# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**CSC174 server-side javascript**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Unit 01 LAB: A simple hello node.js program

# Objectives

In this lab assignment, students will learn:

* How to install and setup Node.js development environment
* How to use WebStorm IDE or Visual Studio IDE to do Node.js development
* How to create, edit and run a simple Hello Node.js program
* Understand the difference between client-side JavaScript and server-side JavaScript development

# COURSE PREPARATION

You should have done your reading assignment listed under “Reading Assignment” and “Video Assignment” sections in BlackBoard. You should also have reviewed the lecture slides in BlackBoard. There is an optional section called “In Case You Don’t Know” in BlackBoard for those who have limited exposure to JavaScript language.

# WHat to submit

For this lab you need to submit the following files:

* **hello.js** (from Part One)
* **app.js** (from Part Two)

# grading rubric:

Be sure to follow the Coding Standard Guidelines. You must properly indent and comment your code. This assignment is worth 100 points. There is an additional 5 points for bonus.

* Indent code and insert comments to document your program. [10 pts]
* Program must be implemented and run with no syntax errors. [40 pts]
* Program must be implemented and run with no logic errors. [40 pts]
* Required source files should be zipped and uploaded to BlackBoard assignment drop box before the deadline. [10 points]

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**PART ONE: A client-side Hello JavaScript Program**

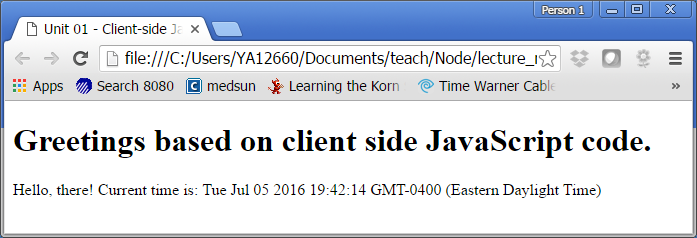
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Project Description:**

You are given a Hello HTML web page with embedded client-side JavaScript code that prints Hello and the current time.

The instructor should provide you the source code in ***unit01\_lab\_part1.zip*** file in BlackBoard Unit 01 Lab sections. There are two files in this project: ***hello.html*** and ***hello.js***.

When you open the hello.html file in a browser, you should see something like this:



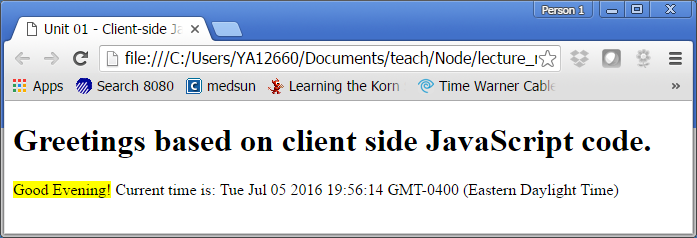
Modify this program so that it will print “Good Morning!”, “Good Afternoon!” or “Good Evening” instead according to the following logic:

Midnight till noon: Good morning (midnight inclusive, noon exclusive)

Noon till 5:00 PM: Good afternoon (noon inclusive, 5:00 PM exclusive)

5:00 PM till midnight: Good evening (5:00 PM inclusive, midnight exclusive)

A sample run of this project will look something like this:



You only need to submit the modified hello.js file.

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**PART TWO: A server-side Hello JavaScript program using Node.js**

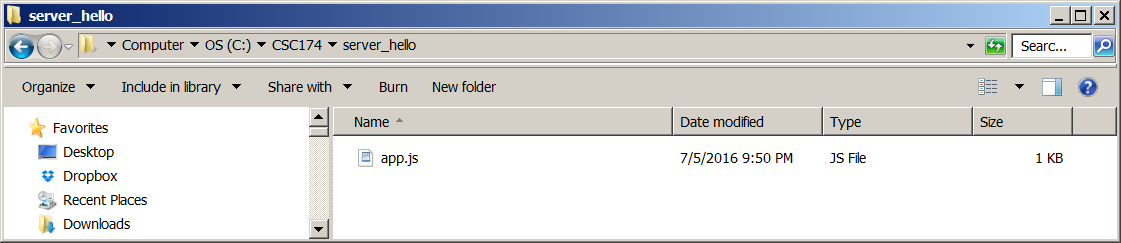
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Project Description:**

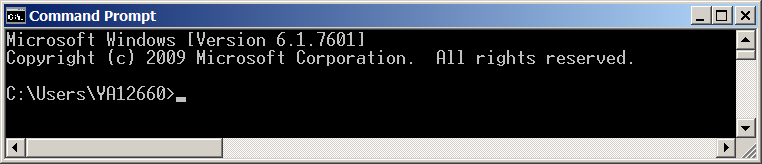
You are given a Hello web site implemented with Node.js engine on your machine. The web site address is <http://localhost:3333/>. (If your machine name is ***trump***, you can access this web site from your home network at <http://trump:3333/>.)

The instructor should provide you the source code in ***unit01\_lab\_part2.zip*** file in BlackBoard Unit 01 Lab sections. There is only one file in this project: ***app.js***.

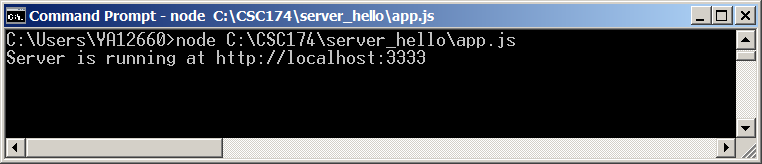
You can run this Node.js application from a command line window or from an IDE such as WebStorm. Let’s assume you have downloaded server\_hello.zip file under your local disk location: ***C:\CSC174\server\_hello*** folder.



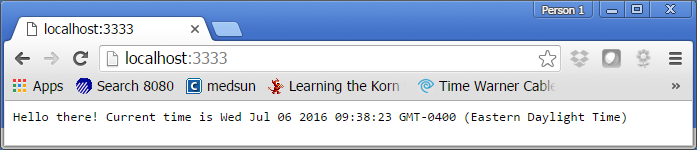
If you have Node.js engine installed according to textbook Chapter 1 and BlackBoard software installation guide, you should be able to run this program by opening a Window command line window or MacOS terminal window. Below is a Windows command prompt window example:



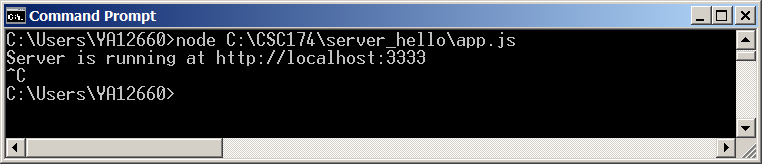
Enter command: **node C:\CSC174\server\_hello\app.js** and you should see this:



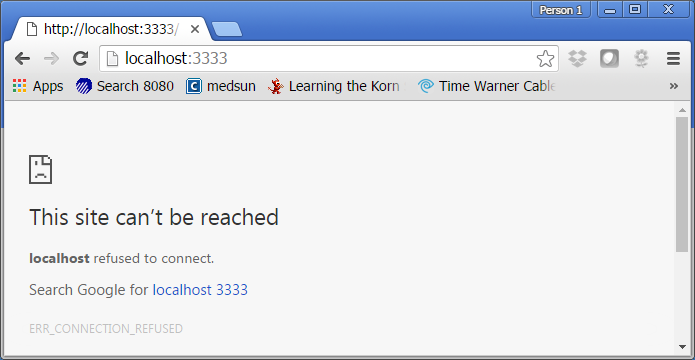
That’s it. Your Node.js powered Hello web site is up running on your local machine port 3333. To verify it’s working properly, open a browser and enter the URL: <http://localhost:3333/> or <http://andrews-pc:3333/> if your machine name is andrews-pc. If your machine is open to the internet (outside your home network), you have got a running web site that everyone can access now.



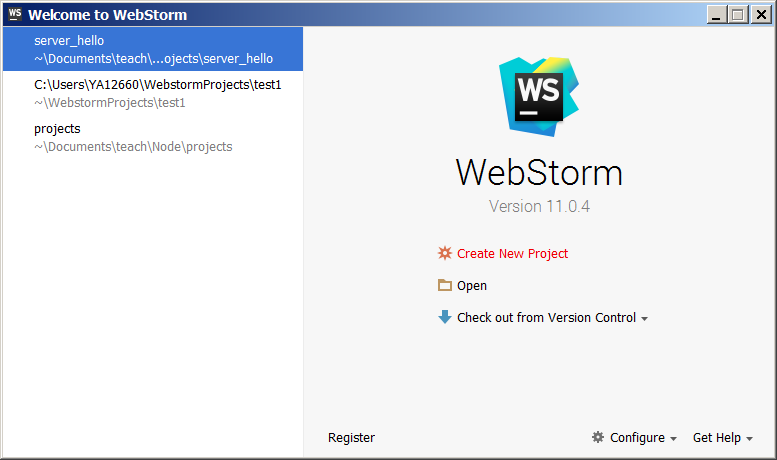
You can stop this Node.js program by pressing <ctrl>+ C keys at the same time like this:



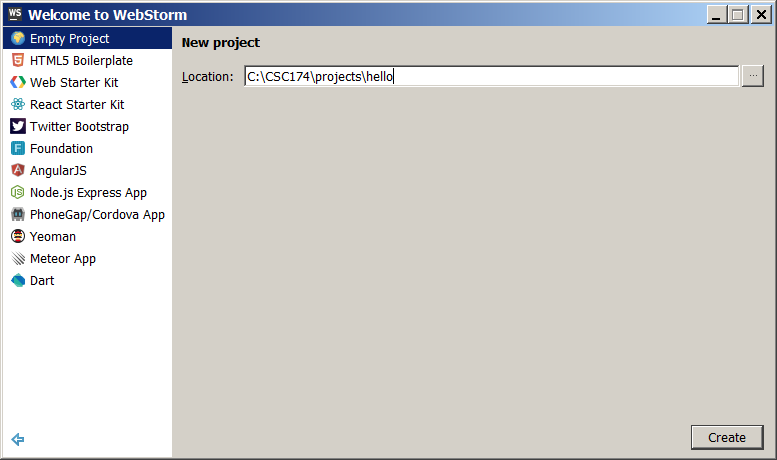
Now if you want to access this web site you will get the following error:



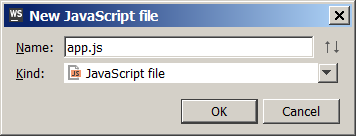
Another way to run your Node.js program is to run it in an IDE. I highly recommend WebStorm. It will make your life a lot easier. It’s free to all students. Let’s create a new project in WebStorm for server\_hello project.

Step 1: Start WebStorm  


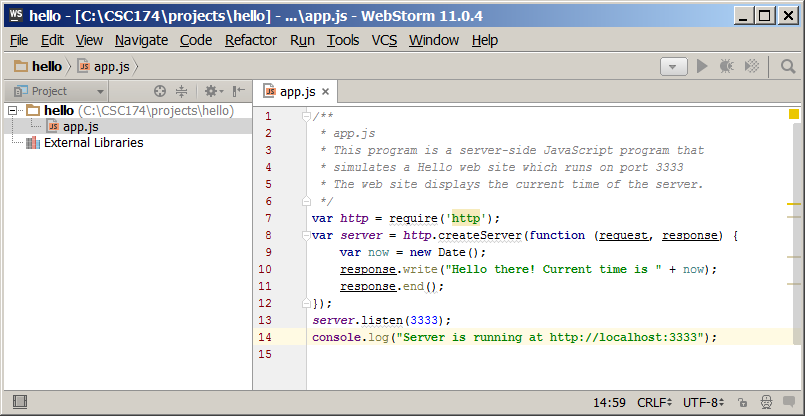
Step 2: Create New Project called “hello” under C:\CSC174\projects\hello.



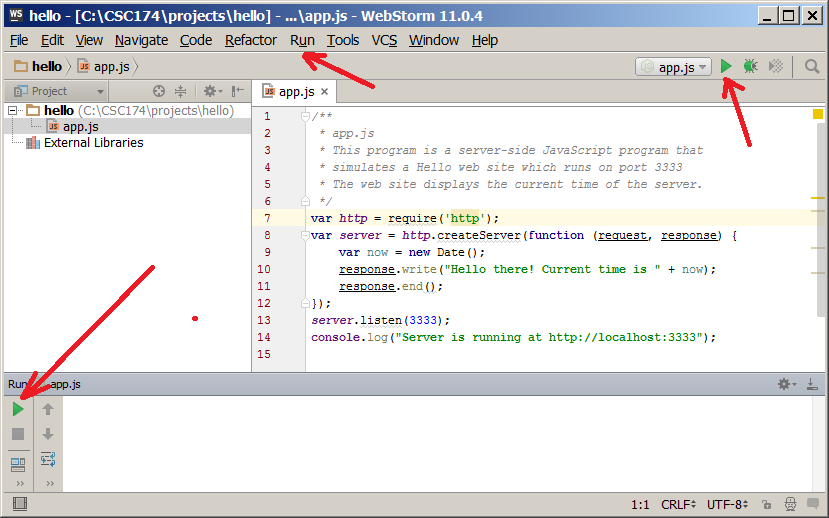
Step 3: Add a new JavaScript file called **app.js** to the hello project folder by right click on the project folder name -> new -> JavaScript file.



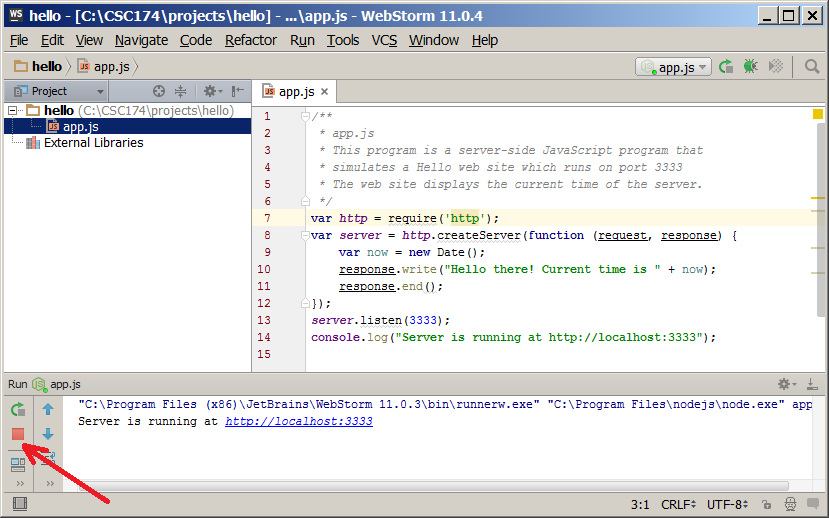
Step 4: Copy paste the content of app.js provided by the instructor.



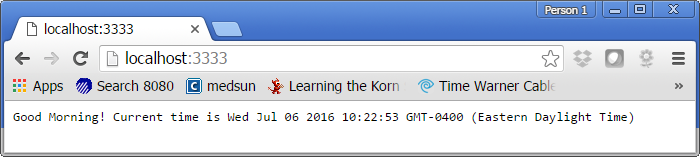
Step 5: Run this application by clicking “Run” on the top menu or <ctrl> + <shift> + F10 or the green triangle at the bottom left corner or top right corner.



You can verify by entering the URL in a web browser. To stop this program, simply click the red square button on the bottom left corner.



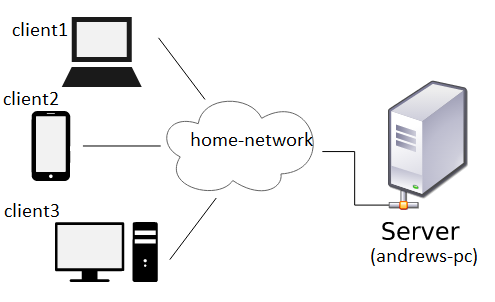
Now modify this program and make the greetings to be Good Morning, Good Afternoon and Good Evening according to the same logic as in Part One of this lab. A sample screenshot can be found below:



You only need to submit the modified app.js file.

# After thought

You might be wondering why we are doing two versions of this project. I would like you to see that JavaScript code can be run on the client-side (inside a web browser) and on the server-side (inside a Node.js V8 engine developed by Google). Let’s say you have home network like below:



With client-side JavaScript code, the program is running on each of the client devices. Each client needs to get a copy of this source code and run it in its own environment. The time displayed is the system time of that device.

With server-side JavaScript code, the program is running on the server (in this case a machine called andrews-pc). When a client access this web site hosted on this machine, the time displayed on the web page is from this server. It’s the same for all clients. There is no processing on the client side. The client sends an HTTP request to the server and the server returns the data.

What’s the difference then? Do we need both? Why can’t we just have one? There are many good reasons we need both in large-scale web applications. Here is a very incomplete list:

* Client-side JavaScript cannot access files on the server without server-side script.
* Client-side JavaScript cannot access databases.
* Client-side JavaScript cannot access local file system on the client machine.
* Client-side JavaScript cannot close a window if it didn’t open it.
* Client-side JavaScript cannot access web pages hosted on another domain.

Node.js is a server-side JavaScript technology that allows JavaScript to overcome the above limitations. It makes JavaScript a really powerful language for web application development. In this course we will explore mainly server-side JavaScript features.