



# Introduction to Programming

## Tutorial Task 1.1: Hello World

### Overview

As a first step, create the classic “Hello World” program. This will help ensure that you have all of the software installed correctly, and are ready to move on with creating other programs.

- Purpose:** Install and test the tools needed to get started programming.
- Task:** Create your own Hello World program using the command line interpreter.
- Time:** This task should be completed in your first lab class.
- Resources:** [Pine, C 2013 \*Learn to Program, The Pragmatic Programmer\*](#). (available in the library).
- Learning Material for Week 1.**

### Submission Details

You must submit the following files to Canvas:

- Hello World source code (hello\_world.rb)
- Screenshot of the Terminal showing use of **cd**, **ls**, and **ruby** commands as well as execution of your Hello World program.

Make sure that your task has the following in your submission:

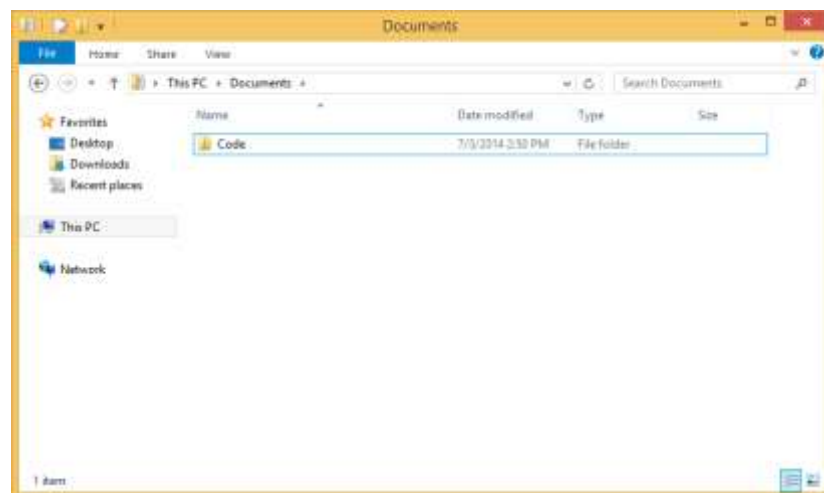
- Code layout - match the example for indentation and use of case (UPPERCASE, lowercase, etc) for the different aspects of the code.
- The code must run and the screenshot show it working on your computer.



# Instructions

The first task includes the steps needed for you to install the tools you will need in this unit. You will then use these tools to create the classic '*Hello World*' program.

1. If using a laptop Install the tools you need to get started for your operating system.  
Install **Atom**, **SublimeText** or **Notepad++** or **equivalent** (or use a text editor of your choice) see the "install Ruby" notes on Blackboard/Canvas
2. If you don't already have one, make a directory (i.e., a 'folder') to store your code (e.g., *Documents/Code/Lab1*). On a Swinburne computer you may wish to use a directory on your student drive or a USB storage device.
  - Navigate to your *Documents* directory in Finder or File Explorer
  - Right click in the *Documents* directory and select **New Folder**, name it **Code**



3. Open your text editor, and create a new file.
4. Enter the text for the *Hello World* program. It should appear as shown here:



5. Save the file as **hello\_world.rb** in your code directory.
6. Open a **Terminal** (or a CMD shell in Windows), then perform the following commands:
  - Change into the directory containing your code using the **cd** command.  
`cd c:/Users/your_user/Documents/Code` on Windows<sup>1</sup> or  
`cd ~/Documents/Code` on MacOS or Linux
  - List the files in this directory using the **ls** command
  - Print the working directory using the **pwd** command
  - Run your program using **ruby hello\_world.rb**

```
MacBook-Pro:Tute-Tasks mmitchell$ cd 1.1TT/
MacBook-Pro:1.1TT mmitchell$ ls
Resources      TT1_1.docx      TT1_1.pdf
MacBook-Pro:1.1TT mmitchell$ cd Resources/
MacBook-Pro:Resources mmitchell$ ls
hello_world.rb
MacBook-Pro:Resources mmitchell$ ruby hello_world.rb
Hello World
MacBook-Pro:Resources mmitchell$
```

**Tip:** Bash commands (e.g., `cd`, `ls`, `pwd`, `fc`) do not like spaces in directory or file names (e.g., `My Documents`, or `hello_world.rb`). If you have a space in the name of something you need to add in a reverse slash:

`My\ Documents` and `hello_world.rb`

**Avoid spaces** in the names of your files and folders!

<sup>1</sup> Replace `your_user` with your computer user name

TIP: you can also run Ruby commands in the terminal window using Interactive Ruby (irb). See the following screenshot:

```
MacBook-Pro:Resources mmitchell$ irb
2.4.1 :001 > puts 'Hello World'
Hello World
=> nil
2.4.1 :002 > █
```

Now that the Task is complete you can submit it for assessment, which will help prepare it for your portfolio.

7. Use [Skitch](#) (or your preferred screenshot program) to take a screenshot of the Terminal, as this is one of the things you will need to submit.
8. Login to Canvas, and locate Tutorial Task 1.1
9. Change the status of the task to **Ready To Mark**
10. Upload your completed Hello World code and the screenshot.
11. If you check back later Canvas will have prepared these as PDFs for your tutor to assess.
12. Now, remember to save the document and **backup** your work to multiple locations!
  - Once you get things working you **do not** want to lose them.
  - Work on your computer's storage device most of the time... but backup your work when you finish each task.
  - Use **Dropbox** or a similar online storage provider, as well as other locations.
  - Canvas is not a Backup of your work, so make sure you keep a copy!
  - USB keys and portable hard drives are good secondary backups... but can be lost/damaged (do not rely upon them).

You now have your first portfolio piece. This will help demonstrate your learning from the unit.

**Note:** This is one of the tasks you need to **submit to Canvas**. Tutors should give guidance and perhaps feedback in the tutorial class.

Check the assessment criteria for the important aspect your tutor MAY check when assessing your finished portfolio.