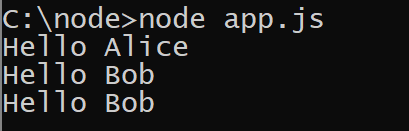


В файле определена только одна строка, которая устанавливает свойство name.

Изменим код файла app.js



Несмотря на то, что здесь два раза получаем модуль с помощью функции require, но обе переменных - greeting1 и greeting2 будут указывать на один и тот же объект.



**Структура модулей**

Нередко модули приложения образуют какие-то отдельные наборы или области. Такие наборы модулей лучше помещать в отдельные каталоги. Например, создадим в каталоге приложения подкаталог welcome и создадим в нем три новых файла:

index.js

morning.js

evening.js

В итоге общая структура проекта пусть будет выглядеть следующим образом:

welcome

index.js

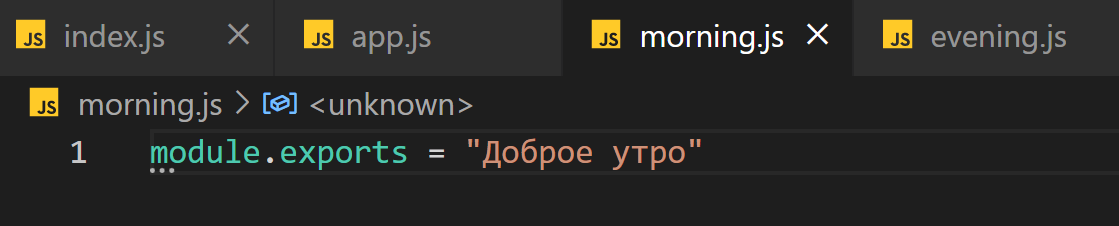
morning.js

evening.js

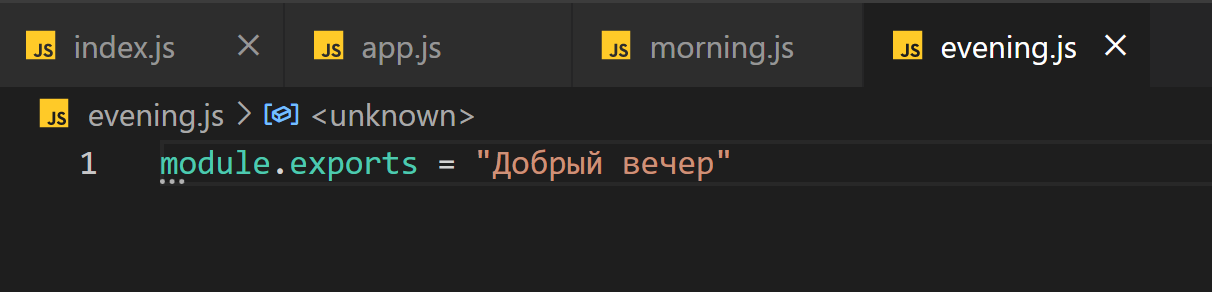
app.js

greeting.js

В файл morning.js поместим следующую строку:

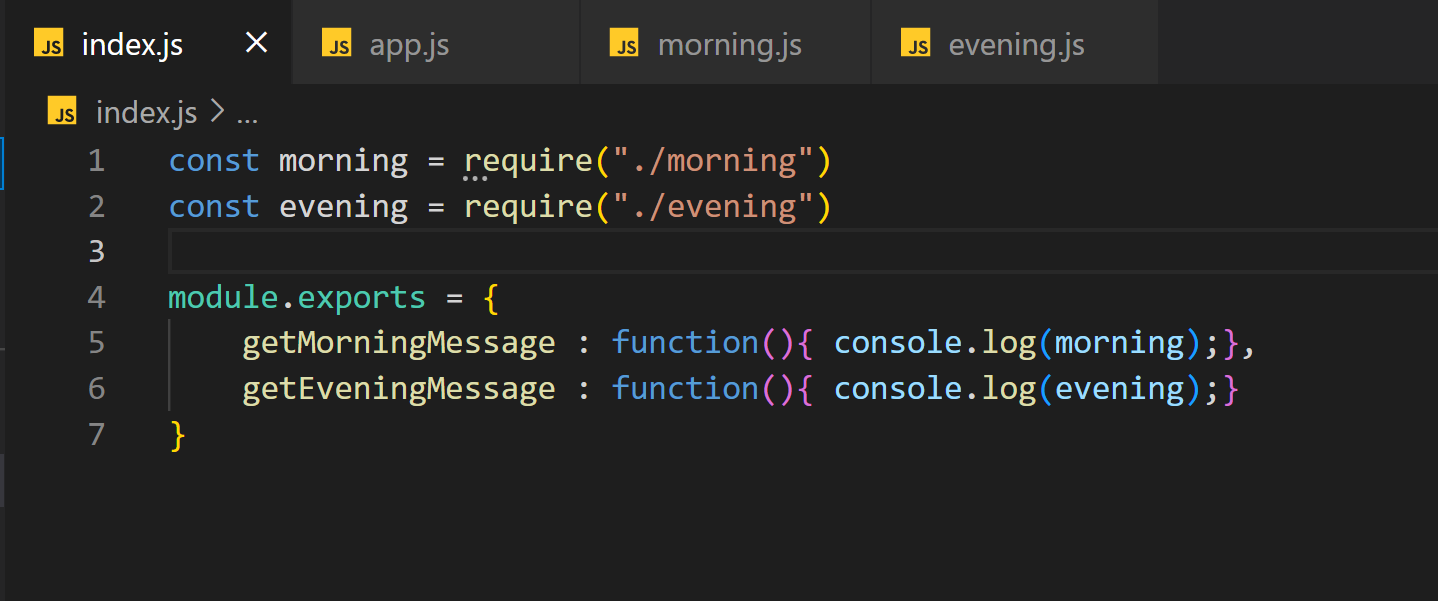


Аналогично изменим файл evening.js



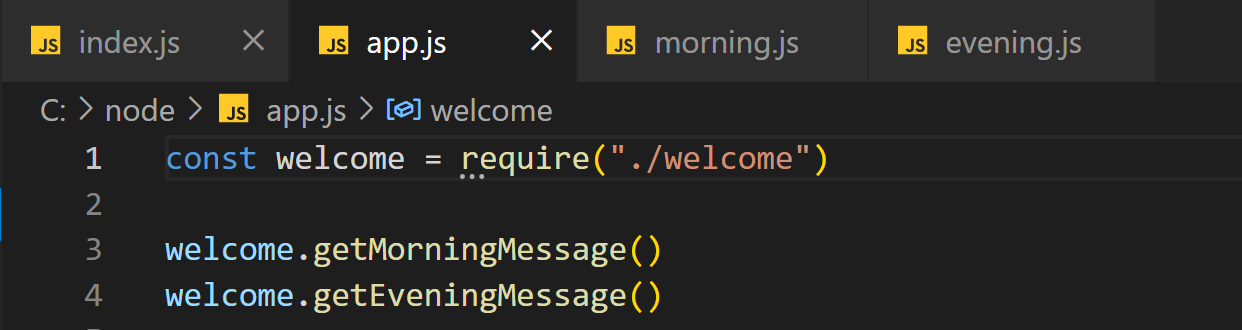
Эти два файла определяют сообщения приветствия в зависимости от времени суток.

И определим в файле index.js следующий код



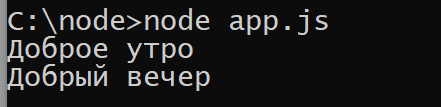
В модуле определен объект, который имеет две функции для вывода приветствий.

Теперь используем этот модуль в файле app.js:



Несмотря на то, что нет такого файла как welcome.js, но если в проекте есть каталог, который содержит файл с именем index.js, то мы можем обращаться к модулю по имени каталога, как в данном случае.

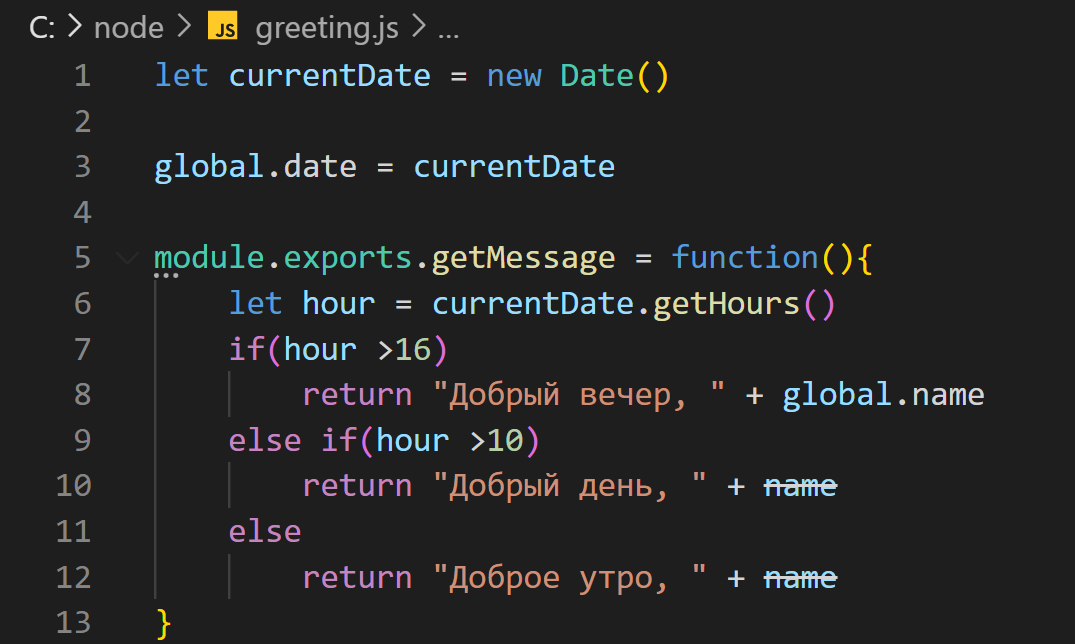
Запустим приложение, и на консоль будут выведены оба приветствия:



**Объект global и глобальные переменные**

Node.js предоставляет специальный объект global, который предоставляет доступ к глобальным, то есть доступным из каждого модуля приложения, переменным и функциям. Примерным аналогом данного объекта в javascript для браузера является объект window. Все доступные глобальные объекты можно посмотреть в документации.

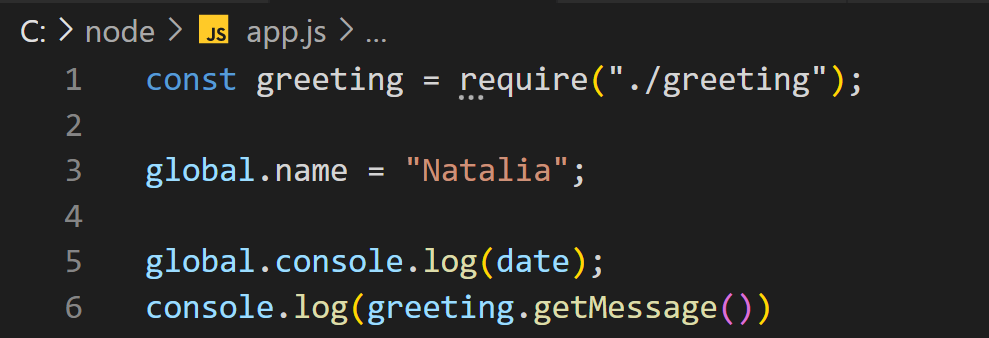
Для примера создадим следующий модуль greeting.js



Здесь, во-первых, происходит установка глобальной переменной date: global.date = currentDate;

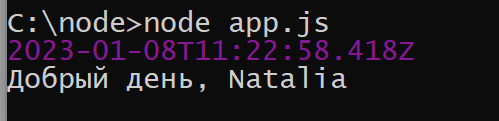
Во-вторых, в модуле получаем глобальную переменную name, которая будет установлена из вне. При этом обратиться к глобальной переменной name мы можем через объект global: global.name, либо просто через имя name, так как переменная глобальная.

Определим следующий файл приложения app.js



Здесь устанавливаем глобальную переменную name, которую мы получаем в модуле greeting.js. И также выводим на консоль глобальную переменную date. Причем все глобальные функции и объекты, например, console, также доступны внутри global, поэтому мы можем написать и global.console.log(), и просто console.log().

Запустим файл app.js



Однако по возможности всё-таки рекомендуется избегать определения и использования глобальных переменных, и преимущественно ориентироваться на создание переменных, инкапсулированных в рамках отдельных модулей.

**Этапы выполнения:**

***Требования к выполнению задания:***

1. ***Создать новый файл app.js;***
2. ***Подключить модуль http;***
3. ***Создать новый сервер для прослушивания входящих подключений и обработки запросов;***
4. ***Запустить сервер через командную строку;***
5. ***Открыть браузер и ввести в адресную стоку адрес http://localhost:3000/;***
6. ***Выполнить код из созданного файла при помощи командной строки;***
7. ***При помощи модуля os получить имя текущего пользователя;***
8. ***Создать собственный модуль, который будет содержать в себе стрелочную функцию. Результатом работы функции должен быть синус числа х;***
9. ***Вывести в командной строке дату и время запроса.***
10. ***Вывести приветствие с обращением по имени пользователя ПК. Если приветствие выводится утром, должно быть выведено «доброе утро», если днём - «добрый день», если вечером – «добрый вечер».***
11. ***Вывести приветствие с обращением по имени. Имя должно быть определено в отдельном файле;***
12. ***Вывести приветствие с обращением к двум разным людям, пользуясь кэшированием модулей;***
13. ***Поработайте со структурой модулей: создайте три новых файла в подкаталоге, затем выведите два приветствия;***
14. ***Установите глобальную переменную date;***
15. ***Получите глобальную переменную name, которая установлена извне;***
16. ***Выведите глобальную переменную на консоль.***