

Simple to-do app

Write a simple console-based to-do app.

Recommended prior understanding

- Basics (variables and operations, selection, loops, functions)
- Dates
- Files
- Exceptions

Task 1: Creating a task

- Create a function that accepts 2 parameters for the `task_description` and `due_date` of a task.
- Verify that `due_date` has been provided in an acceptable date format
- Verify that `due_date` is indeed in the future
- Save the contents to `tasks.txt`, creating the file if it doesn't exist, or adding to it if it does.

An example of the function header might be...

```
def create_task( due_date, task_description ):
    # ..... create your solution .....
```

An example function call might be...

```
create_task( "29/07/2020", "Mr Baumgarten's birthday" )
```

Which might then create the following lines in a file...

```
29/07/2020,Mr Baumgarten's birthday
```

Task 2: Reading the task list and providing a user interface

When the program first starts up, it should display...

1. Active tasks
2. Dates remaining on each task
3. Provide a menu of options for closing (deleting) a task or creating a new task

An example program, when started up, might present the following...

```
Welcome to my simple to-do app. Current jobs are:  
1. Mr Baumgarten's birthday      Due in 198 days  
2. Math homework                Due in 3 days  
3. Chinese new year holiday      Due in 11 days  
  
To close a task, type the task number and <enter>  
To create a new task, type the task description and <enter>  
To exit, hit <enter> on an empty line  
  
Your choice:
```

You don't have to structure your menu as the above, it is an example.

Task 3: Modifying existing tasks

Provide the mechanism for the user to close/delete a task. When closed, the line containing that task should be removed from the file.

Extension opportunities

- Sort tasks so the one due soonest is listed first, etc
- Functionality to change the due date of a task
- Alarm/alert reminders
- Use Tkinter to give the app a GUI interface

Remember

- Your program must include appropriate prompts for the entry of data.
- Error messages and other output need to be set out clearly.
- All variables, constants and other identifiers must have meaningful names.