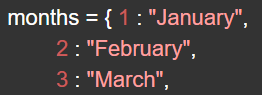
# Python Dictionary Worksheet

## Task 1

Create a months lookup dictionary, allow the user to enter a number, output the month.



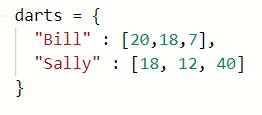
## Task 2

Create a dictionary of fizzy sports drinks, name is the key and calories is the value.

* Print the keys
* Print the values
* Print the key and corresponding value
* Allow the user to enter the name of a drink, print the calories in it.
* Allow the user to display the items sorted alphabetically

## Task 3

Bill and Sally are playing darts. Use a dictionary to store a list of scores for both players. Allow the user to enter the darts scores (Bill, then Sally, then Bill, etc) and show their current score (301 minus what they have scored) and what is required to get to zero. Example dictionary below.



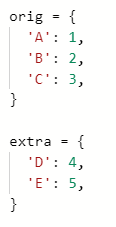
Hint:



Will print ‘10’

## Task 4

Merge the following two dictionaries:



<https://docs.python.org/3/library/stdtypes.html#dict.update>

Write an example that shows what happens when the same key exists in both dictionaries.

## Task 5

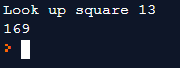
Write a Python script to print a dictionary where the keys are numbers between 1 and 15 (both included) and the values are square of keys.

e.g.



Then allow the user to look up the square value of a number between 0 and 15:





## Task 6

Loop over a dictionary and print out all the keys

Loop over a dictionary and print out all the values

Loop over a dictionary and print out both the keys and the values

## Task 7

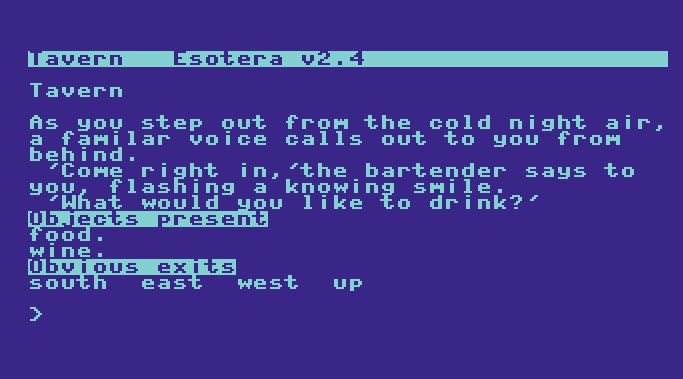
Create a dictionary of dictionaries. E.g:



Add in at least 5 people. Allow the user to choose what to see, e.g. if I select **pets**, loop over the dictionary and display all people and their pets.

## Task 8 - RPG

Create an old school, command line role play game.



Play the game:

<https://replit.com/@JohnGlazebrook/rpg-pick-up-item-SIMPLE#main.py>

Actions are “go village” or “go tower”.

Each location has a number of exits, these are valid routes to other locations. The valid routes match the dictionary key names.

Task:

Implement your own version of a RPG game.

Expand each location dictionary to include items. Some locations will require an item to open a new exit, e.g. for the player to enter the wizard tower, they need to have a key with them. The key could be located in the village.

The player can pick up things from locations, “take key”.

<https://replit.com/@JohnGlazebrook/RPG-take-item#main.py>

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

The code looks like this:

A picture containing graphical user interface

Description automatically generated

When a player picks up an item, the item is removed from **locations -> “village” -> items** list and added to the **player\_items** list:

Text

Description automatically generated with low confidence

Make sure you look at the drop item code, this does the reverse.