Knowledge Café

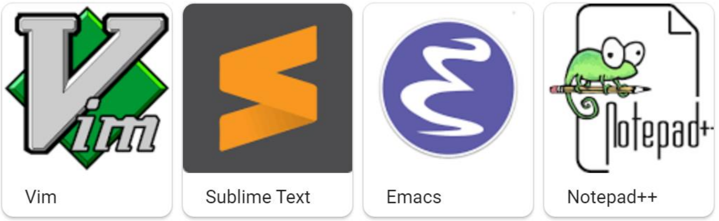
Sadjad Anzabi Zadeh

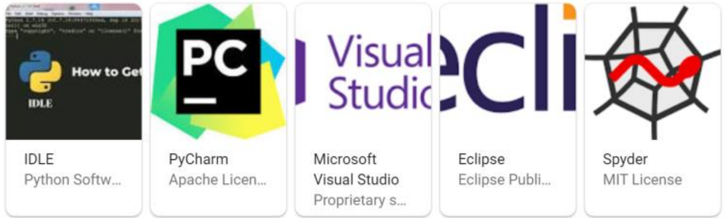
The choice of code editor

Keeping track of code changes

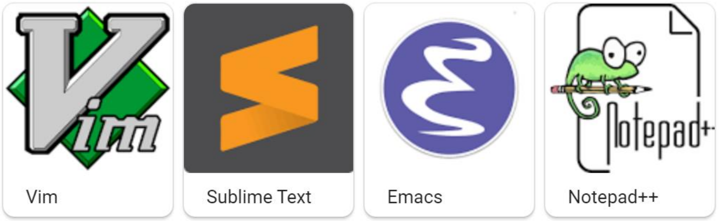
Utilizing Python’s tools and techniques

Managing multiple runs of your code





IDE / Code Editor

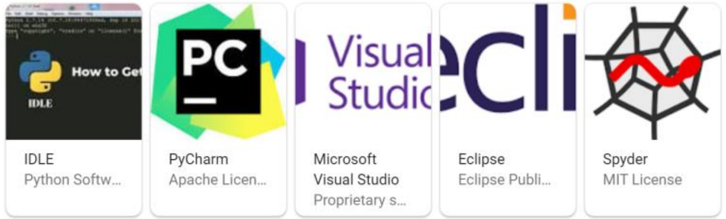


Vim

Sublime Text

Emacs

Notepad++



IDLE
Python Softw...

PyCharm
Apache Licen...

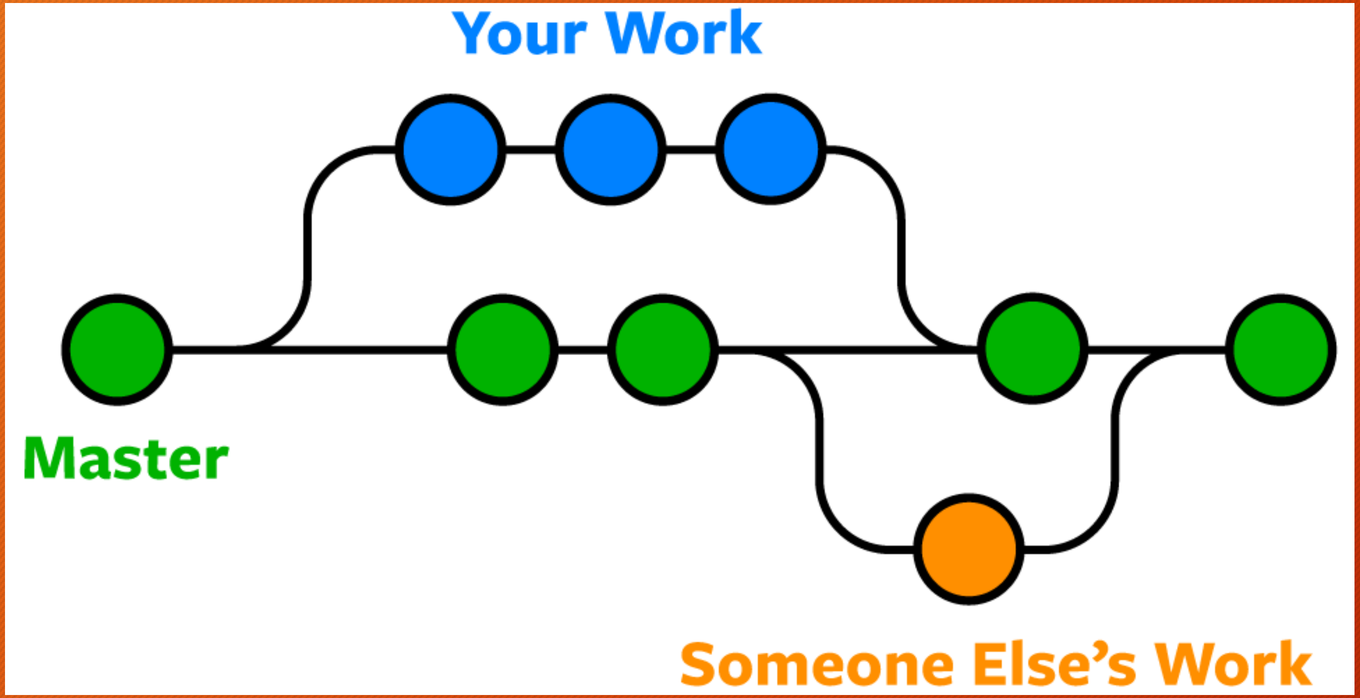
Microsoft
Visual Studio
Proprietary s...

Eclipse
Eclipse Publi...

Spyder
MIT License

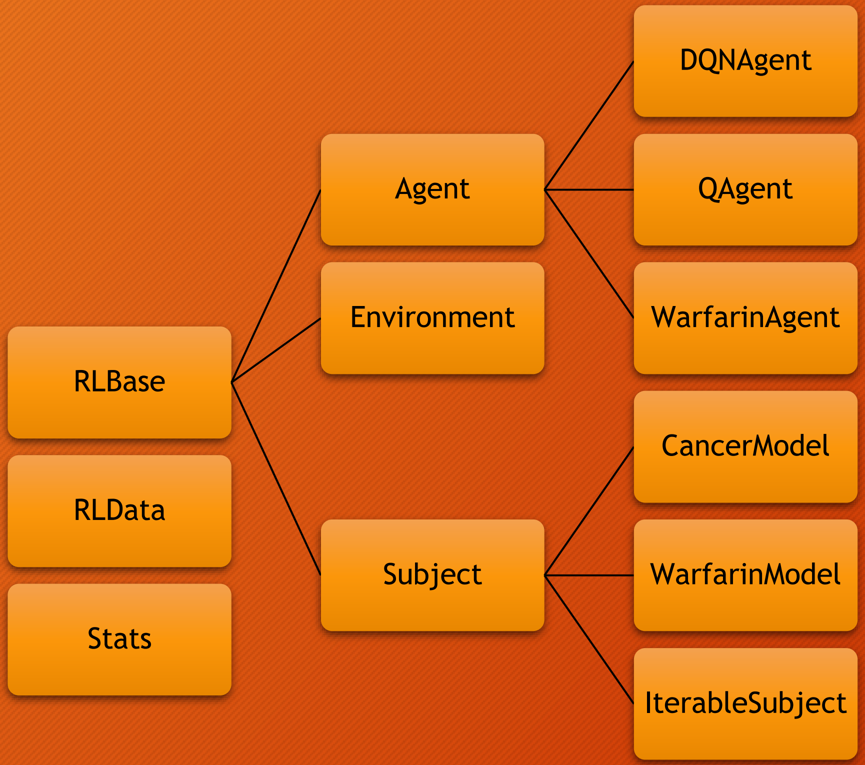
IDE / Code Editor

Version Control



Python Tools and Techniques

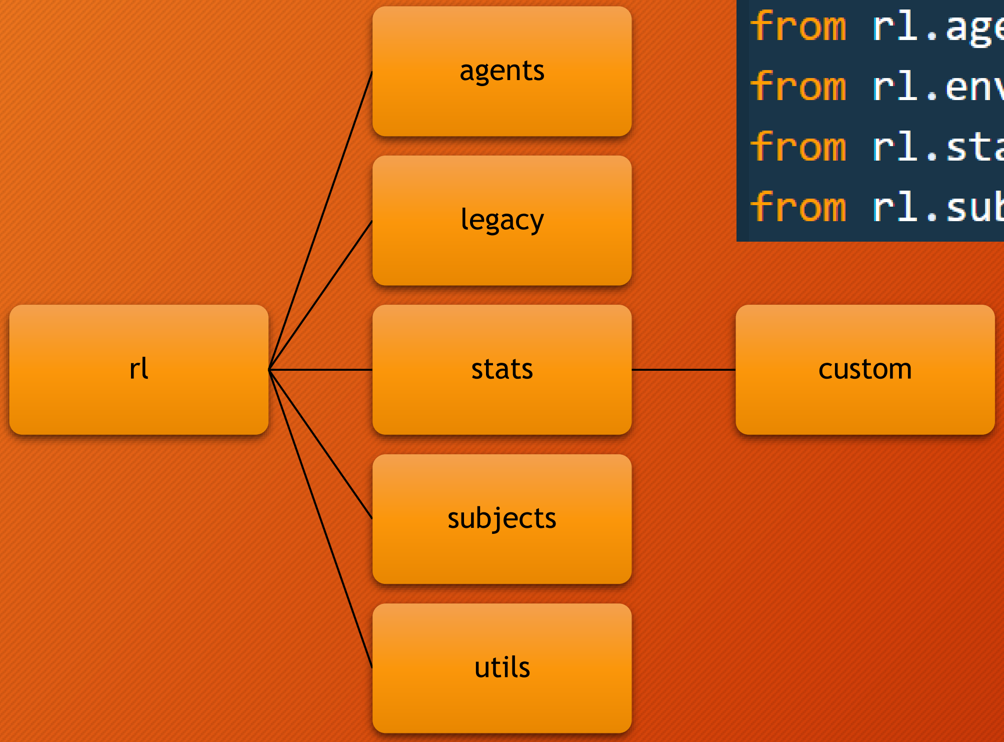
Object Oriented Programming



Example:

```
load(self, **kwargs)
save(self, **kwargs)
```

Python Tools and Techniques
Packages



rl

agents

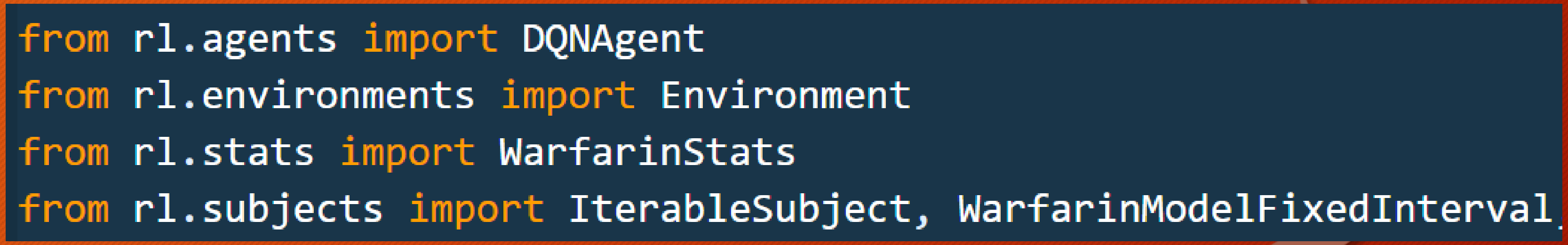
legacy

stats

subjects

utils

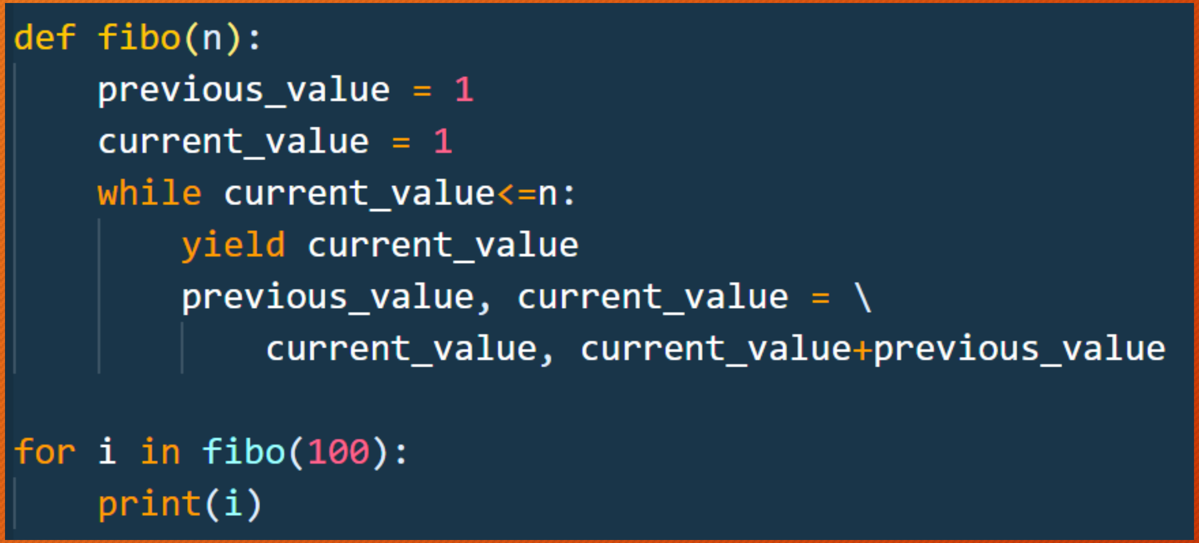
custom



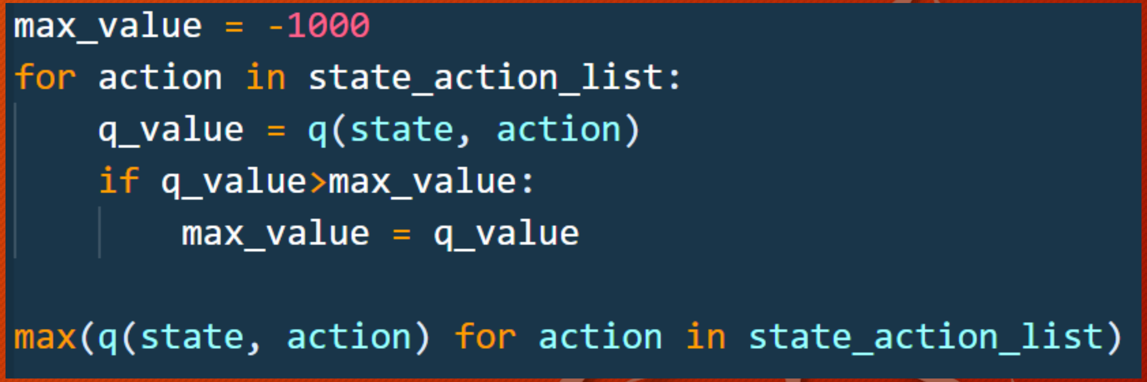
```
from rl.agents import DQNAgent
from rl.environments import Environment
from rl.stats import WarfarinStats
from rl.subjects import IterableSubject, WarfarinModelFixedInterval
```

Python Tools and Techniques

Generators/ List Comprehensions



```python
def fibo(n):
    previous_value = 1
    current_value = 1
    while current_value<=n:
        yield current_value
        previous_value, current_value = \
            current_value, current_value+previous_value

for i in fibo(100):
    print(i)
```

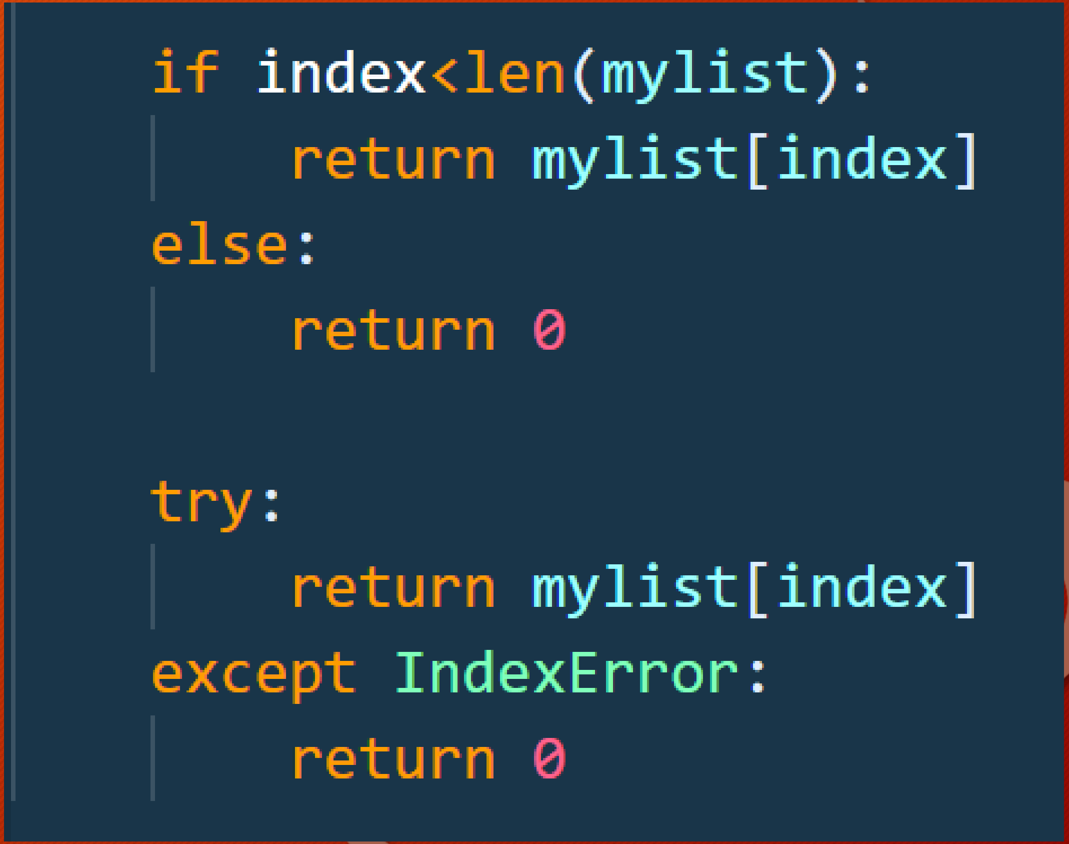


```python
max_value = -1000
for action in state_action_list:
    q_value = q(state, action)
    if q_value>max_value:
        max_value = q_value

max(q(state, action) for action in state_action_list)
```

Python Tools and Techniques

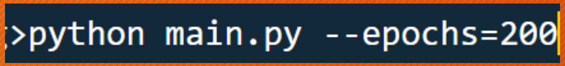
Exception Handling

It is often easier to ask for forgiveness than to ask for permission.
(EAFP)

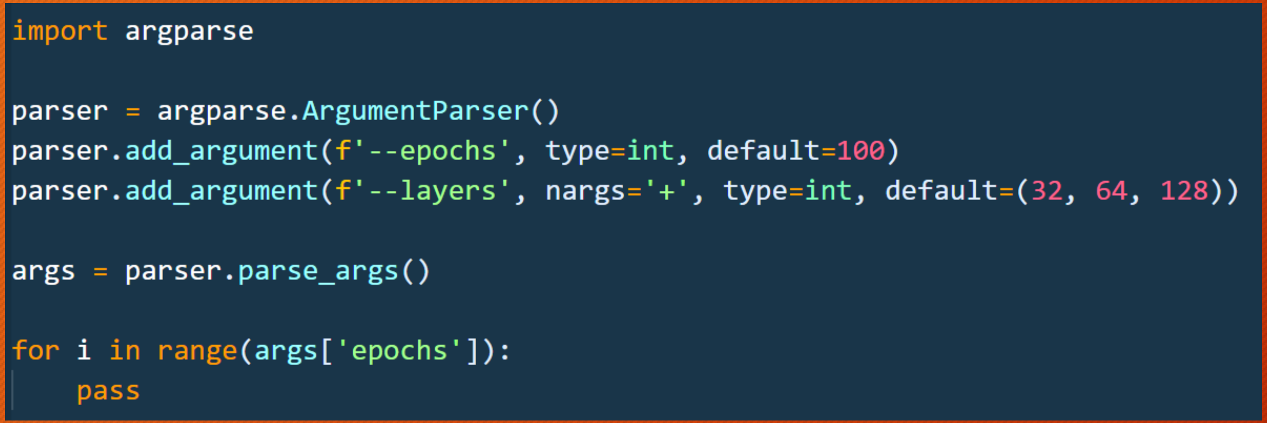


```
if index<len(mylist):
    return mylist[index]
else:
    return 0


try:
    return mylist[index]
except IndexError:
    return 0
```

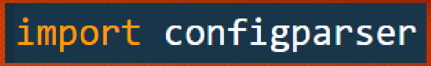
Python Tools and Techniques
Command line arguments



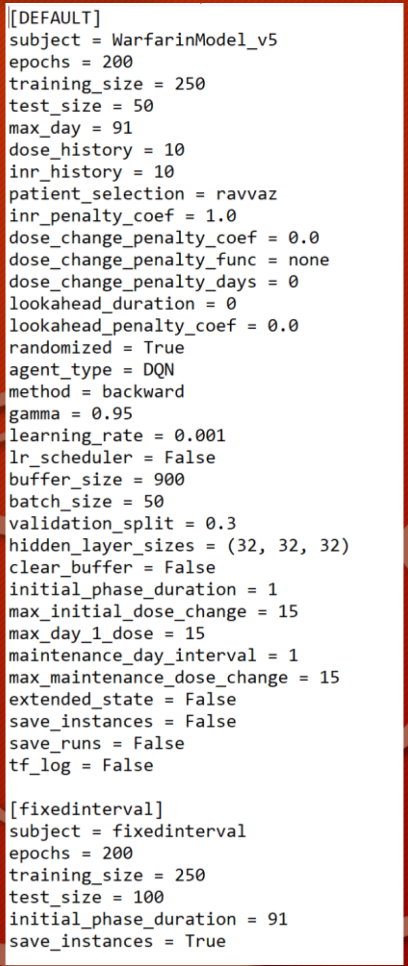
```
>python main.py --epochs=200
```



```python
import argparse

parser = argparse.ArgumentParser()
parser.add_argument(f'--epochs', type=int, default=100)
parser.add_argument(f'--layers', nargs='+', type=int, default=(32, 64, 128))

args = parser.parse_args()

for i in range(args['epochs']):
    pass
```

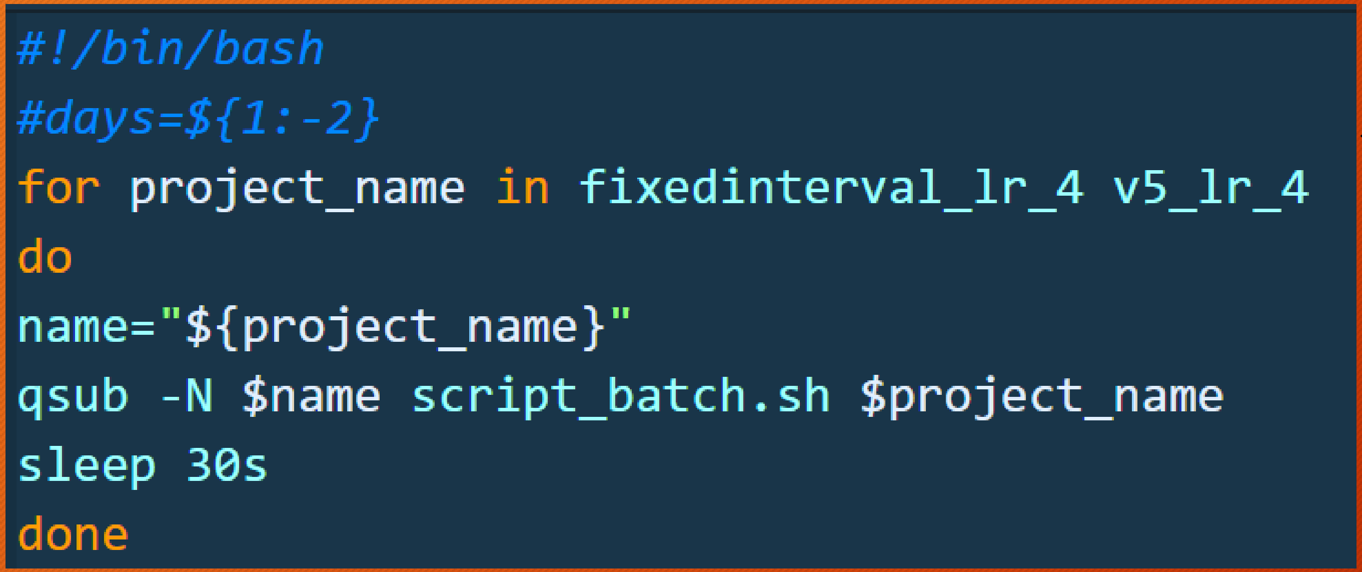


```python
import configparser
```



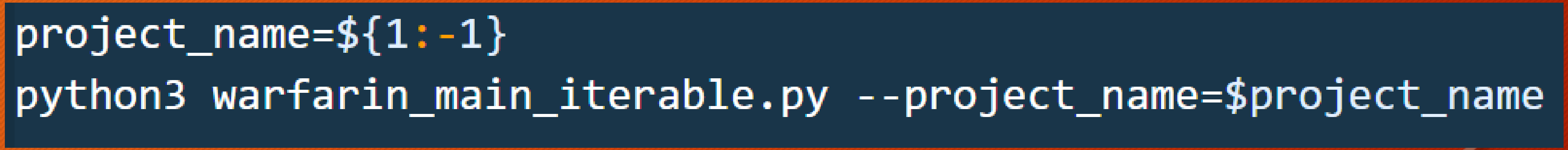
```
[DEFAULT]
subject = WarfarinModel_v5
epochs = 200
training_size = 250
test_size = 50
max_day = 91
dose_history = 10
inr_history = 10
patient_selection = ravvaz
inr_penalty_coef = 1.0
dose_change_penalty_coef = 0.0
dose_change_penalty_func = none
dose_change_penalty_days = 0
lookahead_duration = 0
lookahead_penalty_coef = 0.0
randomized = True
agent_type = DQN
method = backward
gamma = 0.95
learning_rate = 0.001
lr_scheduler = False
buffer_size = 900
batch_size = 50
validation_split = 0.3
hidden_layer_sizes = (32, 32, 32)
clear_buffer = False
initial_phase_duration = 1
max_initial_dose_change = 15
max_day_1_dose = 15
maintenance_day_interval = 1
max_maintenance_dose_change = 15
extended_state = False
save_instances = False
save_runs = False
tf_log = False

[fixedinterval]
subject = fixedinterval
epochs = 200
training_size = 250
test_size = 100
initial_phase_duration = 91
save_instances = True
```

Submit multiple jobs



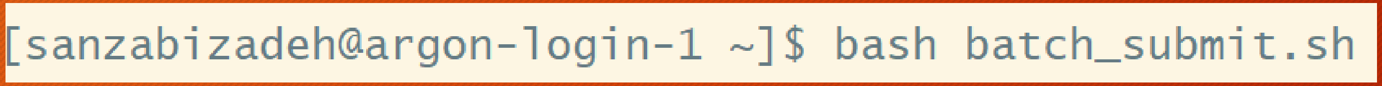
```
#!/bin/bash
#days=${1:-2}
for project_name in fixedinterval_lr_4 v5_lr_4
do
name="${project_name}"
qsub -N $name script_batch.sh $project_name
sleep 30s
done
```

batch_submit.sh



```
project_name=${1:-1}
python3 warfarin_main_iterable.py --project_name=$project_name
```

script_batch.sh



```
[sanzabizadeh@argon-login-1 ~]$ bash batch_submit.sh
```