# CODE: 5 Minute Node Crash Course

## Section 16, Lecture 273

**//Print "HELLO WORLD" 500 times using Node**

```
for(var i = 0; i < 500; i++){
  console.log("HELLO WORLD!");
}
```

**// Execute file with:**

`node filename.js`

# CODE: Introduction to NPM and Faker

Section 16, Lecture 275

Find Faker Docs Here: https://github.com/marak/Faker.js/

## STEP 1: Install and Require Faker

**// Install Faker via command line:**

```
npm install faker
```

**// Require it inside of a JS file:**

```
var faker = require('faker');
```

## STEP 2: Use Faker!

**// Print a random email**

```
console.log(faker.internet.email());
```

**// Print a random past date**

```
console.log(faker.date.past());
```

**// Print a random city**

```
console.log(faker.address.city());
```

**// We can define a new function**

```
function generateAddress(){
  console.log(faker.address.streetAddress());
  console.log(faker.address.city());
  console.log(faker.address.state());
}
```

**// And then execute that function:**

```
generateAddress();
```

# CODE: Connecting Node to MySQL

Section 16, Lecture 278

Documentation for the MySQL Node Package:

## Step 1: Install the MySQL Node Package

```
npm install mysql
```

## Step 2: Connect to Database

```
var mysql = require('mysql');

var connection = mysql.createConnection({
  host     : 'localhost',
  user     : 'learnwithcolt',  //your username
  database : 'join_us'         //the name of your db
});
```

## Step 3: Run Queries

### Running a super simple SQL query like:

```
SELECT 1 + 1;
```

### Using the MySQL Node Package:

```
connection.query('SELECT 1 + 1 AS solution', function (error, results, fields) {
   if (error) throw error;
   console.log('The solution is: ', results[0].solution);
});
```

### Another sample query, this time selecting 3 things:

```
var q = 'SELECT CURTIME() as time, CURDATE() as date, NOW() as now';
connection.query(q, function (error, results, fields) {
  if (error) throw error;
  console.log(results[0].time);
  console.log(results[0].date);
  console.log(results[0].now);
});
```

The equivalent SQL query:

```
SELECT CURTIME() as time, CURDATE() as date, NOW() as now;
```

# CODE: Creating Our Users Table

Section 16, Lecture 280

## Simple SQL To Create The Users Table

```
CREATE TABLE users (
    email VARCHAR(255) PRIMARY KEY,
    created_at TIMESTAMP DEFAULT NOW()
);
```

# CODE: Selecting Using Node

## Section 16, Lecture 282

### To SELECT all users from database:

```
var q = 'SELECT * FROM users ';
connection.query(q, function (error, results, fields) {
  if (error) throw error;
  console.log(results);
});
```

### To count the number of users in the database:

```
var q = 'SELECT COUNT(*) AS total FROM users ';
connection.query(q, function (error, results, fields) {
  if (error) throw error;
  console.log(results[0].total);
});
```

# CODE: Inserting Using Node

Section 16, Lecture 284

## Inserting Data Using Node

### Approach #1

```
var q = 'INSERT INTO users (email) VALUES ("rusty_the_dog@gmail.com")';

connection.query(q, function (error, results, fields) {
  if (error) throw error;
  console.log(results);
});
```

### An easier approach that allows for dynamic data

```
var person = {
    email: faker.internet.email(),
    created_at: faker.date.past()
};

var end_result = connection.query('INSERT INTO users SET ?', person, function(err, result) {
  if (err) throw err;
  console.log(result);
});
```

# CODE: Bulk Inserting 500 Users

Section 16, Lecture 287

## The Code To INSERT 500 Random Users

```
var mysql = require('mysql');
var faker = require('faker');


var connection = mysql.createConnection({
  host     : 'localhost',
  user     : 'learnwithcolt',
  database : 'join_us'
});


var data = [];
for(var i = 0; i < 500; i++){
    data.push([
        faker.internet.email(),
        faker.date.past()
    ]);
}


var q = 'INSERT INTO users (email, created_at) VALUES ?';

connection.query(q, [data], function(err, result) {
  console.log(err);
  console.log(result);
});

connection.end();
```

# CODE: 500 Users Exercises Solutions

Section 16, Lecture 290

## Solutions To 500 Users Exercises

### -- Challenge 1

```
SELECT
    DATE_FORMAT(MIN(created_at), "%M %D %Y") as earliest_date
FROM users;
```

### -- Challenge 2

```
SELECT *
FROM   users
WHERE  created_at = (SELECT Min(created_at)
                     FROM   users);
```

### -- Challenge 3

```
SELECT Monthname(created_at) AS month,
       Count(*)              AS count
FROM   users
GROUP  BY month
ORDER  BY count DESC;
```

### -- Challenge 4

```
SELECT Count(*) AS yahoo_users
FROM   users
WHERE  email LIKE '%@yahoo.com';
```

### -- Challenge 5

```
SELECT CASE
         WHEN email LIKE '%@gmail.com' THEN 'gmail'
         WHEN email LIKE '%@yahoo.com' THEN 'yahoo'
         WHEN email LIKE '%@hotmail.com' THEN 'hotmail'
         ELSE 'other'
       end      AS provider,
       Count(*) AS total_users
FROM   users
GROUP  BY provider
ORDER  BY total_users DESC;
```