**Exercises**

*Lecture 9:*

NodeJS summary

1. Exercise - Node Console

Let's try out the Node console and execute a command. Drop into the node console by typing node. Now try the following:

1. Add 10 + 10.
2. Use console.log('hello world') to output hello world.
3. Find out how many seconds there are in a year. How many seconds there are in a century.

*Further exercise*

* + Make a timeTillChristmas program that tells you how many seconds there are until Christmas.
  + Optionally make it tell you how many days there are.

## 2. Exercise - Create a server

* Modify your server to make it return a string of your choosing.
* Modify your server to make it listen to port 5000.
* Modify your server. Use console.log() to output the entire request object to the console. Have a look around inside, everything is there.
* Now modify your code. Try to write the current URL as a string to the response. If I visit http://localhost:3000/nodejsfresher I should see a web page containing the word nodejsfresher, ideally wrapped in an h1 tag.
* Write a simple 2 page website that responds to [http://localhost:3000/nodetest](http://localhost:3000/nacktschnecke) and <http://localhost:3000/about> and serves content appropriately. Do this using if, else if and else.If no route matches, have your server return a 404 file not found status code and page.

## 3. Exercise – fs

* Create a little program that outputs the current working directory. Extend your Node server so that it can output the value of \_\_dirname.
* Make a file containing some content. Write a program which can read the file from the current directory and output it to to console.
* Farther further exercise – 404 If the file doesn't exist, the error object will be non-null. Detect this case and return a 404.

## 4. Exercise – Module

* Create an ultra-simple date module that can tell you the current date and time. Create some code that requires this module and have it output the time.
* Create a router module. It should receive a URL. It should return the path of a static file which you can then serve out, or else return nil.

Your node app can now var router = require('./router');

You can pass the url to the router and get back the file that should be returned.

* Create a controller module. It should have methods in it which can write to the response object. Now have your router send messages to your controller.

## 5. Exercise - Request-Promise

Use the [request-promise](https://github.com/tyabonil/request-promise) module. Write a node route that pulls content from an open weather map URL:

<http://api.openweathermap.org/data/2.5/weather?q=London,uk>

When you have got the content (the promise is fulfilled), return the data to the user.