

CS118

Menu Exercises, IF

A “menu” is a method for allowing the user to choose which part(s) of the program s/he wishes to execute. Programs can have a single (“main”) menu, or they can have a series of “cascading” menus – choosing an option from one menu can lead to more options on a separate menu.

There are three parts to apply when applying a menu for a program:

1. Display the menu
2. Obtain a choice from the user
3. Process the choice provided

1. Display the menu

Menu options typically offer either single letters or numbers that will be used as input values by the user, followed by a description of the option. There may or may not be a descriptive phrase before the actual menu options.

Example:

```
Please choose from this menu:

1.    Do this
2.    Do that
3.    Do something else
4.    Exit the program
```

Implemented in Python3 as a series of `print()` statements.

2. Obtain a choice from the user

This is a simple input from the user. The data type of the input should match the data type of the input values (strings for letter values, numbers for numeric values).

Example:

```
Your choice (1-4)?
```

Implemented in Python3 as:

```
choice = int(input("Your choice (1-4)? "))
```

3. Process the choice provided

Since the purpose of a menu is to allow the user to do different things with the program, the menu processing will always be a conditional.

Implemented using `IF` in Python3:

```
if choice == 1:
    print("OK, I'm doing *this*.")
elif choice == 2:
    print("This time I'm doing *that*.")
elif choice == 3:
    print("Fine! I'll do something else!")
elif choice == 4:
    pass # Nothing to do in this elif
else:
    print("That is not one of the menu options!")
```

1. Prepare a Python3 program that will provide and process the menu as shown on the previous page, using the `IF` conditional.
2. Copy the program from #1 and extend the processing actions for options 1-3 so that each uses two `print()` statements.
3. Copy the program from #2 and change the input values shown in the menu to the letters "A", "B", "C", and "X". For the program to run, you'll also need to change the `input()` line to expect a string; and change the `if` and `elif` conditions to use "A", "B", "C", and "X". Note that these letter values are *string literals* and **must** be delimited by double-quotes.
- 4a. An error will occur if the double-quote delimiters are omitted in #3. Answer this question with a comment in a .PY file:
Why does this happen (i.e. how does Python3 interpret the code without the delimiters)?

It is possible that more than one input value from the menu will need to perform the same actions. For example:

```
Enter a year:
1990-1999:      The 90s
2000-2009:      The "noughts"
2010-2019:      The teens

What year (1990-2019)?:
```

When implementing multiple options which will have the same actions in Python3, the `if` and `elif` conditions are handled through logical OR conjunctions (i.e. `or`). Alternatively, the `<`, `<=`, `>`, `>=` relational operators can be used if applied appropriately. In the following, the `IF` uses the (very long-winded) logical OR and the `elif`s use the relational operators:

```
if year==1990 or year==1991 or year==1992 or year==1993 or year==1994 or year==1995 or year==1996 or year==1997 or year==1998 or year==1999:
    print("You are about to enter the 90s!")
elif 2000 <= year <= 2009:
    print("You are about to enter the noughts!")
elif year >= 2010 and year <= 2019:
    print("You are about to enter the teens!")
else:
    print("That wasn't an option!\n")
```

Note that you must not shortcut the `or` operator. The following would not function correctly:

```
if year==1990 or 1991 or 1992 or 1993 or 1994 or 1995 or 1996 or 1997 or 1998 or 1999:
    print("You are about to enter the 90s!")
```

However, the interpreter DOES understand this common mathematical expression for a condition:

```
elif 2000 <= year <= 2009:
    print("You are about to enter the noughts!")
```

Which is equivalent to this condition:

```
elif year>=2010 and year <= 2019:
    print("You are about to enter the teens!")

else:
    print("That wasn't an option!\n\n")
```

- 4b. Prepare a program where the value for year is obtained as an integer from the user – prompt the user to provide a number between 1990 and 2019, inclusive. Use the code **in red** above to process the user's input. Run the program with an input of 2005 – if written as shown, the program will produce the output "You are about to enter the 90s!". Capture a screenshot of the Spyder window with the console demonstrating this inappropriate output. Correct the code to be like the code in black and capture another screenshot. **Instead of submitting those two programs, include both screenshots in your exercise submission ZIP file.**

5. A similar processing as done with numbers can be done with string item values. Copy the program from #3 and have the same action be performed whether an uppercase or lowercase letter is chosen. Don't keep two separate conditions; and don't use any string-related functions: use one condition with a logical OR. **Be sure you do not shortcut the logical operator!**