

Lecture 7 – Code Workshop



Naveh Porat

June 2022

Learning Code

- A large portion of the course is dedicated to code.
- Today we will see some useful tools and coding concepts.
- It's very hard to teach\learn code because it tends to be very technical.
- We'll try to have everything demonstrated live.
- Try to follow along and ask as many questions as you can!

What is good code?

We would like to write code that :

1. Is easy to use
2. Is easy to read (by others)
3. Is easy to modify
4. Is fast

Table of Contents

01

Pycharm

02

VMs

03

Bash

04

Coding Practices

05

Debugging

06


Loggers

Pycharm - Shortcuts



Text Editing


Ctrl + right/left

```
def foo():  
     long_text = "This is a |text containing many words"  
    words = []  
    for i in long_text.split(" "):  
        words.append(i)  
    return words
```

Home & End

```
def foo():  
    = long_text = "This is a text containing many words"  
    words = []  
    for i in long_text.split(" "):  
        words.append(i)  
    return words
```


Ctrl + w

```
def foo():  
    long_text = "This is a text containing many words"  
    words = []  
     for i in long_text.split(" |"):  
        words.append(i)  
    return words
```

Ctrl + c & Ctrl + v

```
def foo():  
    long_text = "This| is a text containing many words"  
    words = []  
    for i in long_text.split(" "):  
        words.append(i)  
    return words
```

Ctrl + x & Ctrl +v

```
def foo():  
    long_text = "This| is a text containing many words"  
    words = []  
    for i in long_text.split(" "):  
        words.append(i)  
    return words
```

Ctrl + d

```
def foo():  
    long_text = "This is a text containing many words"  
    words = []  
    for i in long_text.split(" "):  
        words.append(i)  
    return words
```


Code Folding

```
- def foo(text):  
  letters = []  
  - for word in text.split(" "):  
    - for letter in word:  
      - letters.append(letter)  
  - return letters  
  
- def foo_2(text):  
  letters = foo(text)  
  - print(letters)
```

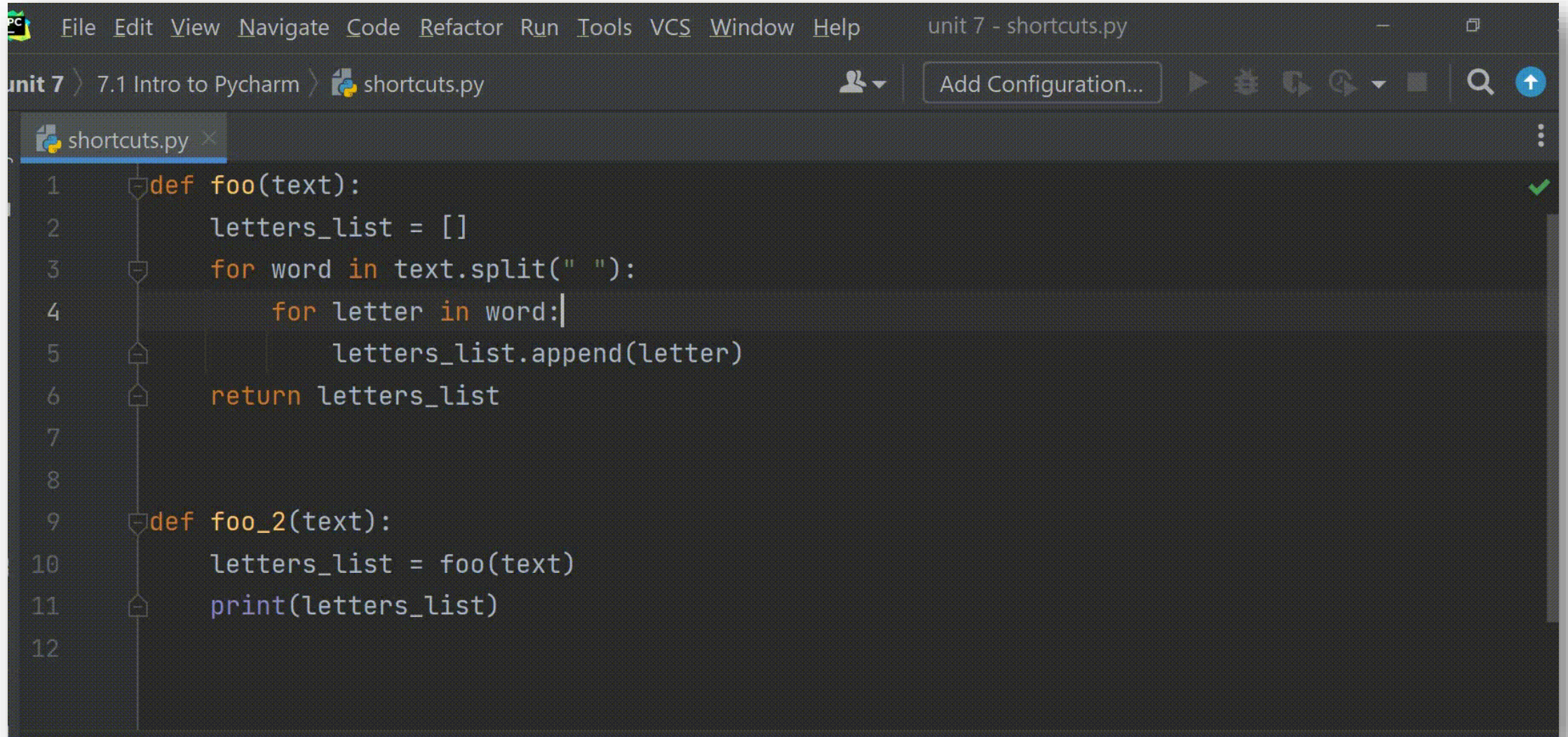
Commenting

```
def foo():  
    long_text = "This is| a text containing many words"  
    words = []  
    for i in long_text.split(" "):  
        words.append(i)  
    return words
```

Search & Replace

```
shortcuts.py x
1  def foo(text):
2      letters = []
3      for word in text.split(" "):
4          for letter in word:
5              letters.append(letter)
6      return letters
7
8
9  def foo_2(text):
10     letters = foo(text)
11     print(letters)
12
```

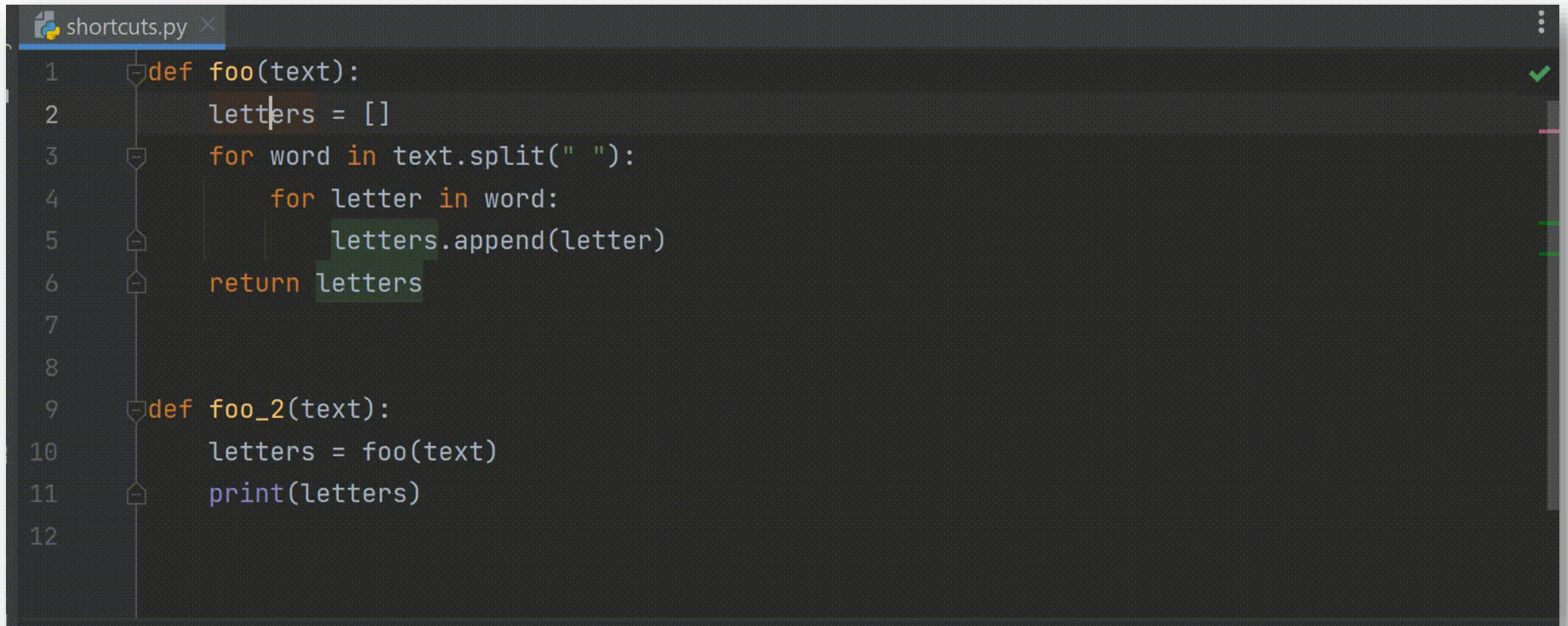

Global Search & Replace



```
unit 7 > 7.1 Intro to Pycharm > shortcuts.py

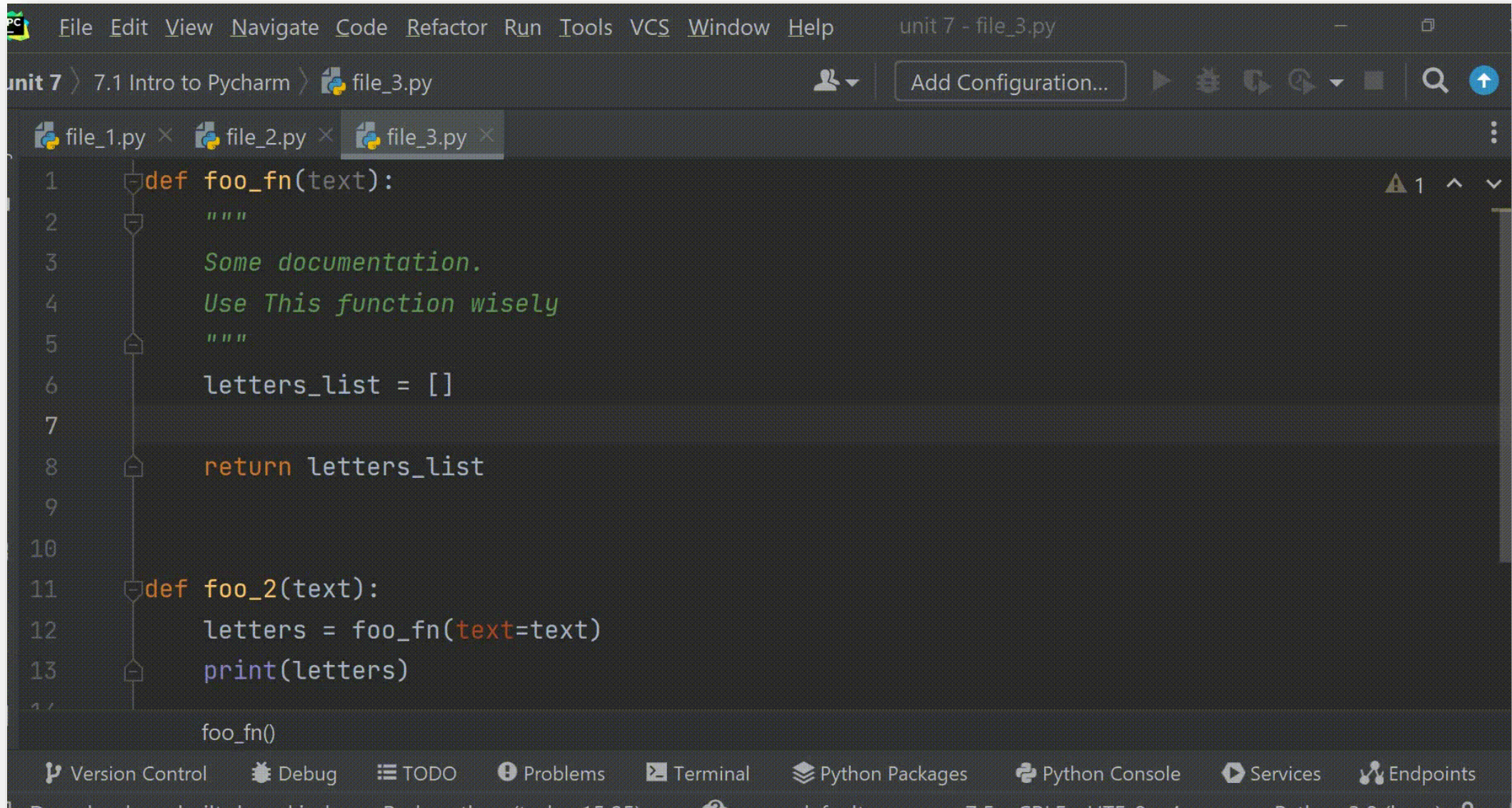
shortcuts.py x
1  def foo(text):
2      letters_list = []
3      for word in text.split(" "):
4          for letter in word:
5              letters_list.append(letter)
6      return letters_list
7
8
9  def foo_2(text):
10     letters_list = foo(text)
11     print(letters_list)
12
```


Rename

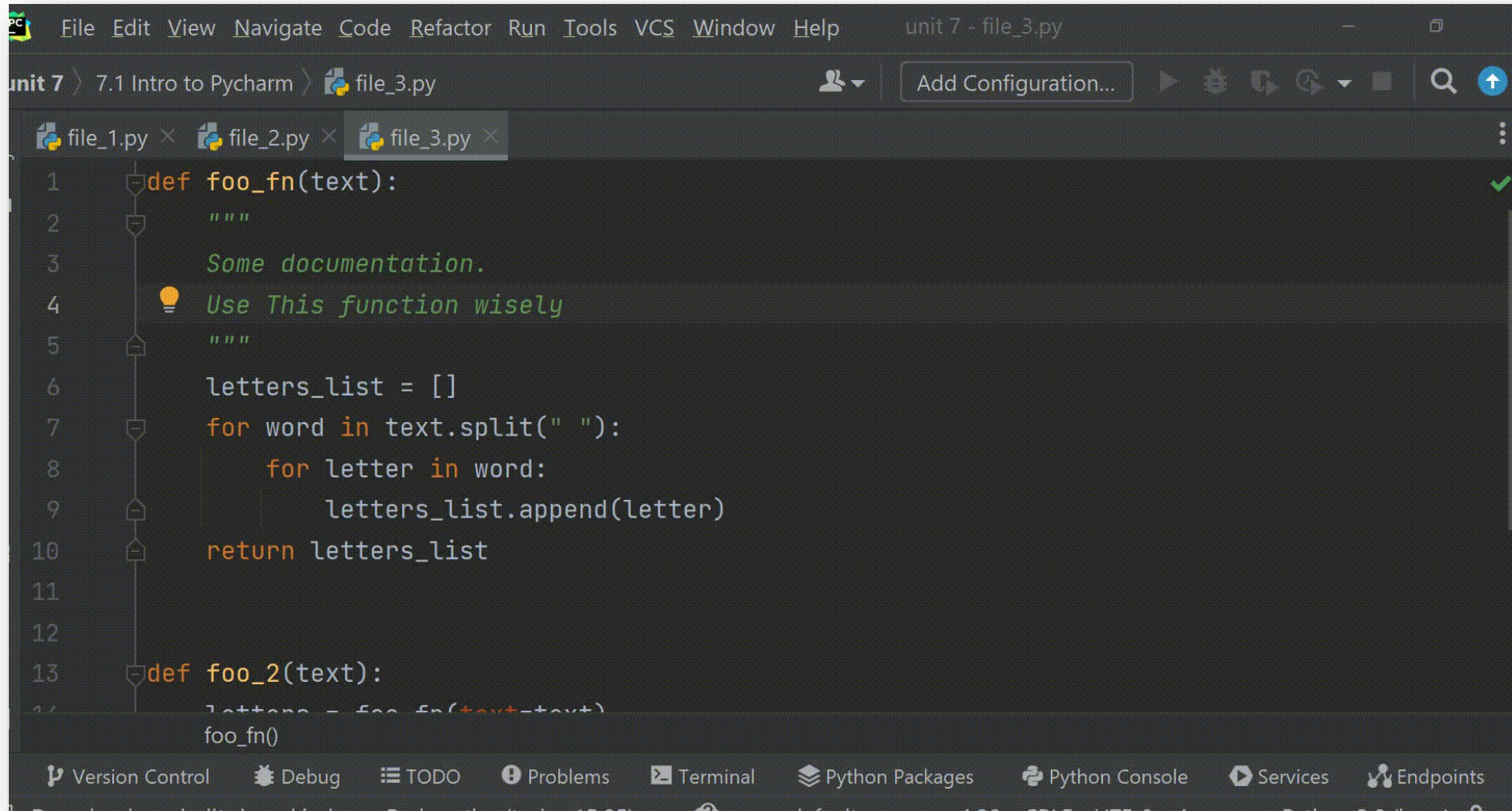


```
shortcuts.py x
1 def foo(text):
2     letters = []
3     for word in text.split(" "):
4         for letter in word:
5             letters.append(letter)
6     return letters
7
8
9 def foo_2(text):
10     letters = foo(text)
11     print(letters)
12
```

Smart Completion



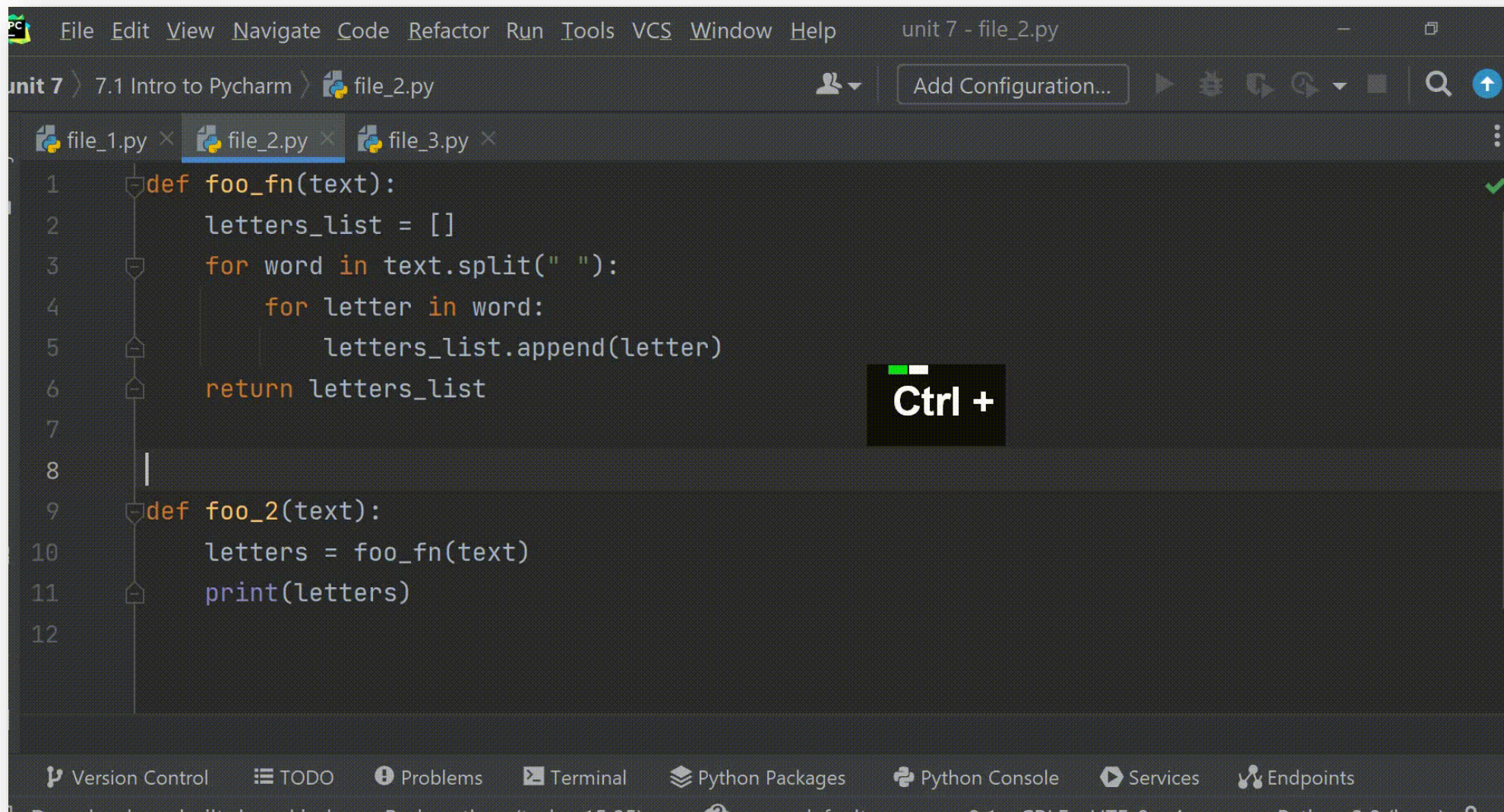
Documentation, Param & Referencing



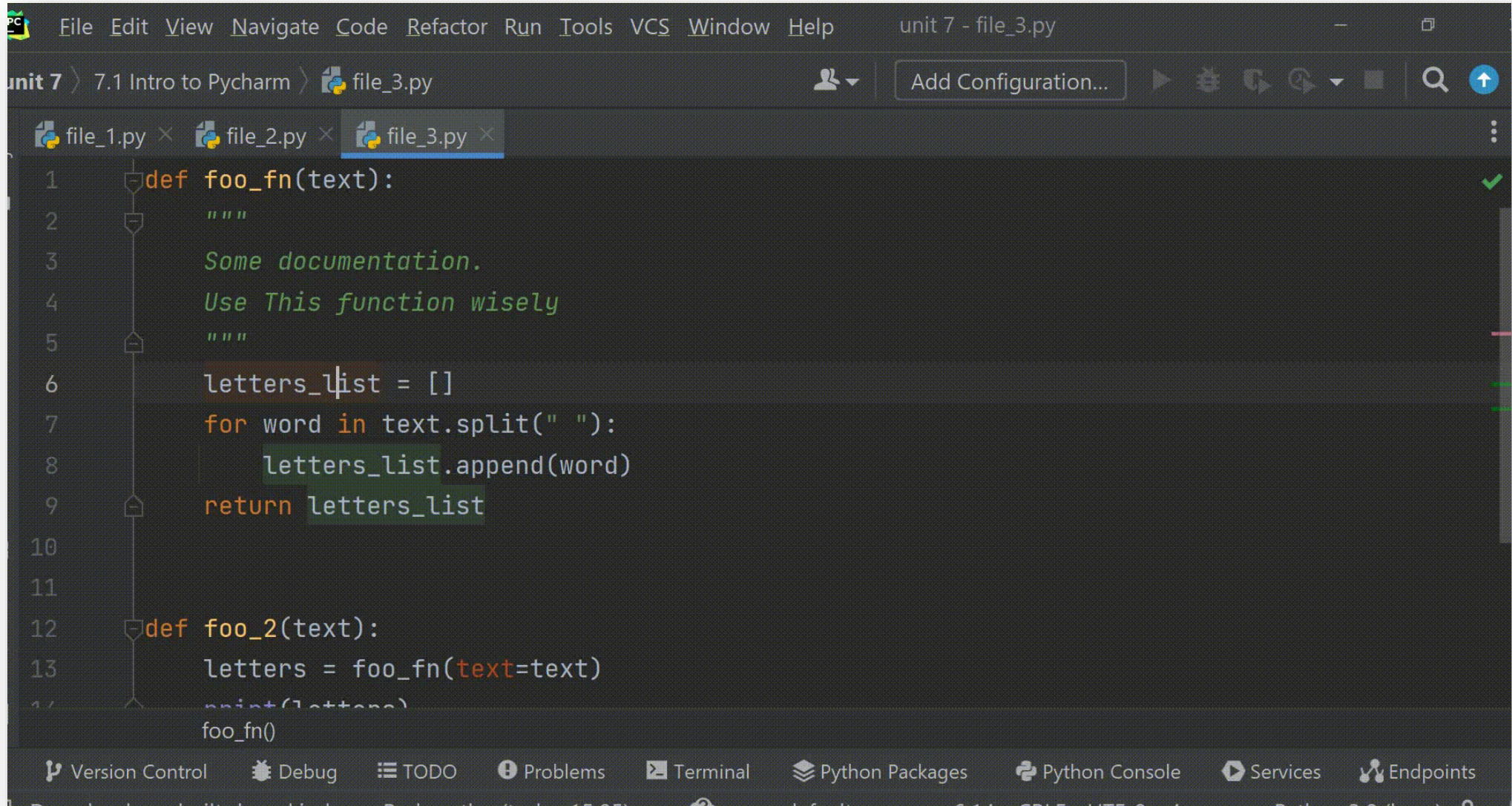
The screenshot shows the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The title bar indicates the current file is 'unit 7 - file_3.py'. The breadcrumb navigation shows 'unit 7 > 7.1 Intro to Pycharm > file_3.py'. The toolbar contains an 'Add Configuration...' button and several icons for running, debugging, and other actions. The main editor window displays the code for 'file_3.py'. The code defines a function 'foo_fn(text)' with a docstring that includes a lightbulb icon and the text 'Use This function wisely'. The function body initializes a list 'letters_list', iterates over the words in 'text', and appends each letter to the list. A second function 'foo_2(text)' is partially visible at the bottom, calling 'foo_fn'.

```
1 def foo_fn(text):
2     """
3     Some documentation.
4     Use This function wisely
5     """
6     letters_list = []
7     for word in text.split(" "):
8         for letter in word:
9             letters_list.append(letter)
10    return letters_list
11
12
13 def foo_2(text):
14     letters = foo_fn(text)
15     foo_fn()
```


Navigating Windows



Search Actions & Create Shortcuts





Code With Me

Code With Me

- Code With Me is a PyCharm plugin that will allow you to code collaboratively, like in google docs.



```
Input.java
1 package com.company;
2
3 import java.util.Scanner;
4
5 Guest
6 public class Input {
7
8     public static void main(String[] args) {
```

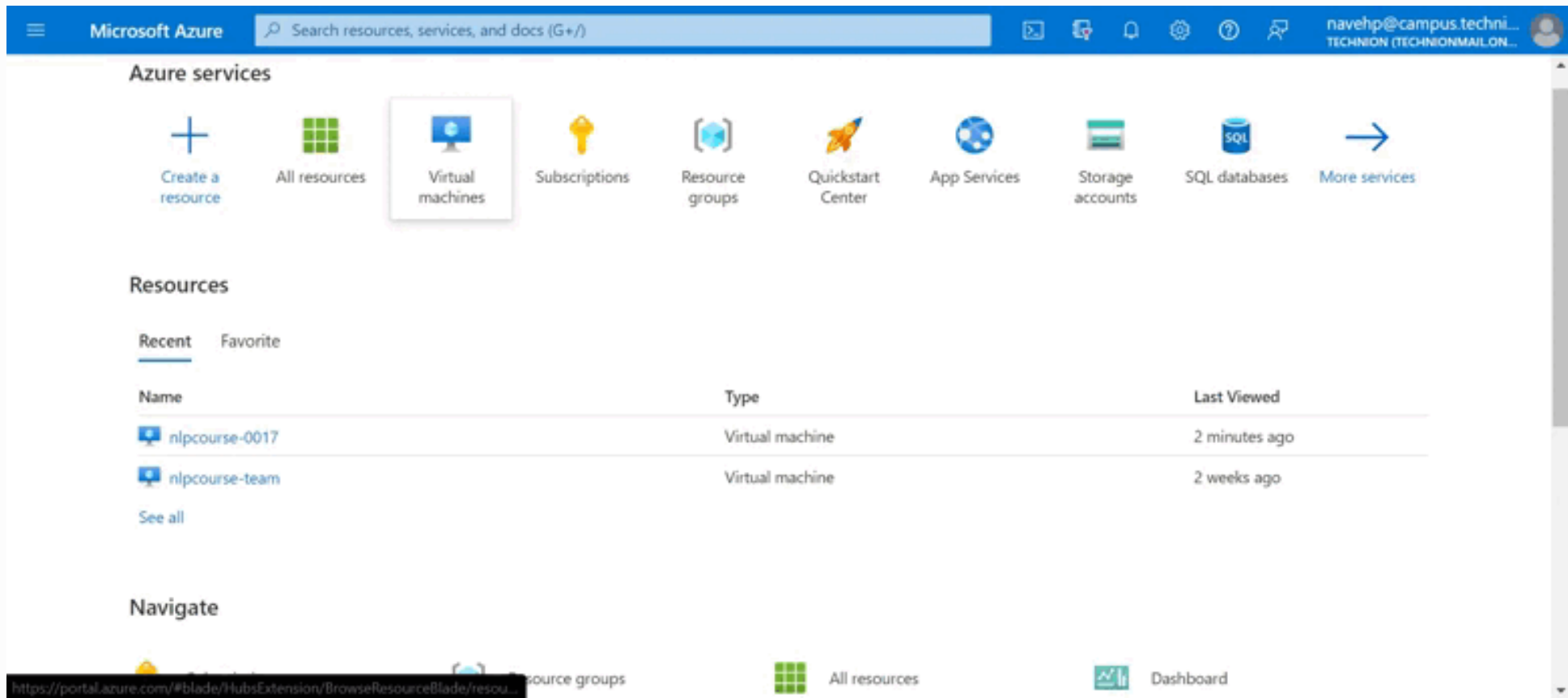
Working with VMs

Virtual Machines

- Pycharm Professional allows you to work on a remote virtual machine.
- The following slides will take you step by step in the process of configuring a remote machine in PyCharm.

Turn on the Machine

- Visit the Azure Portal at <https://portal.azure.com/#home>
- Go into the **Virtual Machines** section.



The screenshot shows the Microsoft Azure Portal interface. At the top, there is a blue header bar with the Microsoft Azure logo, a search bar, and user information. Below the header, the 'Azure services' section is displayed, featuring a grid of icons for various services. The 'Virtual machines' icon, which depicts a computer monitor with a blue square, is highlighted with a red rectangular border. Other visible icons include 'Create a resource', 'All resources', 'Subscriptions', 'Resource groups', 'Quickstart Center', 'App Services', 'Storage accounts', 'SQL databases', and 'More services'. Below the services grid, the 'Resources' section is visible, with tabs for 'Recent' and 'Favorite'. Under the 'Recent' tab, a table lists two virtual machines: 'nlpcourse-0017' and 'nlpcourse-team', both of type 'Virtual machine'. The 'Last Viewed' column shows '2 minutes ago' and '2 weeks ago' respectively. At the bottom, the 'Navigate' section shows a list of resource groups, with 'All resources' and 'Dashboard' also visible.

Microsoft Azure

Search resources, services, and docs (G+)

navehp@campus.techni...
TECHNION (TECHNIONMAILON...)

Azure services

Create a resource

All resources

Virtual machines

Subscriptions

Resource groups

Quickstart Center

App Services

Storage accounts

SQL databases

More services

Resources

Recent Favorite

Name	Type	Last Viewed
nlpcourse-0017	Virtual machine	2 minutes ago
nlpcourse-team	Virtual machine	2 weeks ago

See all

Navigate

source groups

All resources

Dashboard

Turn on the Machine

- Check you're on the courses subscription.
- Enter your machine's page and click **start**.

Microsoft Azure

Search resources, services, and docs (G+)

Home >

Virtual machines

Technion (technionmail.onmicrosoft.com)

+ Create ▾ Switch to classic Reservations ▾ Manage view ▾ Refresh Export to CSV Open query Assign tags Start Restart Stop ...

Filter for any field... Subscription equals **NLP course 4 Rafael** Type equals **all** Resource group equals **all** X Location equals **all** X Add filter

No grouping ▾ List view ▾

<input type="checkbox"/> Name ↑↓	Type ↑↓	Subscription ↑↓	Resource group ↑↓	Location ↑↓	Status ↑↓	Operating system ↑↓	Size ↑↓
<input type="checkbox"/> nlpcourse-0013	Virtual machine	NLP course 4 Rafael	nlp4rcourse-gpu-lab----	East US	Stopped (deallocated)	Linux	Standard_NV12s_v3
<input type="checkbox"/> nlpcourse-0014	Virtual machine	NLP course 4 Rafael	nlp4rcourse-gpu-lab----	East US	Stopped (deallocated)	Linux	Standard_NV12s_v3
<input type="checkbox"/> nlpcourse-0015	Virtual machine	NLP course 4 Rafael	nlp4rcourse-gpu-lab----	East US	Stopped (deallocated)	Linux	Standard_NV12s_v3
<input type="checkbox"/> nlpcourse-0016	Virtual machine	NLP course 4 Rafael	nlp4rcourse-gpu-lab----	East US	Stopped (deallocated)	Linux	Standard_NV12s_v3
<input checked="" type="checkbox"/> nlpcourse-0017	Virtual machine	NLP course 4 Rafael	nlp4rcourse-gpu-lab----	East US	Stopped (deallocated)	Linux	Standard_NV12s_v3
<input type="checkbox"/> nlpcourse-team	Virtual machine	NLP course 4 Rafael	nlp4rcourse-gpu-lab----	East US	Stopped (deallocated)	Linux	Standard_NV12s_v3

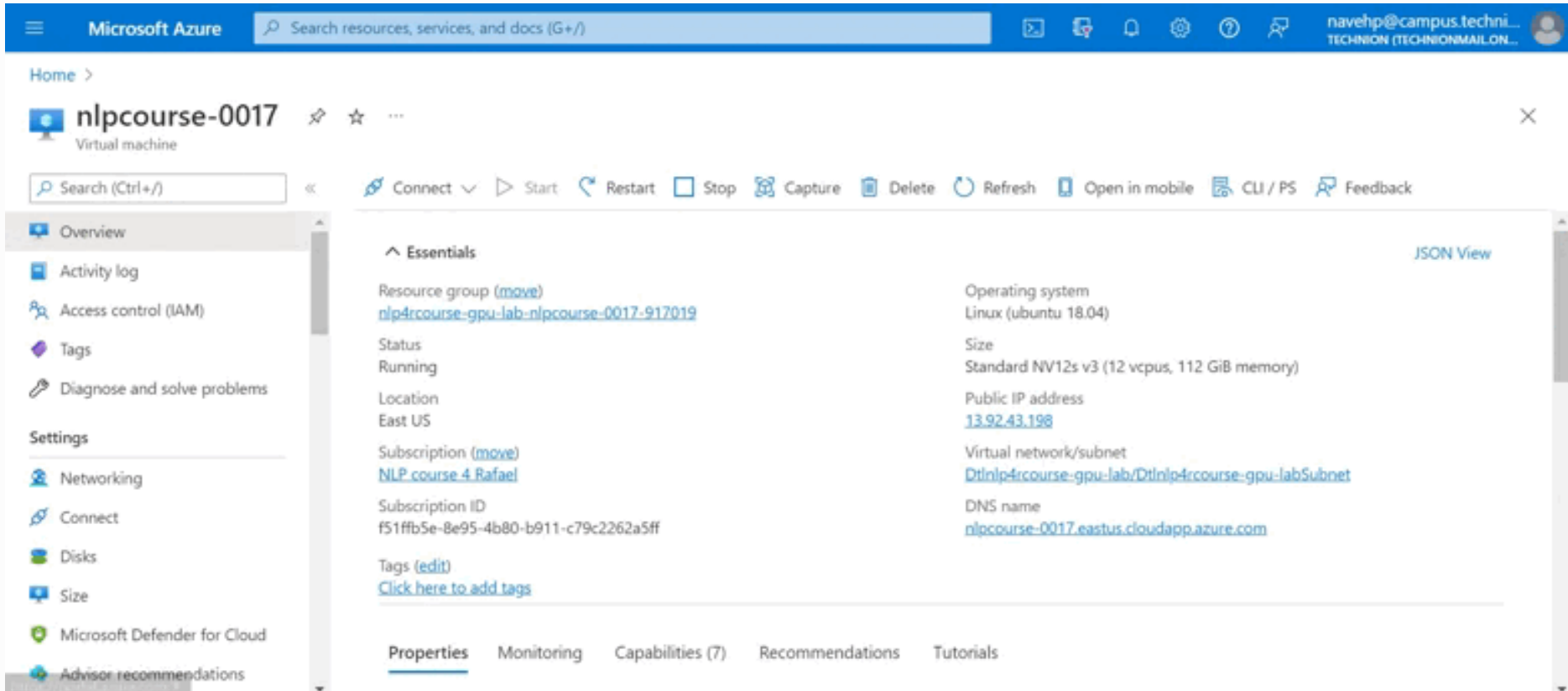
< Previous Page 1 of 1 Next > Showing 1 to 19 of 19 records.

Give feedback

<https://portal.azure.com/#@technionmail.onmicrosoft.com/resource/subscriptions/1511fb5e-8e95-4b80-b911-c79c2262a5ff/resourceGroups/nlp4rcourse-gpu-lab-nlpcourse-0017-917019/providers/Microsoft.Compute/Virtual...>

Turn on the Machine

- Once the machine has started, go into the **connect** section and copy the ssh connection command



The screenshot displays the Microsoft Azure portal interface. At the top, the header includes the 'Microsoft Azure' logo, a search bar, and user information for 'navehp@campus.techni...'. The main content area shows the details for a virtual machine named 'nlpcourse-0017'. The left sidebar contains navigation links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, and Settings. The 'Overview' section is currently selected, showing a table of 'Essentials' with columns for Resource group, Status, Location, Subscription, and Tags. The 'Connect' button is visible in the top right of the overview section. The 'Properties' tab is selected at the bottom of the page.

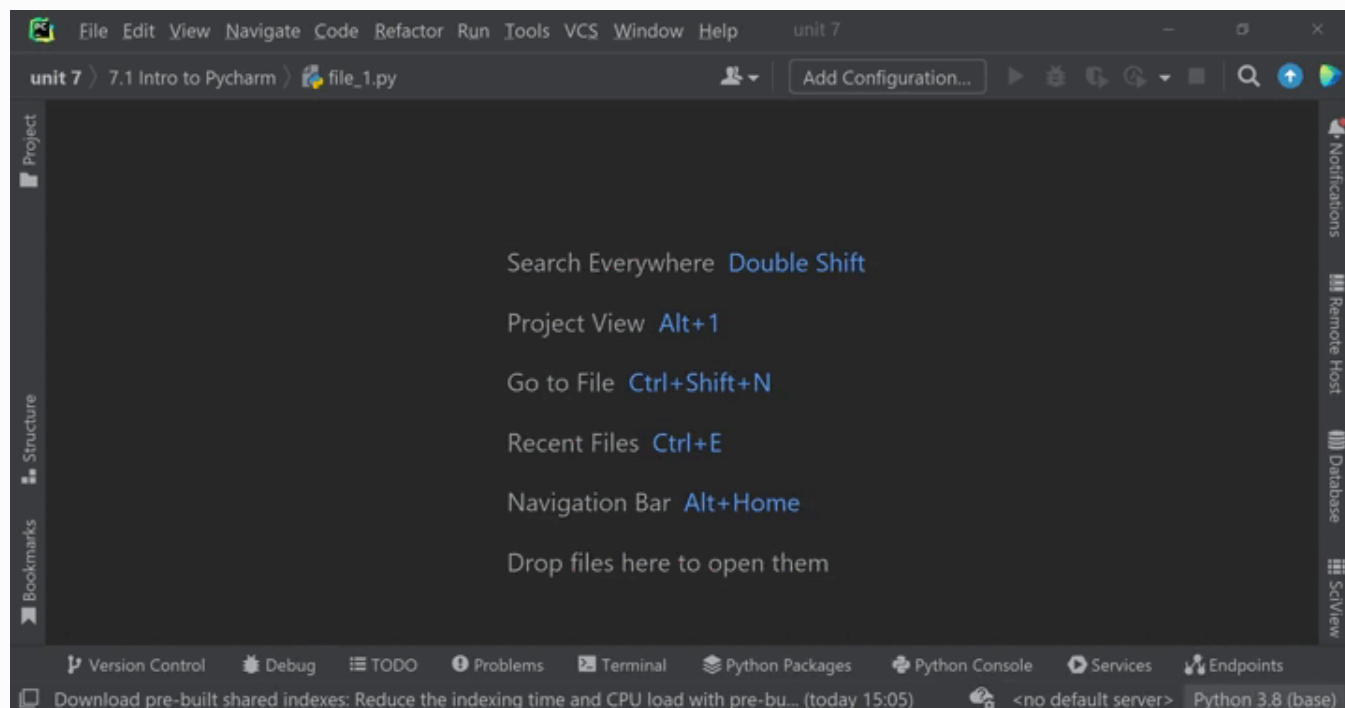
Resource group	Status	Location	Subscription	Tags
nlpcourse-gpu-lab-nlpcourse-0017-917019	Running	East US	NLP course 4 Rafael	Click here to add tags

The 'Connect' section is visible, showing the following details:

- Operating system: Linux (ubuntu 18.04)
- Size: Standard NV12s v3 (12 vcpus, 112 GiB memory)
- Public IP address: [13.92.43.198](#)
- Virtual network/subnet: [Dtnlpcourse-gpu-lab/Dtnlpcourse-gpu-labSubnet](#)
- DNS name: [nlpcourse-0017.eastus.cloudapp.azure.com](#)

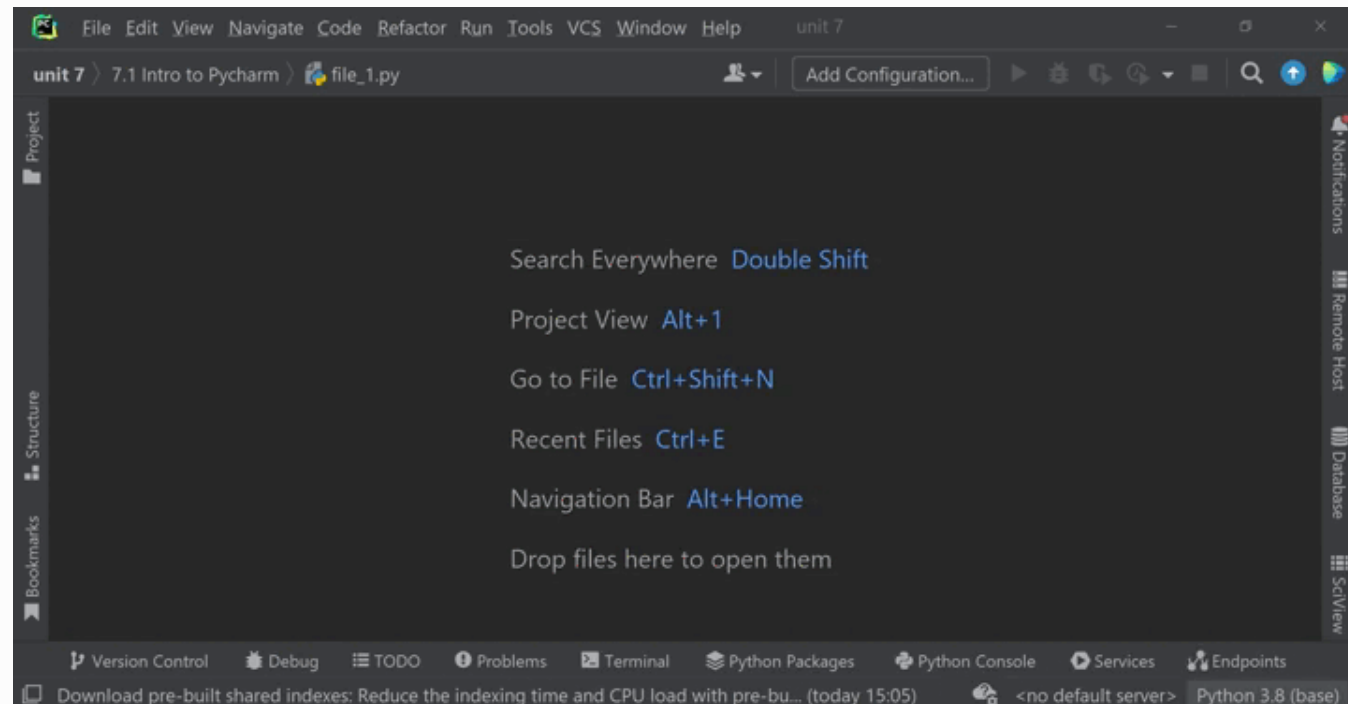
Configure The VM in PyCharm

- Configure a new ssh interpreter.
- The domain is part of the command you copied.
- The username should be **student**
- The password is *Technion2021!*



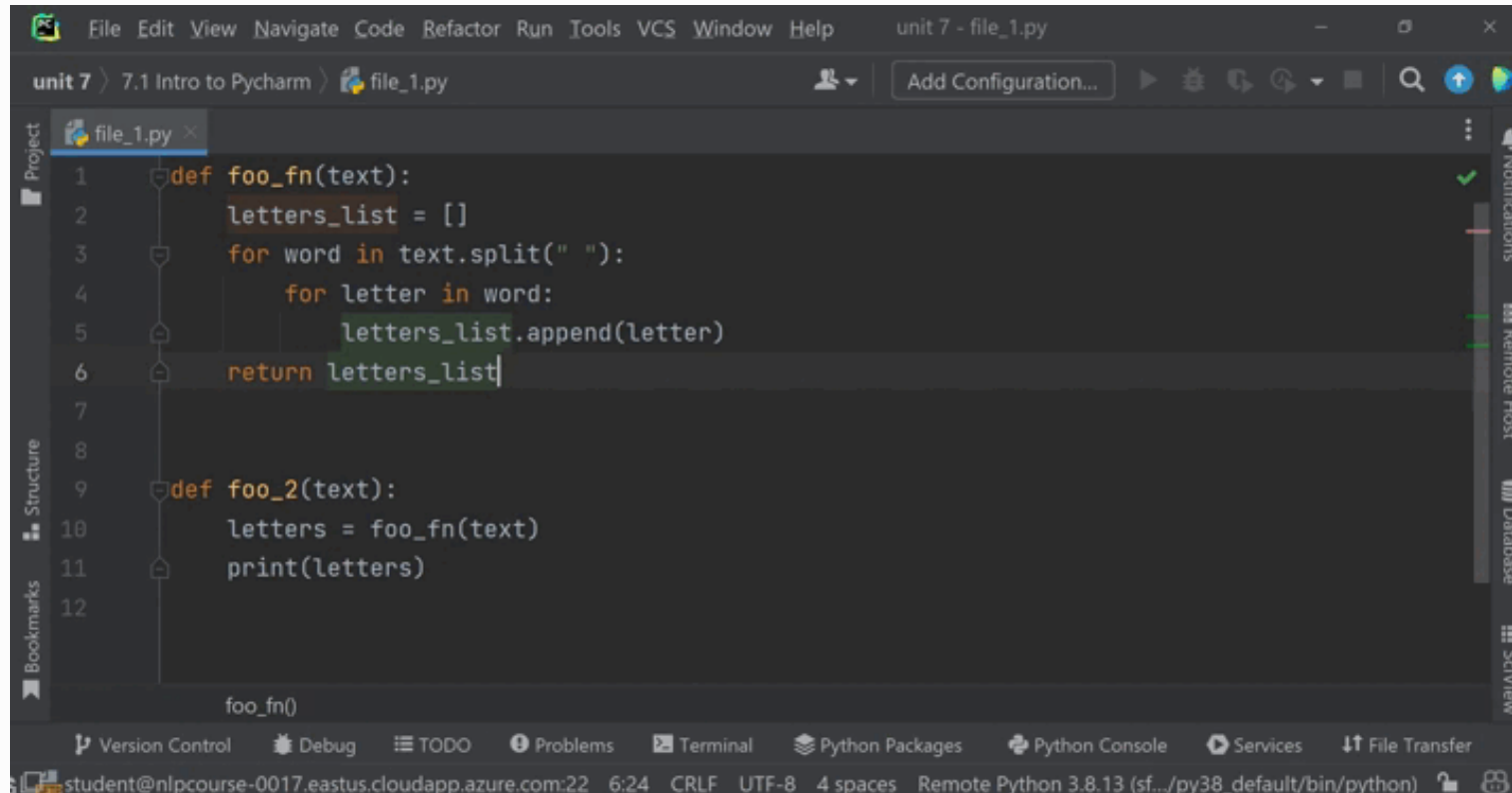
Configure The VM in PyCharm

- Once you connected to the machine, your interpreter will be found at `/anaconda/envs/py38_default/bin/python`
- In the **Sync Folders** section enter your local project directory and the remote one. An example of the remote path would be `/home/student/project_dir`



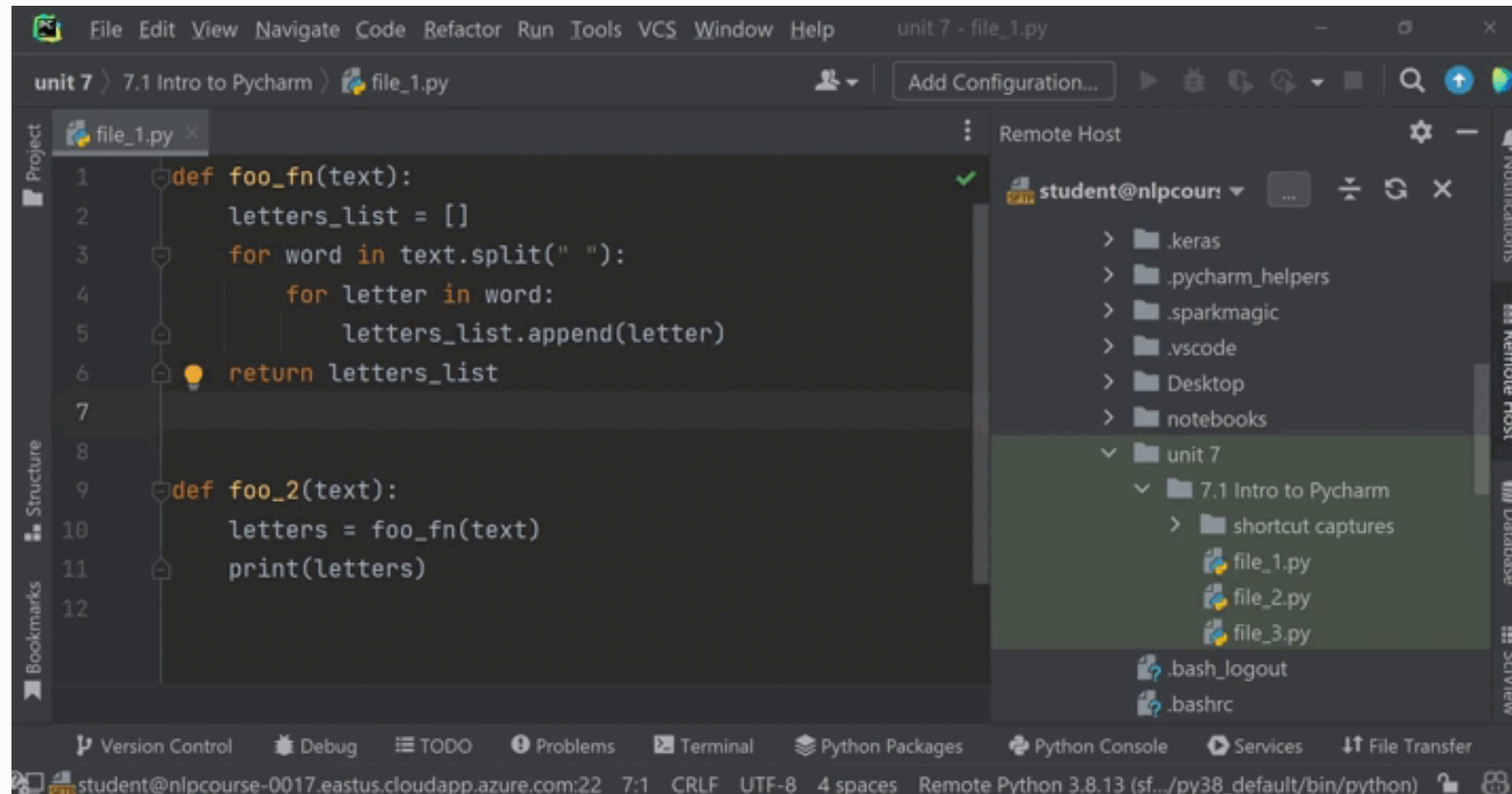
Uploading Files & Browsing

- Even though PyCharm will now automatically sync files from your local dir to the remote one, you can manually upload files using the Ctrl + Alt + Shift + x shortcut.
- You can browse the remote server via the **Remote Host** pane.



SSH Sessions

- You can now run an ssh session of your remote machine using tools > start ssh session.

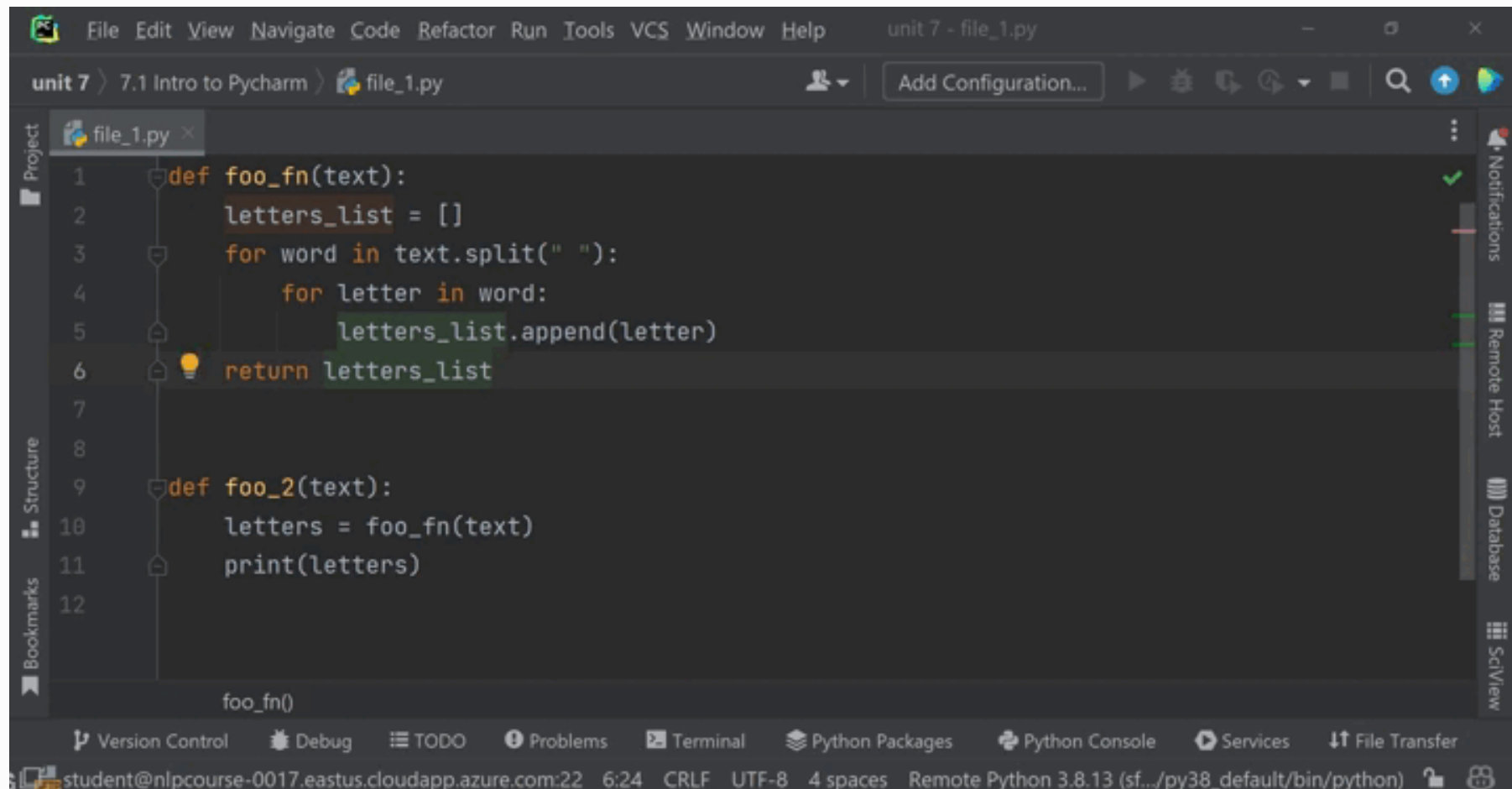


Question

- Configure you own machine!

Question

- Configure **Ctrl + Alt + Shift + s** as a shortcut for starting a ssh session.





Bash

Echo

```
student@nlpcourse-0017:~$ echo hi  
hi
```

Variables

```
student@nlpcourse-0017:~$ msg="hi"
student@nlpcourse-0017:~$ echo "${msg}"
hi
student@nlpcourse-0017:~$ msg="bi"
student@nlpcourse-0017:~$ echo "${msg}"
bi
```

For Loops 1

```
student@nlpcourse-0017:~$ for msg in hi bi
> do
>     echo "${msg}"
> done
hi
bi
```

For Loops 2

```
student@nlpcourse-0017:~$ for i in {1..5}
> do
>   echo "$i"
> done
1
2
3
4
5
```

ls – listing directories

```
student@nlpcourse-0017:~$ ls  
Desktop  notebooks  'unit 7'
```


ls – listing directories

```
student@nlpcourse-0017:~$ ls 'unit 7'
```

```
'7.1 Intro to Pycharm'
```

```
tutorial_04
```

```
'Lecture 4 - Intro to DL.pptx'
```

```
'~$Lecture 4 - Intro to DL.pptx'
```

```
'recitation_04 - intro to pytorch'
```

Mkdir – making directories

```
student@nlpcourse-0017:~$ mkdir tmp
```

```
student@nlpcourse-0017:~$ ls
```

```
Desktop  notebooks  tmp  'unit 7'
```

cd – changing working directory

```
student@nlpcourse-0017:~$ cd tmp/  
student@nlpcourse-0017:~/tmp$ ls  
student@nlpcourse-0017:~/tmp$ █
```

Changing to parent directory

```
student@nlpcourse-0017:~/tmp$ cd ../  
student@nlpcourse-0017:~$ ls  
Desktop  notebooks  tmp  'unit 7'
```

cp – copy files

```
student@nlpcourse-0017:~$ echo "hi" > tmp/hi.txt  
student@nlpcourse-0017:~$ cp tmp/hi.txt tmp/bi.txt  
student@nlpcourse-0017:~$ ls tmp/  
bi.txt  hi.txt
```

cp – copy files

```
student@nlpcourse-0017:~$ cp tmp/ tmp2/
cp: -r not specified; omitting directory 'tmp/'
student@nlpcourse-0017:~$ cp -r tmp/ tmp2/
student@nlpcourse-0017:~$ ls tmp2/
bi.txt  hi.txt
```

mv – moving and renaming files

```
student@nlpcourse-0017:~$ ls
Desktop  notebooks  tmp  tmp2  'unit 7'
student@nlpcourse-0017:~$ mv tmp2/ tmp3/
student@nlpcourse-0017:~$ ls
Desktop  notebooks  tmp  tmp3  'unit 7'
```

rm – removing files

```
student@nlpcourse-0017:~$ ls tmp/
```

```
bi.txt  hi.txt
```

```
student@nlpcourse-0017:~$ rm tmp/bi.txt
```

```
student@nlpcourse-0017:~$ ls tmp/
```

```
hi.txt
```


rm – removing directories

```
student@nlpcourse-0017:~$ ls
Desktop  notebooks  tmp  tmp3  'unit 7'
student@nlpcourse-0017:~$ rm tmp3/
rm: cannot remove 'tmp3/': Is a directory
student@nlpcourse-0017:~$ rm -r tmp3/
student@nlpcourse-0017:~$ ls
Desktop  notebooks  tmp  'unit 7'
```

pwd – getting absolute path

```
student@nlpcourse-0017:~$ pwd  
/home/student
```

curl – get web pages

```
student@nlpcourse-0017:~$ curl https://gist.githubusercontent.com/simonsarris/9980a385af4f4c4d3967/  
raw/e4ba954591aaa5de8e4d36a4952e6229402e0fbd/poem.txt
```

by Manuel Gutiérrez Nájera

I want to die as the day declines,
at high sea and facing the sky,
while agony seems like a dream
and my soul like a bird that can fly.

curl – save web pages

```
student@nlpcourse-0017:~$ curl https://gist.githubusercontent.com/simonsarris/9980a385af4f4c4d3967/
raw/e4ba954591aaa5de8e4d36a4952e6229402e0fbd/poem.txt --output poem.txt
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100    625    100    625     0     0  49856      0  --:--:-- --:--:-- --:--:-- 52083
student@nlpcourse-0017:~$ echo poem.txt
poem.txt
```

find – finding files

```
student@nlpcourse-0017:~$ find tmp/  
tmp/  
tmp/hi.txt
```

find – finding files

```
student@nlpcourse-0017:~$ find -name "hi.txt"  
./tmp/hi.txt
```

find – finding files

```
student@nlpcourse-0017:~$ find -name "*.txt"
./.pycharm_helpers/pydev/pydevd_attach_to_process/README.txt
./.pycharm_helpers/pydev/pydev_sitecustomize/__not_in_default_pythonpath.txt
./.pycharm_helpers/pydev/merge_pydev_pycharm.txt
./.pycharm_helpers/pydev/build_tools/pydevd_release_process.txt
./.pycharm_helpers/icon-robots.txt
./.pycharm_helpers/build.txt
./.pycharm_helpers/pycharm/teamcity/README.txt
./.pycharm_helpers/pycharm/django_manage_commands_provider/readme.txt
./.pycharm_helpers/pycharm/__jb_for_twisted/twisted/plugins/README.txt
./.pycharm_helpers/generator3/version.txt
./tmp/hi.txt
```

nvidia-smi – monitoring gpus

```
(cpada) [navehp@nlp13 ~]$ nvidia-smi
Mon Jul  4 20:43:29 2022
```

NVIDIA-SMI 460.32.03 Driver Version: 460.32.03 CUDA Version: 11.2									
GPU	Name		Persistence-M	Bus-Id	Disp.A	Volatile	Uncorr.	ECC	
Fan	Temp	Perf	Pwr:Usage/Cap		Memory-Usage	GPU-Util	Compute	M.	
								MIG	M.
0	GeForce	GTX	108...	Off	00000000:17:00.0	Off			N/A
19%	36C	P0	52W / 250W		0MiB / 11178MiB	0%		Default	N/A
1	GeForce	GTX	108...	Off	00000000:65:00.0	Off			N/A
18%	38C	P0	52W / 250W		0MiB / 11178MiB	0%		Default	N/A
Processes:									
GPU	GI	CI	PID	Type	Process name			GPU Memory	
	ID	ID						Usage	
No running processes found									

clear – clearing outputs

```
• SSH-browser      : ✓
• X11-forwarding   : ✗ (disabled or not supported by server)

> For more info, ctrl+click on help or visit our website.
```

Last login: Sun Jul 3 07:02:15 2022 from 132.69.236.214
(cpada) [navehp@nlp13 ~]\$ nvidia-smi
Mon Jul 4 20:43:29 2022

NVIDIA-SMI 460.32.03 Driver Version: 460.32.03 CUDA Version: 11.2									
GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile	Uncorr. ECC			
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage	GPU-Util	Compute M.			
						MIG M.			
0	GeForce GTX 108...	Off	000000000:17:00:0	Off		N/A			
19%	36C	P0	52W / 250W	0MiB / 11178MiB	0%	Default			
						N/A			
1	GeForce GTX 108...	Off	000000000:65:00:0	Off		N/A			
18%	38C	P0	52W / 250W	0MiB / 11178MiB	0%	Default			
						N/A			

Processes:						
GPU	GI	CI	PID	Type	Process name	GPU Memory Usage
	ID	ID				
No running processes found						

(cpada) [navehp@nlp13 ~]\$

aliasing – creating your own commands

```
student@nlpcourse-0017:~$ alias nvid="nvidia-smi"
```

```
student@nlpcourse-0017:~$ nvid
```

```
Mon Jul  4 20:49:51 2022
```

```
+-----+
| NVIDIA-SMI 470.103.01    Driver Version: 470.103.01    CUDA Version: 11.4    |
|-----+-----+-----+
| GPU   Name                Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|                                       |                    |      MIG M.         |
|=====+=====+=====+

```

vi – file editing

- The vi command allows you to inspect and edit files inside bash.
- Run *vi file_path* to enter the vi editor.
- Vi has two modes: command and insert mode.
- To enter insert mode, press *i*.
- After editing your file, press *escape* to exit insert mode.
- You can quit your file in the following ways:
 - :q – quit an unchanged file.
 - :wq – write and quit a changed file.
 - :q! – force quitting a changed file without saving.

Screen – running processes in the background

- Many times, we would like to run processes in the background and have them not shut down once we disconnect ourself from the machine.
- Screen sessions are bash sessions that continue to run in the background.
- To create a new screen session run
screen -S session_name
- To detach from a screen session press
Ctrl a + d
- To reattach to a screen session run
screen -r session_name

Screen – running processes in the background

- To kill a screen session run
screen -XS session_name quit
- Or press **Ctrl a + k**
- Scrolling isn't available inside screen sessions.
 - To move around, enter copy mode by pressing
Ctrl a + [

Sudo

- To install things in bash you need **sudo privileges**.
- You do not have sudo privileges on your machines, but if you run into something you need, contact us.

bash scripts

- You can place bash commands in a shell script and run it:

```
my_script.sh x
1 ▶ echo "This is my script"
2   mkdir some_dir
3   echo "Made a directory called some_dir"
4   rm -r some_dir
5   echo "Removed the directory called some_dir"
```

```
student@nlpcourse-0017:~$ sh my_script.sh
This is my script
Made a directory called some_dir
Removed the directory called some_dir
```

Question

- Write a bash script called “download_alice.sh” that:
 - Creates a directory called *alice*.
 - Saves the alice book from github to the file *alice_book.txt* inside the dir.
 - Renames the file to *alice.txt*.
 - Prints the first 3 lines from the book (search the web!)
 - Prints informative messages between commands.
- Run the script

Solution

download_alice.sh ✕

```
1 ► mkdir alice
2   echo "created the alice dir"
3   curl https://gist.githubusercontent.com/philipj/4944029/raw/75ba2243dd5ec2875f629bf5
4   --output ./alice/alice_book.txt
5   echo "Downloaded alice_book.txt"
6   cp ./alice/alice_book.txt ./alice/alice.txt
7   echo "Renamed alice_book.txt to alice.txt"
8   echo "First 3 line of the book:"
9   head -3 ./alice/alice.txt
```

Solution

```
student@nlpcourse-0017:~$ sh download_alice.sh
```

```
created the alice dir
```

% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current	
			Dload	Upload	Total	Spent	Left	Speed
100	145k	100	145k	0	0	6339k	0	--:--:-- --:--:-- --:--:-- 6308k

```
Downloaded alice_book.txt
```

```
Renamed alice_book.txt to alice.txt
```

```
First 3 line of the book:
```

```
Alice's Adventures in Wonderland
```

```
ALICE'S ADVENTURES IN WONDERLAND
```




Coding Practices

How to Write SOLID Code

- **S**ingle-responsibility principle (SRP)
- **O**pen-closed principle (OCP)
- **L**iskov substitution principle (LSP)
- **I**nterface Segregation Principle (ISP)
- **D**ependency Inversion Principle (DIP)

How to Write SOLID Code

- **Single-responsibility principle (SRP)**
- **Open-closed principle (OCP)**
- **Liskov substitution principle (LSP)**
- **Interface Segregation Principle (ISP)**
- **Dependency Inversion Principle (DIP)**

Single- Responsibility Principle

Every component of your code
should have one and only one
responsibility

Open-Closed Principle

You should not need to modify the code you have already written to accommodate new functionality, but simply add what you now need.

Liskov Substitution Principle

Derived classes must be substitutable for their base classes

How to Write DRY Code

Don't Repeat Yourself

Every piece of knowledge or logic must have a single, unambiguous representation within a system.



Example Project

Debugging

PyCharm Debugger

Live demonstration. A tutorial can be found [here](#).

Some More Good Practices

- Splitting code into logical files.
- `utils.py` file.
- `consts.py` file.
- Using `pathlib` for paths.
- Documentation
- Configurable Code (e.g. YAML)
- Using `argparse` to run code from shell.
- Reformatting the code to PEP-8:
Ctrl + Alt + L in PyCharm

Loggers

Wandb

- Live demonstration.
- A tutorial can be found [here](#).
- **Sweeps** – We will use sweeps late on in the course. A tutorial can be found [here](#).
- **Clear ML** is another (Israeli) logger that has some great features.

Tutorial

Task

- Take the existing project and add the following features:
 - Choice between an RNN/LSTM/GRU in the model.
 - Choice if to load pretrained embedding weights or not.
 - Whether to print the loss and accuracy in the evaluation loop.
 - Write evaluation results to csv
- Bonus: Create a hyperparameter search script that finds good hyperparameters.