

Task 1 – Restaurant menu subroutine

Create a new Python source file, save it as `restaurant.py` and create a subroutine called `menu()`.

It should print a menu with three sections:

- Starters
- Mains
- Desserts

Within each section, print at least 3 dishes.

After the subroutine definition is complete, test it by calling the subroutine.

Task 2 – Call the subroutine multiple times

Develop the restaurant program so it:

1. Displays a message welcoming customers to the restaurant and asking what they would like to order for a starter.
2. Displays the menu.
3. Displays another message asking the customers what they would like to order for their main meal.
4. Displays the menu.
5. Finally, display a message asking the customers if they would like to order dessert.
6. Displays the menu one final time

Your program should only be 6 lines long (not counting the lines you used to create the subroutine itself).

Task 3 – Using a parameter to display only part of the menu

Whilst it is impressive to be able to fill the screen with text with 13 lines of code, it would be more useful if we only displayed part of the menu, not the full menu, at different points of the program. Modify your restaurant program so it is passed a string which determines which part of the menu is displayed.

- `menu("starters")` will only print the starters
- `menu("mains")` will only print the mains
- `menu("desserts")` will only print the desserts

Extension: Usually when you enter a restaurant you want to see the whole menu when you first sit down. Add an option so that `menu("all")` prints the whole menu, and call that instead of “starters”. The other two calls should remain as they are.



Task 4 - Square

Create a subroutine that takes in one number and returns the same number multiplied by itself (i.e. squares the number).

- Store the return value into a variable called result
- Print the contents of the result variable
- Test the program by calling the subroutine with the value 2 (returns 4)
- Test the program by calling the subroutine with the value 5 (returns 25)

Task 5 - Sum

Create a subroutine that takes in two numbers and returns the sum of the two numbers (i.e. adds the two numbers).

- Store the return value into a variable called result
- Print the contents of the result variable
- Test the program by calling the subroutine with the values 14 and 15 (returns 29)
- Test the program by calling the subroutine with the values 89 and 127 (returns 216)

Task 6 – Odd?

Create a subroutine which is passed a number and returns true if the number is odd, or false if it is even. Think especially carefully when deciding on a name for this subroutine.

To make use of your subroutine, add code to do the following:

- Take in a number at the keyboard from the user
- Pass this number to your subroutine
- Depending on the value that is returned from the subroutine, print “is odd” or “is even” as appropriate

Task 7 – Validate password

Create a subroutine which is passed a password string, and returns True if it is valid or False if it is invalid.

To be valid the password must:

- Be at least 8 characters long, but no longer than 12 characters
- Contain at least one number
- Contain at least one of the following symbols: !\$%^&*()