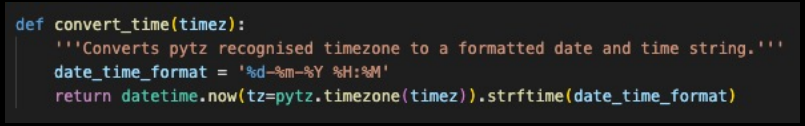
Welcome to the Slide Deck for T1A3 – Terminal Application

Created by Sam Carter

Idea of the Application and Packages Used

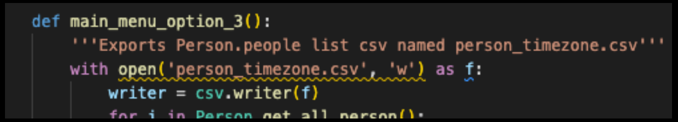
The application is designed to allow users to input people and their time zones into the app and view the current times for one or all people added. Will get into the technical side of the features in slides to come. The modules that are included in the application are:

- Datetime – was used in the process of determining the current time and allowing pytz to convert
- Pytz – this third-party package provided all of the time zones in the app along with the information about the time difference between.

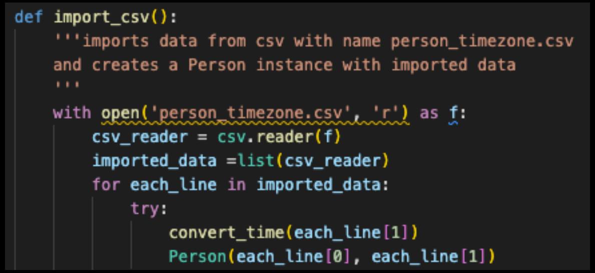


```python
def convert_time(timez):
    '''Converts pytz recognised timezone to a formatted date and time string.'''
    date_time_format = '%d-%m-%Y %H:%M'
    return datetime.now(tz=pytz.timezone(timez)).strftime(date_time_format)
```

- Csv – was used in the process of importing and exporting the stored list of people to a CSV file.



```python
def main_menu_option_3():
    '''Exports Person.people list csv named person_timezone.csv'''
    with open('person_timezone.csv', 'w') as f:
        writer = csv.writer(f)
        for i in Person.get_all_person():
```

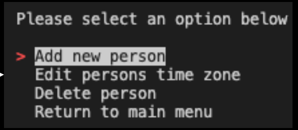


```python
def import_csv():
    '''imports data from csv with name person_timezone.csv
    and creates a Person instance with imported data
    '''
    with open('person_timezone.csv', 'r') as f:
        csv_reader = csv.reader(f)
        imported_data =list(csv_reader)
        for each_line in imported_data:
            try:
                convert_time(each_line[1])
                Person(each_line[0], each_line[1])
```

- Simple Term Menu – was used to create the menus all throughout the app. This was a great help when it came to pre-emptive error handling

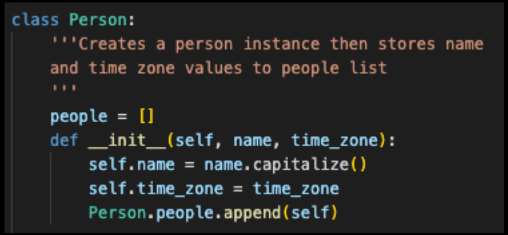
How it Ended Up – Feature 1 – Managing a person

Create, change time zone and delete user

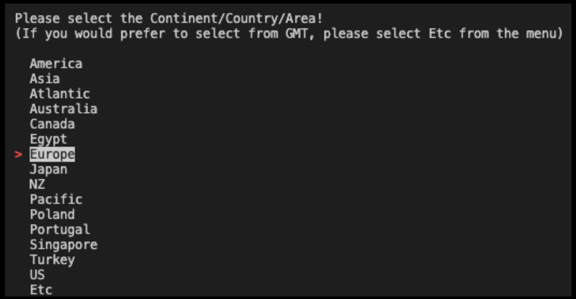


```
Please select an option below

> Add new person
  Edit persons time zone
  Delete person
  Return to main menu
```

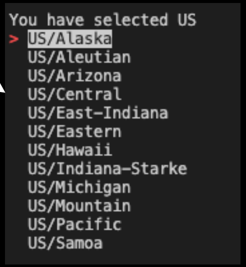
Creates using a class method – Two arguments: name (input) and time zone (Lists created with simple term menu that drill down by continent/country/area.



```python
class Person:
    '''Creates a person instance then stores name
    and time zone values to people list
    '''
    people = []
    def __init__(self, name, time_zone):
        self.name = name.capitalize()
        self.time_zone = time_zone
        Person.people.append(self)
```

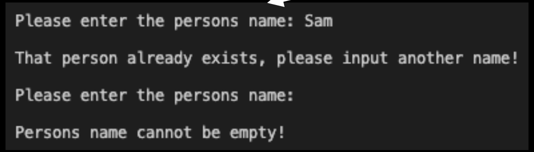


```
Please select the Continent/Country/Area!
(If you would prefer to select from GMT, please select Etc from the menu)

  America
  Asia
  Atlantic
  Australia
  Canada
  Egypt
> Europe
  Japan
  NZ
  Pacific
  Poland
  Portugal
  Singapore
  Turkey
  US
  Etc
```

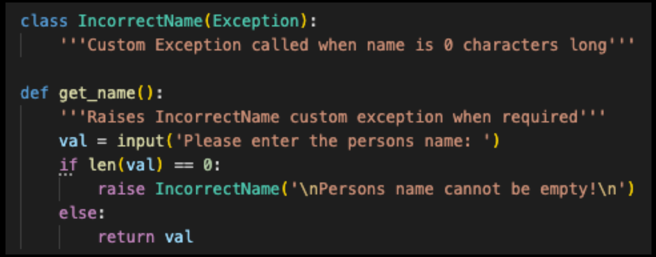


```
You have selected US
> US/Alaska
  US/Aleutian
  US/Arizona
  US/Central
  US/East-Indiana
  US/Eastern
  US/Hawaii
  US/Indiana-Starke
  US/Michigan
  US/Mountain
  US/Pacific
  US/Samoa
```

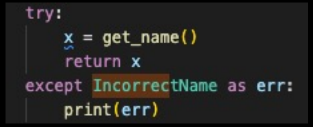
Error Handling – checks for duplicate or empty name inputs and raises custom errors



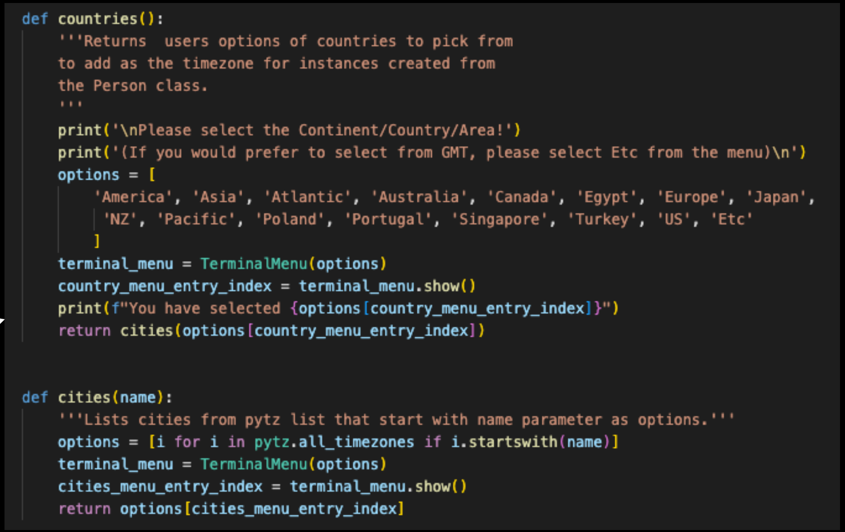
```
Please enter the persons name: Sam

That person already exists, please input another name!

Please enter the persons name:

Persons name cannot be empty!
```



```python
class IncorrectName(Exception):
    '''Custom Exception called when name is 0 characters long'''

def get_name():
    '''Raises IncorrectName custom exception when required'''
    val = input('Please enter the persons name: ')
    if len(val) == 0:
        raise IncorrectName('\nPersons name cannot be empty!\n')
    else:
        return val
```



```python
try:
    x = get_name()
    return x
except IncorrectName as err:
    print(err)
```

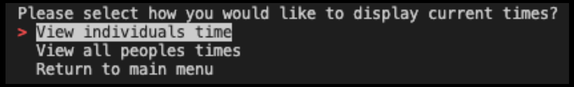


```python
def countries():
    '''Returns  users options of countries to pick from
    to add as the timezone for instances created from
    the Person class.
    '''
    print('\nPlease select the Continent/Country/Area!')
    print('(If you would prefer to select from GMT, please select Etc from the menu)\n')
    options = [
        'America', 'Asia', 'Atlantic', 'Australia', 'Canada', 'Egypt', 'Europe', 'Japan',
        'NZ', 'Pacific', 'Poland', 'Portugal', 'Singapore', 'Turkey', 'US', 'Etc'
        ]
    terminal_menu = TerminalMenu(options)
    country_menu_entry_index = terminal_menu.show()
    print(f"You have selected {options[country_menu_entry_index]}")
    return cities(options[country_menu_entry_index])


def cities(name):
    '''Lists cities from pytz list that start with name parameter as options.'''
    options = [i for i in pytz.all_timezones if i.startswith(name)]
    terminal_menu = TerminalMenu(options)
    cities_menu_entry_index = terminal_menu.show()
    return options[cities_menu_entry_index]
```

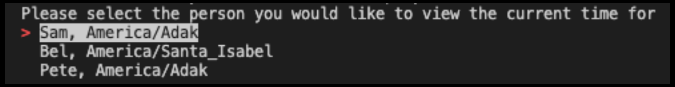
Issues – PYTZ sorting of time zones – not always consistent

How it Ended Up – Feature 2 – Displaying the Times

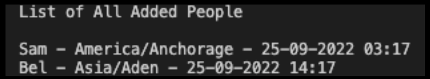
View people from the list either individually or prints all people:



```
Please select how you would like to display current times?
> View individuals time
  View all peoples times
  Return to main menu
```

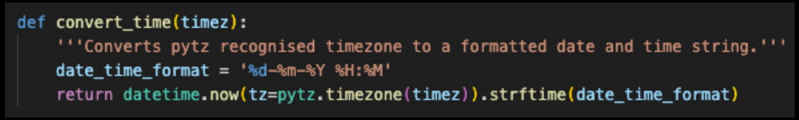


```
Please select the person you would like to view the current time for
> Sam, America/Adak
  Bel, America/Santa_Isabel
  Pete, America/Adak
```

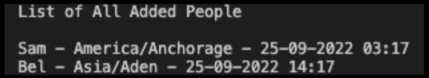


```
List of All Added People

Sam - America/Anchorage - 25-09-2022 03:17
Bel - Asia/Aden - 25-09-2022 14:17
```

This feature also carries out the time zone conversion from a string to a formatted time:

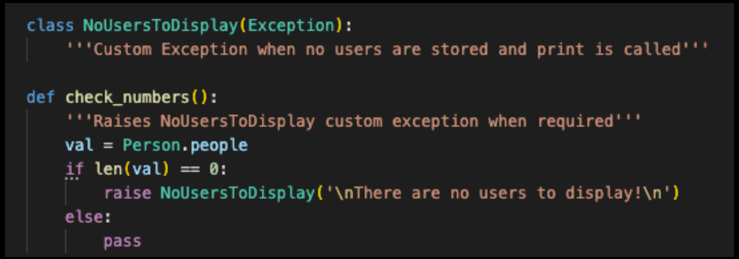


```
def convert_time(timez):
    '''Converts pytz recognised timezone to a formatted date and time string.'''
    date_time_format = '%d-%m-%Y %H:%M'
    return datetime.now(tz=pytz.timezone(timez)).strftime(date_time_format)
```

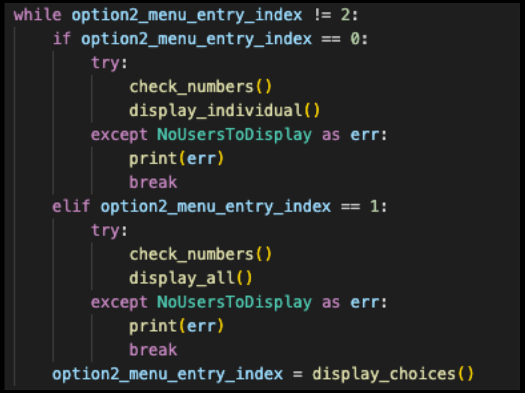


```
List of All Added People

Sam - America/Anchorage - 25-09-2022 03:17
Bel - Asia/Aden - 25-09-2022 14:17
```

Error Handling – checks if there are class instances in the list before displaying:



```
class NoUsersToDisplay(Exception):
    '''Custom Exception when no users are stored and print is called'''

def check_numbers():
    '''Raises NoUsersToDisplay custom exception when required'''
    val = Person.people
    if len(val) == 0:
        raise NoUsersToDisplay('\nThere are no users to display!\n')
    else:
        pass
```

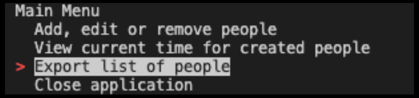


```
while option2_menu_entry_index != 2:
    if option2_menu_entry_index == 0:
        try:
            check_numbers()
            display_individual()
        except NoUsersToDisplay as err:
            print(err)
            break
    elif option2_menu_entry_index == 1:
        try:
            check_numbers()
            display_all()
        except NoUsersToDisplay as err:
            print(err)
            break
    option2_menu_entry_index = display_choices()
```

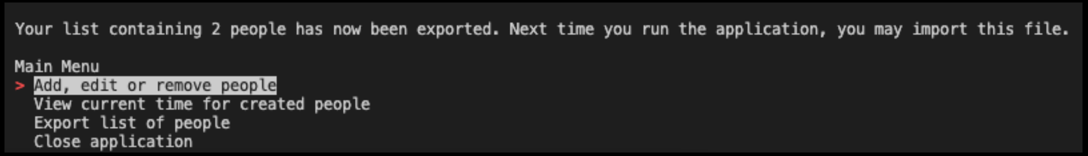
How it Ended Up – Feature 3 – Importing/Exporting CSV

Allows the user to export their list of people to a CSV file which can later be imported when the app next runs

Exporting is done from the main menu:

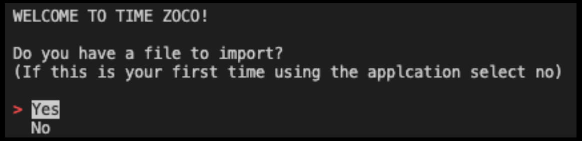


```
Main Menu
  Add, edit or remove people
  View current time for created people
> Export list of people
  Close application
```

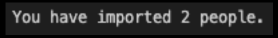


```
Your list containing 2 people has now been exported. Next time you run the application, you may import this file.

Main Menu
> Add, edit or remove people
  View current time for created people
  Export list of people
  Close application
```

While user is given the option to import the CSV file when the app starts:



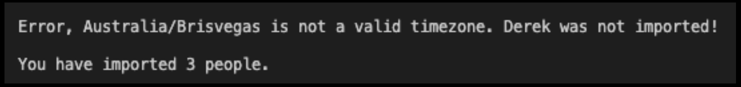
```
WELCOME TO TIME ZOCO!

Do you have a file to import?
(If this is your first time using the applcation select no)

> Yes
  No
```



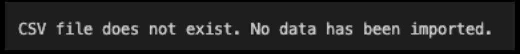
```
You have imported 2 people.
```

Error Handling – When importing files, the time zone data is checked to confirm it has a corresponding time zone in the pytz database. If an error is found, the user will be made aware of the failed import and the app will continue importing the other perople from the CSV

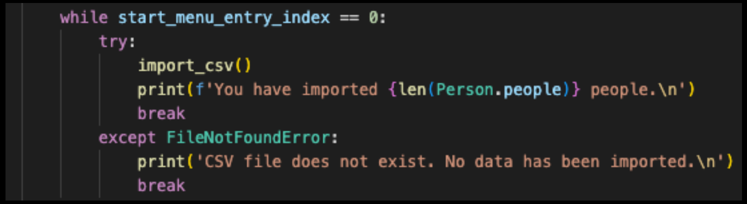
If CSV file does not exist and user tries to import, they will be made aware



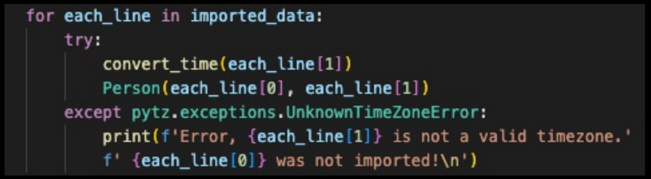
```
Error, Australia/Brisvegas is not a valid timezone. Derek was not imported!

You have imported 3 people.
```



```
CSV file does not exist. No data has been imported.
```



```python
while start_menu_entry_index == 0:
    try:
        import_csv()
        print(f'You have imported {len(Person.people)} people.\n')
        break
    except FileNotFoundError:
        print('CSV file does not exist. No data has been imported.\n')
        break
```



```python
for each_line in imported_data:
    try:
        convert_time(each_line[1])
        Person(each_line[0], each_line[1])
    except pytz.exceptions.UnknownTimeZoneError:
        print(f'Error, {each_line[1]} is not a valid timezone.'
        f' {each_line[0]} was not imported!\n')
```

Final Thoughts

Great learning experience!

Much better understanding of Python syntax and the format.