THE UNIVERSITY OF SYDNEY

INFO1110 / COMP9001 Week 1 Tutorial

Introduction, Terminal And Output

Introduction

Introduce yourself to the rest of the class, what degree you are in, why you are interested in programming and what you did over the break. Make sure you know your tutor's name by the end of the tutorial.

Edstem

We use EdStem for our forum, challenges and assignments. please get familiar with Edstem as it will be used heavily through out the semester and is typically the place where announcements are made. Please make sure you can login and reply to Masa's welcome post. If you cannot login, please notify your tutor so they can address this issue.

Linux

It is recommended that you get familiar with linux and have access to a unix or unix-like operating system (Linux or macOS). For this tutorial and through out the semester we will like if you could reboot into linux on the lab machines.

To reboot into Red Hat Linux or Fedora Linux on the lab machines, please reboot the computer and you will have be presented with a boot screen asking you to choose between Windows and Linux, Please select Linux, login and please move onto the next section.

Terminal and filesystem

Before we learn any python code we will get familiar with the terminal and unix command line environment.

cd	Change directory
mkdir	Make directories
ls	List directory contents
rmdir	Remove directory
rm	Remove
mv	Move (also used to rename files)
cp	Copy
pwd	Print Working Directory

Directory symbols

..	Parent Directory	cd ..
.	Current Directory	./program
-	Previous directory	cd -
~	Home directory	cd
/	Root directory	cd /

Navigating the filsystem

After first launching terminal you will be presented with prompt, waiting for your commands. Your cursor will typically begin after the $ symbol

Example:

```
user@sahara:~$
```

When you first launch terminal, you will typically be in the home directory. To get the contents of the current directory you are in, you can use the command

```
ls
```

Which will list folders and files, to show the contents on a line-by-line basis, you can use the -l flag with the ls command

```
ls -l
```

Time to create a folder, you can create a folder using the mkdir command and passing the name of the folder as an argument, like so

```
mkdir INFO1110
```

We have now created the folder INFO1110, you can see this by running ls to see if it exists

To move INFO1110 folder, use the cd command and use name of the folder, in this case INFO1110

```
cd INFO1110
```

On the left hand side of the terminal, we can see what directory we are currently in, but can also get this information with the command

```
pwd
/home
```

SSH

You will be able to access the undergraduate servers from outside of uni using SSH (Secure Shell). To access ucpu0 from the lab machines you can run the following command:

```
ssh <unikey>@ucpu0.ug.it.usyd.edu.au
```

and access the drive.

Outside of the university will require accessing the University of Sydney's VPN a resource for access the VPN from your home computer

Most of the lab machines will have Atom and Vim installed on them. We recommend you use **Atom** for writing your source code in.

Text Editors

- **Atom:** Developed by github, also has an internal package manager and also can be configured into a full IDE for most languages. FOSS.
- **Vi/Vim, Nano and Emacs:** Text editors with no mouse support, it is incredibly useful to know how to use at least one of these as ssh and telnet protocols for accessing a remote computer are text only (without the appropriate extensions), mainly due to having nearly predating the mouse. The arguments about the merits of Vi and Emacs pre-date the internet and have their own wikipedia page that may be considered suggested reading. In short emacs is considered more featured (having its own web browser, e-mail client and games and is often joked as being an operating system), while vim is more lightweight and is installed on just about everything by default.
- **Vscode:** Developed by Microsoft, has its own internal package manager, can be configured into a full IDE for most languages. FOSS despite being developed by Microsoft.

Python

Time to write some python code! To access the python interpreter, please make sure you still have a terminal window open. You can access the interpreter by using the **python** command.

Python Interpreter

First, we will start by introducing the python interpreter. This program takes the python source code, compiles (on the fly) and then executes it.

```
python
```

However, we will want to use python version 3, to check if the python command is using 2 or 3, run the python command with the –version flag.

```
python --version
```

If the output from the terminal is similar to

```
Python 2.7.14
```

Then the default python interpreter is version 2. In the event of this situation, the system may still have python 3 present but there will be a separate command called **python3**

```
python3
```

Writing python source file

Open your favourite text editor (atom, vim, emacs), create a file and save it as hello.py

After saving the file, we can started with writing a basic hello world program and introduce you to the print function.

```
print("Hello Python!!")
```

To run this program, we execute the python command with the source file name.

```
python hello.py
```

Your program should output

```
Hello Python!
```

Exercise - Reading input

Write a program that will allow the user to read in their first name and last name separately and then output the combined name.

Example:

```
name = input("What is your name? ")
print("My name is: " + name)
```

Example:

```
What is your first name? David
What is your last name? Smith
Your name is David Smith
```

Exercise - Reading input

Create a program called **box.py** that will print out character to draw a box on screen.

```
python box.py
+-------+
|       |
|       |
|       |
+-------+
```

Exercise - Errors in code

Given this error, what do you believe is the issue with the code.

```
SyntaxError: Missing parentheses in call to 'print'.
Did you mean print("Hello World!")?
>>> print(Hello World!)
  File "<stdin>", line 1
    print(Hello World!)
                      ^
SyntaxError: invalid syntax
```

Exercise - Sum of numbers

Part 1 Copy the following program shown below.

```
print("Hello")
print("I will add two numbers for you")
print(:Enter two whole numbers on one line")
n1 = 0
n2 = 0
line = input()
n1, n2 = line.split(" ")
n1 = int(n1)
n2 = int(n2)
print("The sum of your numbers is:")
print(n1+n2)
```

Part 2 Modify your program so it pritns out the sum of three numbers. Run it as Before

Part 3 Write a complete program that works out in which year a person's nth brithday will occur, given the year of their birth.

Comments

Once your program reaches a reasonable amount of complexity you should provide some kind of documentation so that you yourself can understand what you have written.

Code in python is typically commented using # or using multiline strings without assigning it to a variable.

is typically used for inline and non-publishable documentation while multiline strings are used for large code bases which will require developers to read about.

Example of # comments in python:

```
#Iterate 10 times and print the value of x on each iteration
for x in range(10)
  print(x)
```

Help and API Documentation

You will run into problems that will make you consult the documentation for the programming language. You will can access this resource by following this link: Python 3.6 Documentation

Or you can use the python interpreter's help function. This function will allow you to inquire about certain types and packages within the python ecosystem that you have installed.

For example, if we wanted to get help with Type, we would write

```
help(int)
```

Or function

```
help(help)
```

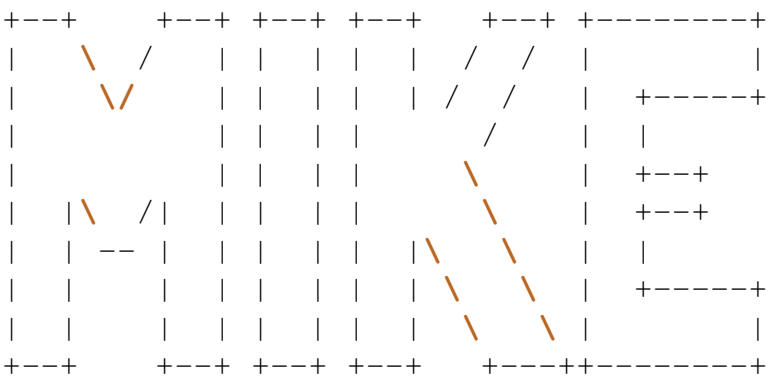
or module

```
help(sys)
```

Making Boxes - Extension

Change your box program to print out a whole load of box, triangles and circles. Once the circle has been created you could find the value of pi (This is not an easy task, unless you are feeling adventurous you could skip this).

Try and construct your name using shapes or draw a picture with text.



If you want to output a backslash or a new line within a string, you will need to use backslash to denote that it is an *escape character*.

Hello Command Line Arguments - Extension

Try and read in command line arguments into your python program and print them out.

Hint: You will need to import sys to your python file and use sys.argv, you can access individual items from command line.

Pattern:

```
sys.argv[index]
```

Example:

```
sys.argv[0]
```

Builtins - Extension

You can checkout other built in functions available to you by using dir(),

Given the following code:

```
dir('__builtins__')
```

And executed, **dir**ectory function in python, you can use this as a way of quickly inspecting what functions an object or module contains.