

Exercises

1. Password verification

Use the input function to get a password, and then ask the user to retype his password. Check if the first and second password are the same or not and give feedback for each case.

2. Age verification

Ask the user to enter his age. Check if the provided age falls within a predetermined range (you can choose your own values). Give feedback for when this is the case and for when it isn't.

3. A login system

Create two variables `VALID_USERNAME` and `VALID_PASSWORD` that you assign a string value of your choosing. Then, ask a user to enter his username and password. If both these inputs match the ones you set, tell the user that he has successfully logged in.

4. Condition assignment

In an experiment, subjects get assigned a numerical id. To map these ids on the condition that a participant is in, the experimenter has come up with a rule that all ids divisible by 4 belong to group A. All the other ids are in group B.

Write a program that gets the id and tells the user which condition he/she belongs to.

5. Breathalyzer test

Write a program to help the police conduct their breathalyzer tests. The program receives the Blood Alcohol Concentration (BAC) level and decides which state the subject is in. You can use the following values:

Safe: BAC is smaller than 0.02

Warning: BAC is between 0.02 and 0.08

Danger: BAC is above 0.08

6. A little casino

Create a program that allows a user to guess the outcome of a die roll. If the user guesses correctly, he gains some money.

To simulate a die roll, you can use a random number generator as in the example below:

```
1. # Include this import statement at the top of your script
2. import random
3.
4. # Use to generate a random number in the interval [1, 10]
5. n = random.randint(1,10)
```

7. Bug counter

Create a variable `bug_count` and set its value to 5. Then, use a while loop that prints the

following text, where xx should be replaced with the value in the bug count variable. Keep repeating this until all bugs are solved!

```
xx little bugs in the code  
xx little bugs in the code  
take one down, patch it around
```

8. *SpaceX Dragon 2 countdown timer*

Create a program that counts down from 10 to zero, pausing for one second between each output. To allow your program to temporarily pause, first include an `'import time'` statement at the top of your script. Then, wherever in your code you want to pause the script, write `time.sleep(x)`, where x is a number representing the time you want to pause in seconds.

9. *An advanced gambling machine*

Create a gambling program that asks the user if he wants to `roll` or `stop` the program. If the user chooses to roll, the program will roll two dice (see exercise 6) and report "Jackpot!" when the user rolls a pair. Keep repeating until the user says he wants to stop.

10. *A vending machine*

Create a program for a vending machine. The program will present the options that are available (let's say 1 for a coke, 2 for a diet coke and 3 for a coffee). The program keeps presenting the options until a valid input (in this example, either 1, 2 or 3), is given

11. *Fizz Buzz*

Create a program that prints the numbers 1 to `MAX_NUMBER`, where the latter is a variable that you assign a value in your script.

In addition to the number itself, check the value of the number to print a specific word if one of the following conditions apply:

If the number is divisible by 3: print Fizz

If the number is divisible by 5: print Buzz

If the number is divisible by both 3 and 5: print FizzBuzz