

Goals

- User-friendly interface
- Neutral colours and simple geometry
- Basic features:
 - add and remove tasks
- Advanced features:
 - initial tutorial to introduce the main features
 - track and update a task's progress (percentage of completion and hours of work left)
 - categorisation and sorting through urgency - overdue, critical, high, medium, and low
 - calculated with current and due dates
 - revisit completed tasks
 - random selection option with weighted probabilities
 - periodic reminders to maintain engagement
 - email alerts once a task gets moved to the critical category (the latter is not operational, because of some difficulties with the e-mail configurations)

Important missteps and considerations

- Categorising by urgency

The initial idea was to ask the user to define the level of urgency of the task when it was introduced. But in all likelihood, when the user revisits the program at a later date, these levels of urgency would've changed. This problem is fixed by using the datetime library to categorise the task by the number of days from today to the due date.
- Completion and hours remaining

When adding a new task, the user is asked to indicate how many hours it should take to complete. So, once the completion percentage got updated, the program could automatically calculate the number of hours remaining using direct proportionality. HOWEVER, we know that it is sometimes the case that once we start a task it reveals more time-consuming than expected, so I decided to leave these two parameters (completion and hours remaining) independent.
- Layout

Creating a balanced layout was a challenging step. Not only does it have to be aesthetically pleasing, but it also tells the user where to focus their attention. I think keeping the main commands on the top left corner and having the task display area in the middle of the page was the most successful option.
- Reminders

Too many reminders can be overwhelming, and, in the long term, the user becomes desensitised to them. This was definitely a concern and the reason why I decided to add only two types: 1) after five minutes of interaction in the platform, the user is asked whether they know what task to work on, otherwise they are suggested the random selection feature and 2) e-mail alert when a task gets moved to the critical category (<2 days)

Notes

- Please run the attached script on Spyder
- The data inserted on the platform is saved between sessions on a JSON file
- When the program first runs, the user is asked to insert their e-mail which is saved locally, so there are no predicted safety issues.