# DBI Lab 001: Setup and Networking Commands

COMP1048: Databases and Interfaces

Matthew Pike & Yuan Yao

- Lab Overview
  - Preparing for the Module
    - Required Software
      - Text/Code Editor
      - Web-Browser
    - Python
    - SQLite Installation
      - Mac Users
      - Linux Users
        - Debian / Ubuntu
        - CentOS / Fedora / RedHat
      - Windows Users
  - Networking Commands
    - ping
      - Exercise 1
    - nslookup
      - Exercise 2
    - Traceroute
      - Exercise 3
    - Whois
      - Exercise 4
  - Submission

# Lab Overview

There are two objective associated with today's lab session:

1. Ensure our local machine (your laptop) has the neccesary software installed for this module
2. Become familiar with the command line environment and execute basic networking commands

## Preparing for the Module

> *Aim* - Ensure that your local machine (your laptop) has the neccesary software installed for the module

As you may be aware from your experiences in other modules, in order to do certain software development activities, we require specific software tools to help us. This is similar for DBI, we need some specialised tools to aid us in developing some of the code we are going to produce in this module. In this activity we are going to ensure that our personal computing device is prepared.

> **Note** - Going forward, these instructions will assume that you are working in a UNIX based operating system. If you are using Window, you may want to consider installing the Windows Subsystem for Linux (WSL) - https://docs.microsoft.com/en-us/windows/wsl/install-win10 . Note that this is not required, but may simplify some activities going forward. We cannot provide support for this solution, as we do not use it ourselves!

## Required Software

Below is a collection of software everyone needs to have installed.

> **Top Tip**
>
> If you are on a **Mac** you can use **Homebrew** . Homebrew is a package manager, similar to those that are provided with mainstream Linux distributions (e.g. `apt` on Debian systems).

### Text/Code Editor

You cannot use Microsoft Word to edit code. You need to install a suitable text/code editor for developing web applications. There are quite literally 100's of editors available. Below are a selection of a few popular ones. If you are not sure which to install, then install the first one in the list:

- Sublime Text
- Visual Studio Code

### Web-Browser

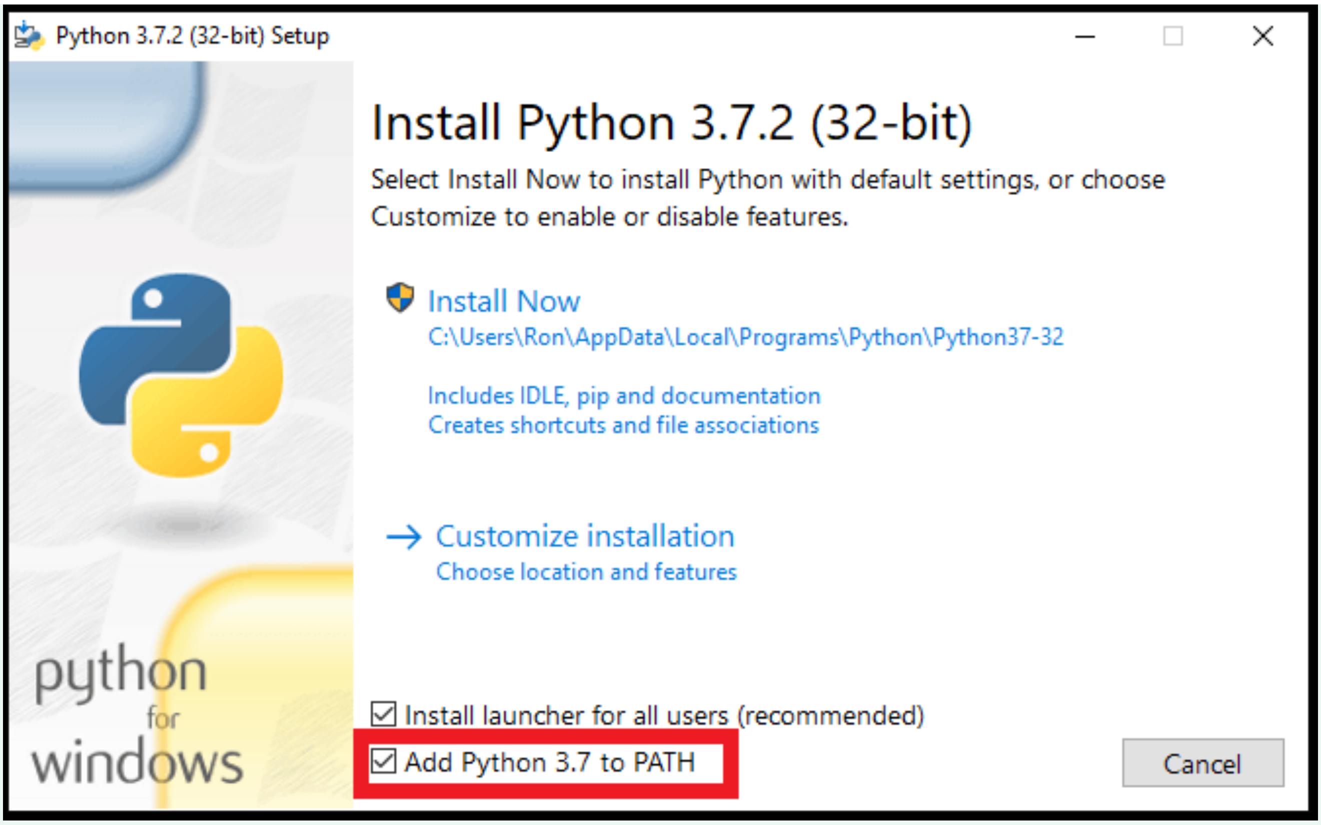
We are going to utilise some of the advanced functionality that comes pre-built into modern web-browsers, namely the developer tools. These will be helpful when debugging our web applications. We particuarly recommend that you have one (or both) of the following web-browsers installed:

- Google Chrome
  - A chrome based alternative, such as Opera , is also fine.
- Mozilla Firefox

## Python

We will use Python (version 3) for parts of this module. Download the latest version for your platform from [ https://www.python.org/downloads/ ]. Note, installation instructions differ for each platform. Please follow the instructions provided on the Python.

> **Important - Windows Users**
>
> When installing, please ensure that you select "Add python to the PATH".



Python 3.7.2 (32-bit) Setup

Install Python 3.7.2 (32-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:\Users\Ron\AppData\Local\Programs\Python\Python37-32
Includes IDLE, pip and documentation
Creates shortcuts and file associations
→ Customize installation
Choose location and features
python for windows
Cancel

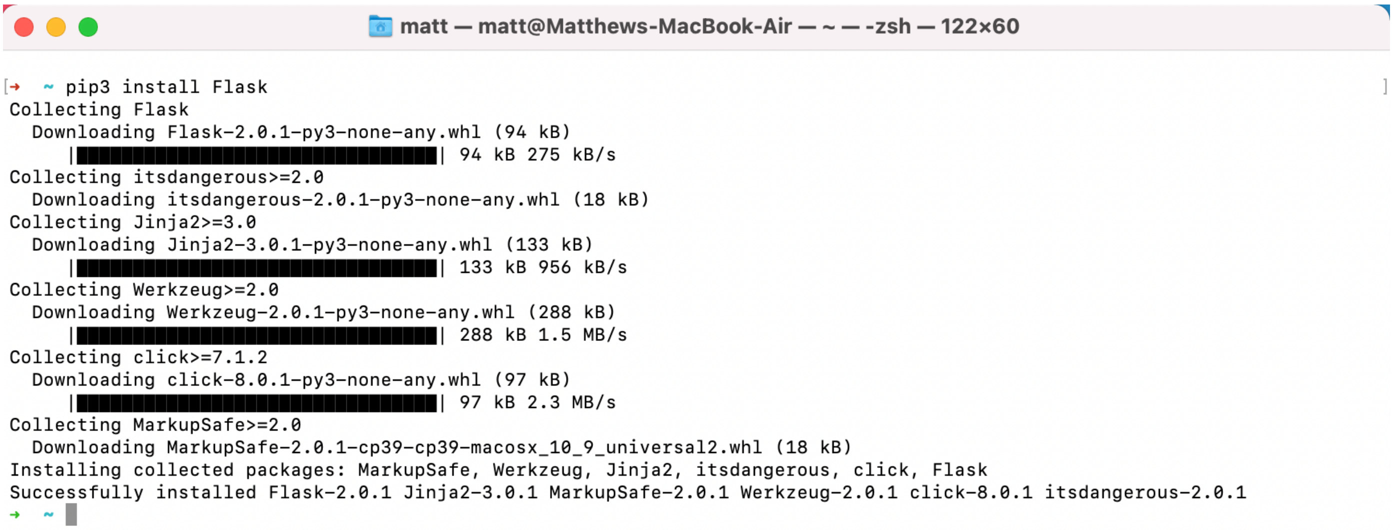
☑ Install launcher for all users (recommended)

☑ Add Python 3.7 to PATH

Once you have completed this step, you now need to install the Flask module. We'll discuss what Flask is later on the module. For now, we'll install Flask using the Command Line / Terminal, via the following command:

```
pip3 install Flask
```

If executed succesfully, your terminal window should look like this:



```
[➜  ~ pip3 install Flask                                                                                                  ]
Collecting Flask
  Downloading Flask-2.0.1-py3-none-any.whl (94 kB)
     |████████████████████████████████| 94 kB 275 kB/s
Collecting itsdangerous>=2.0
  Downloading itsdangerous-2.0.1-py3-none-any.whl (18 kB)
Collecting Jinja2>=3.0
  Downloading Jinja2-3.0.1-py3-none-any.whl (133 kB)
     |████████████████████████████████| 133 kB 956 kB/s
Collecting Werkzeug>=2.0
  Downloading Werkzeug-2.0.1-py3-none-any.whl (288 kB)
     |████████████████████████████████| 288 kB 1.5 MB/s
Collecting click>=7.1.2
  Downloading click-8.0.1-py3-none-any.whl (97 kB)
     |████████████████████████████████| 97 kB 2.3 MB/s
Collecting MarkupSafe>=2.0
  Downloading MarkupSafe-2.0.1-cp39-cp39-macosx_10_9_universal2.whl (18 kB)
Installing collected packages: MarkupSafe, Werkzeug, Jinja2, itsdangerous, click, Flask
Successfully installed Flask-2.0.1 Jinja2-3.0.1 MarkupSafe-2.0.1 Werkzeug-2.0.1 click-8.0.1 itsdangerous-2.0.1
➜  ~ █
```

## SQLite Installation

In the coming weeks we will be using SQL to interact with a modern DBMS. Previously we have used MariaDB, which is one example of a modern and fully featured DBMS. MariaDB however is quite a heavyweight solution and installing it on your personal laptops has been problematic in the past.

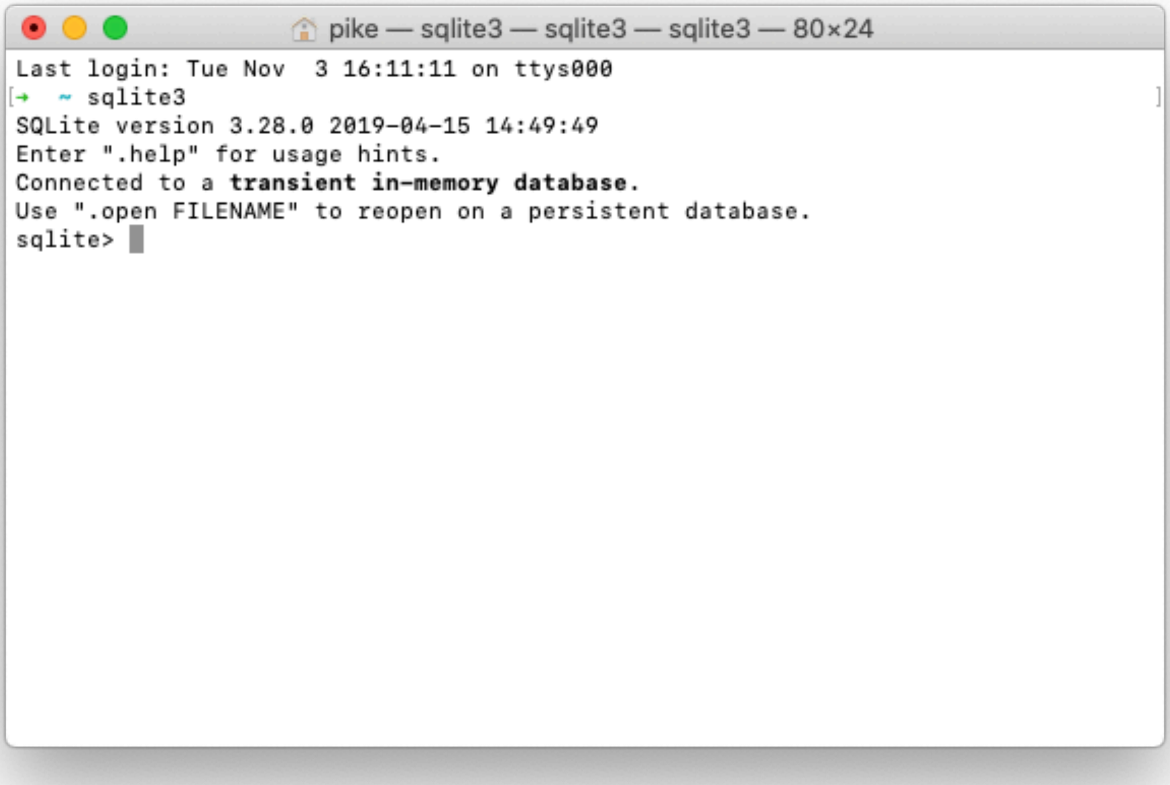
This year we're going to try something new - SQLite. SQLite is summarised very nicely on their website:

> SQLite is a C-language library that implements a small, fast, self-contained, high-reliability, full-featured, SQL database engine. SQLite is the most used database engine in the world. SQLite is built into all mobile phones and most computers and comes bundled inside countless other applications that people use every day.

Today, we simply want to ensure that you are able to run SQLite on your machine. We will be using SQLite extensively in future lab sessions and the remaining coursework. It's important therefore that we ensure that any installation issues are resolved today. Thankfully, SQLite makes this extremely straightforward todo.

### Mac Users

SQLite *should* be preinstalled on your Mac! To test this, open a Terminal Window (`cmd + space` then search for "Terminal") and type `sqlite3`, you should be greeted by the following:



```
Last login: Tue Nov  3 16:11:11 on ttys000
→  ~ sqlite3
SQLite version 3.28.0 2019-04-15 14:49:49
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite>
```

### Linux Users

It's likely that most distributions come with SQLite preinstalled. If not, you will need to use your OS's package manager to install this dependency:

**Debian / Ubuntu**

```
apt-get install sqlite3
```

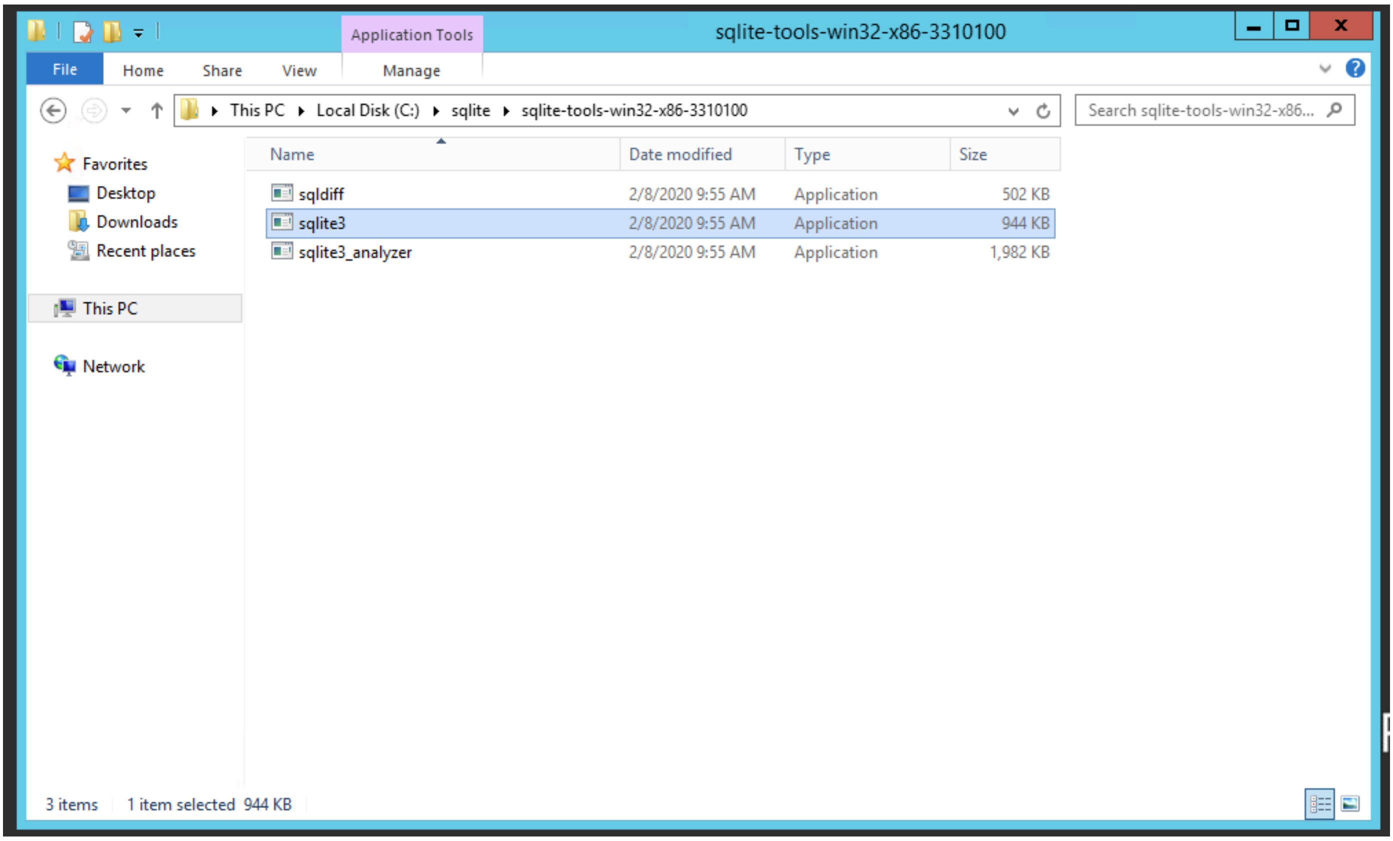
**CentOS / Fedora / RedHat**

```
yum install sqlite3
```

### Windows Users

Typically, things are not as straightforward as we'd hope on Windows. To install:

1. Navigate to https://www.sqlite.org/download.html
2. Find the heading "**Precompiled Binaries for Windows**" and click the "sqlite-tools-win32-x86-XXXXXXX" (where XXXXXX is a version number). This will include a bundle of command-line tools for managing SQLite database files, including the command-line shell program, the sqldiff.exe program, and the sqlite3_analyzer.exe program.
3. Unzip the downloaded archive
4. Double click the `sqlite3.exe` and your view should be similar to that shown below



sqlite-tools-win32-x86-3310100

This PC ▸ Local Disk (C:) ▸ sqlite ▸ sqlite-tools-win32-x86-3310100

| Name | Date modified | Type | Size |
|---|---|---|---|
| sqldiff | 2/8/2020 9:55 AM | Application | 502 KB |
| sqlite3 | 2/8/2020 9:55 AM | Application | 944 KB |
| sqlite3_analyzer | 2/8/2020 9:55 AM | Application | 1,982 KB |

3 items 1 item selected 944 KB

## Networking Commands

> *Aim* - To interact with a command line interface and execute basic networking commands

The below commands may be helpful when you are building or debugging you web applications.

## ping

The command `ping <host>` sends a series of packets and expects to receive a response to each packet. When a return packet is received, ping reports the round trip time (the time between sending the packet and receiving the response).

You may find `ping` useful when you are debugging a web application. If, for example, your web application is loading very slowly, you can use the `ping` command to investigate, at the network level, what the latency between your machine and the server you are interacting with.

Example

```
➜ ping nottingham.edu.cn
PING nottingham.edu.cn (10.179.255.237): 56 data bytes
64 bytes from 10.179.255.237: icmp_seq=0 ttl=126 time=1.634 ms
64 bytes from 10.179.255.237: icmp_seq=1 ttl=126 time=2.891 ms
64 bytes from 10.179.255.237: icmp_seq=2 ttl=126 time=4.134 ms
^C
--- nottingham.edu.cn ping statistics ---
3 packets transmitted, 3 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 1.634/2.886/4.134/1.021 ms
➜  ping baidu.cn
PING baidu.cn (39.156.69.79): 56 data bytes
64 bytes from 39.156.69.79: icmp_seq=0 ttl=46 time=41.478 ms
64 bytes from 39.156.69.79: icmp_seq=1 ttl=46 time=41.143 ms
64 bytes from 39.156.69.79: icmp_seq=2 ttl=46 time=43.324 ms
64 bytes from 39.156.69.79: icmp_seq=3 ttl=46 time=40.790 ms
^C
--- baidu.cn ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 40.790/41.684/43.324/0.978 ms
➜  ping nottingham.ac.uk
PING nottingham.ac.uk (185.18.139.133): 56 data bytes
64 bytes from 185.18.139.133: icmp_seq=0 ttl=240 time=246.083 ms
64 bytes from 185.18.139.133: icmp_seq=1 ttl=240 time=269.162 ms
64 bytes from 185.18.139.133: icmp_seq=2 ttl=240 time=290.620 ms
^C
--- nottingham.ac.uk ping statistics ---
3 packets transmitted, 3 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 246.083/268.622/290.620/18.186 ms
```

Notice, in the above example, reaching `nottingham.edu.cn` happens very quickly when I execute this command on-campus. However, when testing the same command with `nottingham.ac.uk` the round-trip time is 100-150 times slower! Have a think about why these differences exist.

### Exercise 1

Using the `ping` command, test three different websites that you often visit. Note their round-trip time and reason why this may be. Which of the three websites you tested do you think is hosted (geographically) closest to you? Why?

## nslookup

The command `nslookup <host>` will perform a DNS query to identify the IP address for a domain name (it can also do the reverse). `nslookup` performs a "nameserver lookup" through using a DNS server.

```
➜  nslookup nottingham.edu.cn
Server:    10.2.1.6
Address:   10.2.1.6#53

Name: nottingham.edu.cn
Address: 10.2.1.6
Name: nottingham.edu.cn
Address: 10.2.1.3
Name: nottingham.edu.cn
Address: 10.2.1.5
Name: nottingham.edu.cn
Address: 10.179.255.237
Name: nottingham.edu.cn
Address: 10.159.237.232
Name: nottingham.edu.cn
Address: 10.159.237.233

➜  nslookup nottingham.ac.uk
Server:    10.2.1.6
Address:   10.2.1.6#53

Non-authoritative answer:
Name: nottingham.ac.uk
Address: 185.18.139.133
```

`nslookup` may help you when you are deploying larger web applications, especially if you were working on an international project, for example. It may sometimes be necessary to find out exactly which server you are being directed to.

### Exercise 2

Using the same three web addresses you used in Exercise 1, identify the IP address of the server you are being directed too, via DNS.

## Traceroute

> **Windows users** - Use the command `tracert <host>` instead.

The command `traceroute <host>` will show the routers that a packet encounters on their way to the `<host>`.

Traceroute can be a very powerful and useful tool for identifying bottlenecks in your systems' network.

### Exercise 3

Use the `traceroute` command to investigate the connection latency between your machine and `nottingham.ac.uk`.

## Whois

> **Windows** - This command will not work on Windows.

The *whois* command can give detailed information about domain names and IP addresses. The command may also detail who owns or is responsible for the domain. It often includes a list of domain name servers for the organisation.

### Exercise 4

Using the same three web addresses you used in Exercise 1, identify the `whois` information related to these web domains.

# Submission

Please submit a **PDF document** containing the output for the networking commands (`ping`, `nslookup`, `traceroute` and `whois`). There is no prescribed format, but please ensure that your work is clearly presented, using section headings and supportive prose, as neccesary. All submissions should include your name, student id, email and a title for your work. This is the minimum expectation for submitting university standard of work. You will receive no marks for completing this assignment, but this will be good practice for future tasks, which are graded. The submission point is available on Moodle, under the "Week 1" topic heading.

**Submission Deadline** - Monday, 3 October 2022 at 15:00.