

The basics of Bash -- [Pedro Pessoa \(https://pessoap.github.io/\)](https://pessoap.github.io/)

If you are new to using terminals, Bash scripting might seem intimidating at first. However, it is a powerful and versatile tool that can greatly simplify many tasks. Often, running software on supercomputers requires using the terminal exclusively, making Bash an essential skill. Bash, which stands for "Bourne Again SHell," allows you to automate and streamline complex workflows by writing scripts that execute a series of commands.

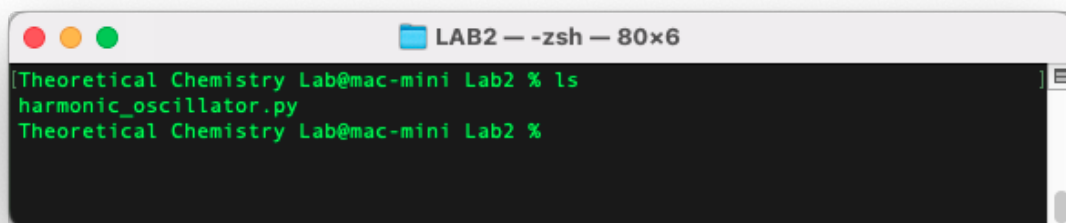
The purpose of this text is *not* to teach you basic Bash commands, [there are many good resources for this online \(https://devhints.io/bash\)](https://devhints.io/bash). Instead, we aim to demonstrate how these commands can be used to accomplish specific tasks. We'll conclude with a simple yet interesting application where we use Bash to generate graphs for harmonic oscillators with different frequencies.

Using Bash on terminal

Open the terminal, in the same directory as the other files for this lab.

Command echo

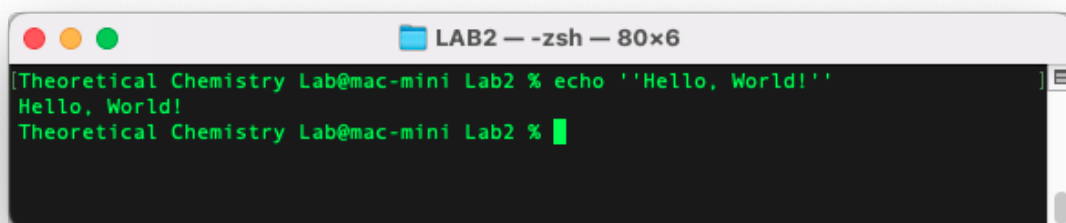
Although it may not look like it sometimes, Bash is a programming language and, like many programming languages, the first example will be the classic "Hello, world!". Yes, everyone gives this example, but it's a tradition at this point! Type `echo 'Hello, world!'` on the terminal and observe the result.



```
LAB2 — -zsh — 80x6
Theoretical Chemistry Lab@mac-mini Lab2 % ls
harmonic_oscillator.py
Theoretical Chemistry Lab@mac-mini Lab2 %
```

Command ls

Now, for something a little more useful, let us list all the files within the directory of interest. This is done with the command `ls`



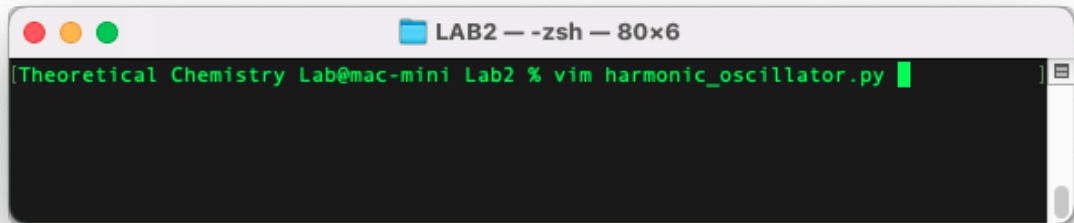
```
LAB2 — -zsh — 80x6
Theoretical Chemistry Lab@mac-mini Lab2 % echo 'Hello, World!'
Hello, World!
Theoretical Chemistry Lab@mac-mini Lab2 %
```

where we can see the single file in the folder - `harmonic_oscillator.py`

Using software vim

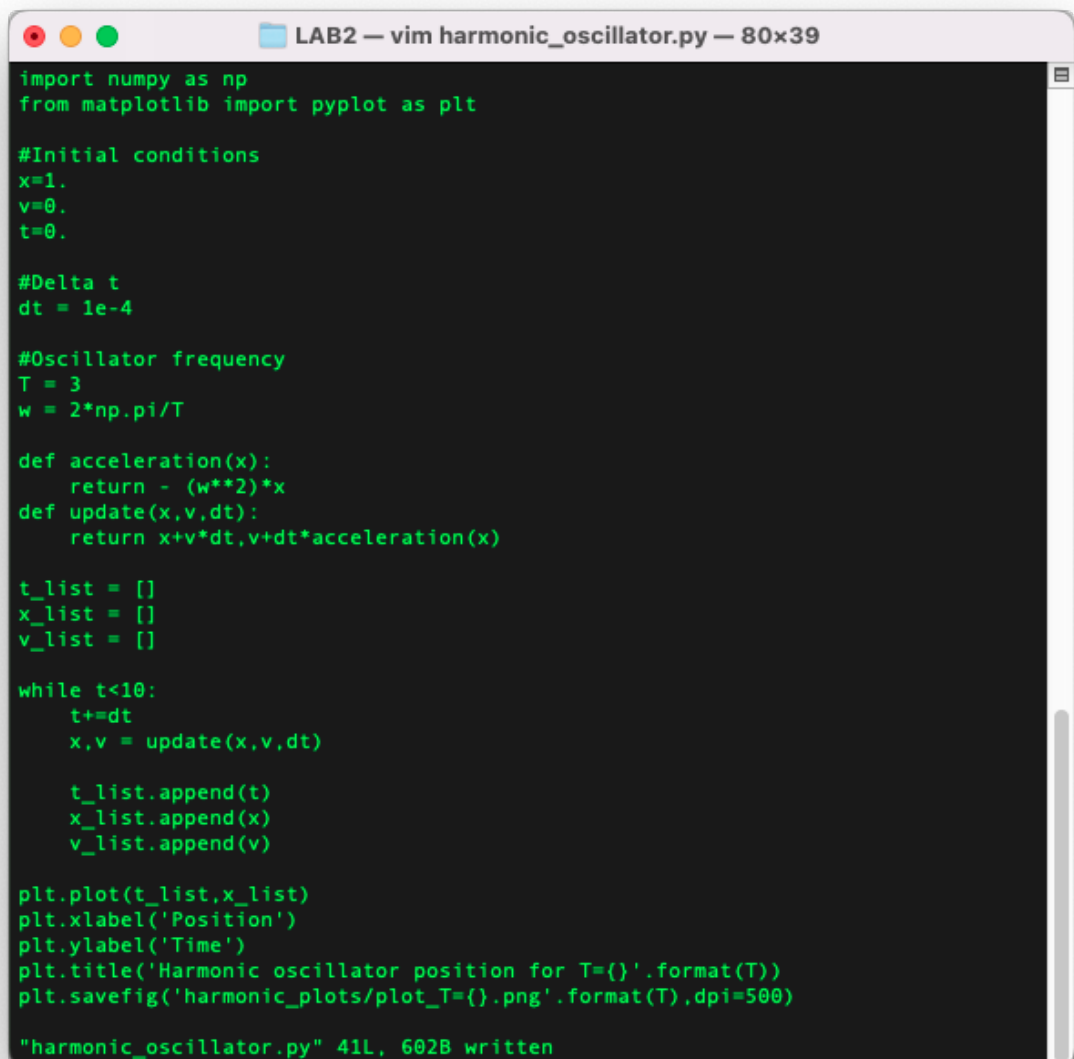
Running software from the terminal is an important task. However, software that relies on a graphical interface often cannot run directly in the terminal. Thus, when it comes to reading or changing a file, only a handful of text editors can be used.

One popular choice among programmers is the text editor called `vim`. Let us use `vim` to open our file `harmonic_oscillator.py` typing the following command:



```
LAB2 — -zsh — 80x6
Theoretical Chemistry Lab@mac-mini Lab2 % vim harmonic_oscillator.py
```

Your terminal should now look something like this:



```
LAB2 — vim harmonic_oscillator.py — 80x39
import numpy as np
from matplotlib import pyplot as plt

#Initial conditions
x=1.
v=0.
t=0.

#Delta t
dt = 1e-4

#Oscillator frequency
T = 3
w = 2*np.pi/T

def acceleration(x):
    return - (w**2)*x
def update(x,v,dt):
    return x+v*dt,v+dt*acceleration(x)

t_list = []
x_list = []
v_list = []

while t<10:
    t+=dt
    x,v = update(x,v,dt)

    t_list.append(t)
    x_list.append(x)
    v_list.append(v)

plt.plot(t_list,x_list)
plt.xlabel('Position')
plt.ylabel('Time')
plt.title('Harmonic oscillator position for T={}'.format(T))
plt.savefig('harmonic_plots/plot_T={}.png'.format(T),dpi=500)

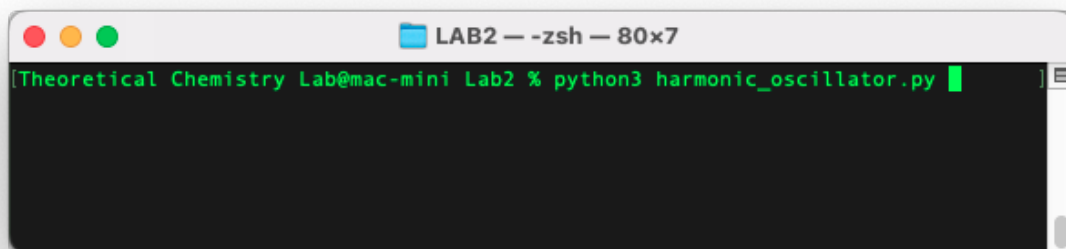
"harmonic_oscillator.py" 41L, 602B written
```

If you try to make any changes to this Python code, you will notice how `vim` may not look that user-friendly at first. Later in

this lab we will eventually to learn how to use it. effective. For now simply type `:q` to close `vim`.

Running python code

Let us move to a more interesting part of this lab: running a Python program from the terminal. As some of you may have noticed, the file `harmonic_oscillator.py` is a python script that calculated the position of a harmonic oscillator as a function of time, as you were taught earlier today. More specifically, it generates a graph showing the position of the harmonic oscillator over time. We will see it in action later. For now, let's run the script using the following command:



```
LAB2 — -zsh — 80x7
Theoretical Chemistry Lab@mac-mini Lab2 % python3 harmonic_oscillator.py
```

However, the way it is setup here you will probably see an error message. Specific instructions on how to fix it will be given in class.

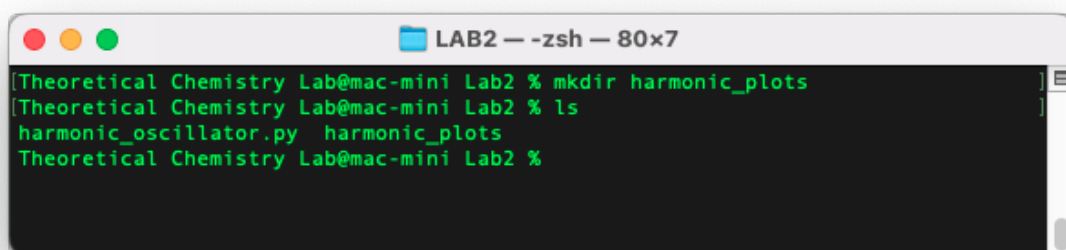
Although I do not want to sound negative, confusing error messages like this are (unfortunately) an intrinsic part of programming.

Here we are going to give, on the board, some instructions on how to setup libraries for python.

Once this is done, what is not working is that the script is trying to save a figure in the directory `harmonic_plots` but such directory does not exist, which is what we are going to fix with our next command.

Command `mkdir`

To create a directory named `harmonic_plots`, use the `mkdir` (make directory) command as shown below:

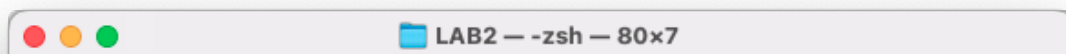


```
LAB2 — -zsh — 80x7
Theoretical Chemistry Lab@mac-mini Lab2 % mkdir harmonic_plots
Theoretical Chemistry Lab@mac-mini Lab2 % ls
harmonic_oscillator.py  harmonic_plots
Theoretical Chemistry Lab@mac-mini Lab2 %
```

Note that we also verified that the directory was indeed created using the `ls` command.

Running python code (again)

With the directory ready, let us run the Python script again



```
LAB2 — -zsh — 80x7
```

```
[Theoretical Chemistry Lab@mac-mini Lab2 % python3 harmonic_oscillator.py
```

This time, it should execute correctly. However, as previously mentioned, it saved the figure in another directory. During this summer school, most of the code you run will do something similar. The next command will help you see the resulting figure.

Checking for output --- Command `cd`

Entering the directory `harmonic_plots` where the figure was saved is straightforward:

```
harmonic_plots — -zsh — 80x8
[Theoretical Chemistry Lab@mac-mini Lab2 % python3 harmonic_oscillator.py
[Theoretical Chemistry Lab@mac-mini Lab2 % cd harmonic_plots
[Theoretical Chemistry Lab@mac-mini harmonic_plots %
```

Here it is also interesting to observe the files within that directory using the command `ls`

```
harmonic_plots — -zsh — 80x7
[Theoretical Chemistry Lab@mac-mini Lab2 % python3 harmonic_oscillator.py
[Theoretical Chemistry Lab@mac-mini Lab2 % cd harmonic_plots
[Theoretical Chemistry Lab@mac-mini harmonic_plots % ls
plot_T=3.png
[Theoretical Chemistry Lab@mac-mini harmonic_plots %
```

Now to, finally, observe the results by opening the file:

```
harmonic_plots — -zsh — 80x7
[Theoretical Chemistry Lab@mac-mini Lab2 % python3 harmonic_oscillator.py
[Theoretical Chemistry Lab@mac-mini Lab2 % cd harmonic_plots
[Theoretical Chemistry Lab@mac-mini harmonic_plots % ls
plot_T=3.png
[Theoretical Chemistry Lab@mac-mini harmonic_plots % open plot_T=3.png
```

- Use the key `esc` followed by `:wq` to exit `vim` with the file saved.
- Type `chmod +x myscript.sh` in the terminal. This gives your script permission to run.
- Type `./myscript.sh` and press Enter. You should see "Hello, World!" printed on the screen

Lab Assignment

1 - Write your second bash script

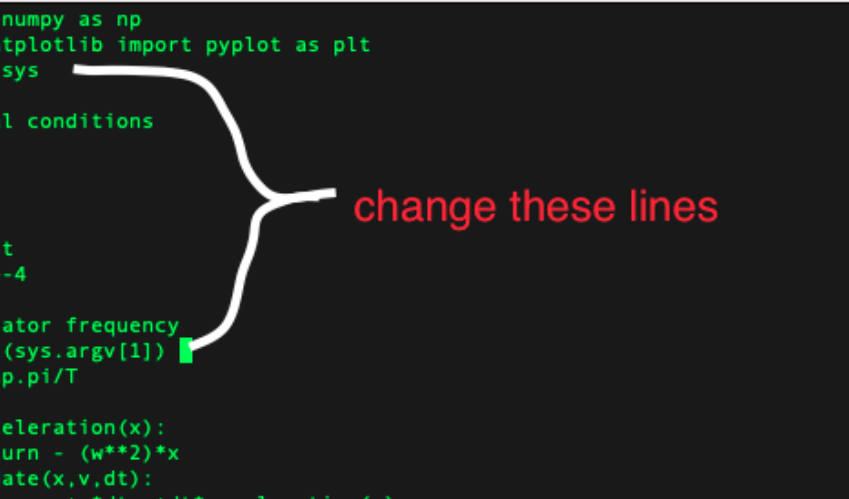
Analogous to how you did your first bash script, create and run another bash script

[illegible]

Once you run it, did it do what you expected?

2 - Change you python file on the terminal

Using vim make the following changes to the file `harmonic_plots.py`



```
LAB2 — vim harmonic_oscillator.py — 80x22

import numpy as np
from matplotlib import pyplot as plt
import sys

#Initial conditions
x=1.
v=0.
t=0.

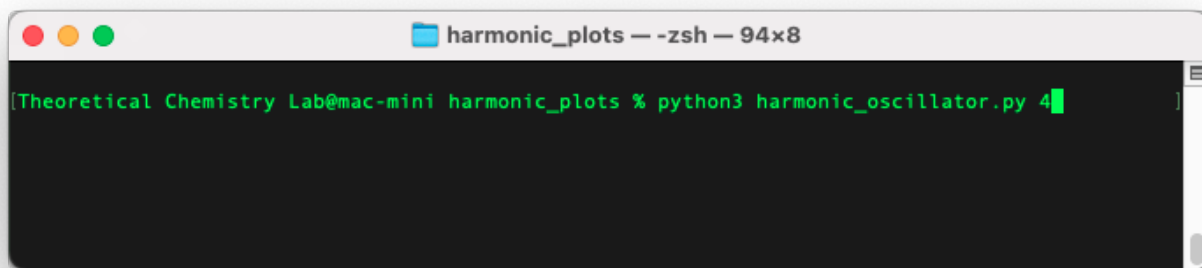
#Delta t
dt = 1e-4

#Oscillator frequency
T = int(sys.argv[1])
w = 2*np.pi/T

def acceleration(x):
    return -(w**2)*x
def update(x,v,dt):
    return x+v*dt,v+dt*acceleration(x)

-- INSERT --
```

Then, run the file as follows

A terminal window titled "harmonic_plots — zsh — 94x8" is shown. The prompt is "[Theoretical Chemistry Lab@mac-mini harmonic_plots %]". The command "python3 harmonic_oscillator.py 4" has been entered, and a green cursor is at the end of the line.

```
harmonic_plots — zsh — 94x8
[Theoretical Chemistry Lab@mac-mini harmonic_plots %] python3 harmonic_oscillator.py 4
```

What did this do?

3 - Run many harmonic oscillators

Based on the results of 1 and 2, create a bash script to run `harmonic_plots.py` with different harmonic oscillator periods (from 1 to 10) automatically. Open the images generated and tell