Activity: loop over different iterables

In a script called **A12-loop1.py** do the following

a. Make a for loop that iterates over characters of **string s**, and print each character lowercase

```
s="PYTHON MONTY PYTHON"
```

b. Make a for loop that iterates over elements of **list veggie** , and print each element uppercase

```
veggie=['spinach', 'broccoli','edamame', 'bell pepper']
```

c. Use a dictionary method to loop over both keys and values of dzoo and obtain this output – the output is not formatted

```
5 pangolin
3 sloth
2 tiger
10 turtle
```

```
dzoo= {"pangolin" : 5, "sloth" : 3, "tiger" : 2, "turtle" : 10}
```

Activity: loop over multiple sequences

In a script called **A12-loop2.py** do the following:

a. Make these variables:

```
veggie=['spinach', 'broccoli', 'edamame', 'bell pepper']
num=[30,15,25,11]
```

You should produce this output:

```
30    spinach
15    broccoli
25    edamame
11    bell pepper
```

in the 3 following ways:

b. one for loop over range(len())

c. one for loop over enumerate()

d. one for loop over zip()

Activity: If-statements

Make a file called **A12-isc.py** and in it:

- Include one input function with this question: "Enjoying this course? yes or no? " Save the user’s input in a variable called **answer**

- Include an **if-elif-else** block that does the following:
  - ✓ if the user answered yes print a statement that says "Good!"
  - ✓ if the user answered no print a statement that says "Sorry!"
  - ✓ if the user answered neither yes nor no, print a statement saying "You didn't pick yes or no! Try again. "

Run the script

Activity: while loop to error-check the user’s input

A common application of a while loop is to error-check the user’s input, i.e., check user input to see if it is valid.

For example, if you ask the user to enter either yes or no and they instead enter 7, then you should re-prompt them for input.

In a script **A12-while1.py**

Part of the code is reported below. You talk is to complete the code by filling in _____

```
while ____:
     answer=input('Enjoying this course? yes or no ')
     if __________:
         break
```

Activity: summing loop

In a script called A12-patients.py do the following

a. Define this dictionary d1, which contains information about height, weight and blood pressure of some patients.
The dictionary structure is the following

{name: [height, weight, blood pressure] }

height is in cm, weight is in kg, and blood pressure is in millimeters of mercury (mmHg)

```
d1={'maria':[168,65,122], 'giovanna':[179,105,110],
'carlo':[190,95,130], 'francesco':[176,100,145]}
```

b. Calculate the average blood pressure and print the result to screen. Use one for loop to calculate the sum. In this exercise, the sum must be calculated within the repeated task of the for loop. Use a function to then calculate the average.

c. Use one for loop to create a string that contains the names of the patients ending with character a. The string should be `mariagiovanna`
You should create the string in the for loop, by using a test condition and concatenation.
Print the final string to screen.

Activity - random module

In a script called **A12-random.py** do the following:

a. Import the random module

b. Generate an integer random number in range 1-10 and print it to screen.

c. Generate a real random number in range [0,1) and print it to screen

d. Define this string

```
s="abracadabra"
```

Randomly select a character from string s, and print it to screen

e. Define this list

```
L=['yellow','blue','orange','pink']
```

Randomly select an item from list L, and print it to screen

f. Randomly shuffle L, and print the modified list to screen

Activity: Nested loops

In a script called **A12-nested-lists.py** do the following

a. Define this list of lists

```
L=[[1,2,3,4],[5,6,7],[7,8,9,10]]
```

b. Loop over elements of list L and calculate the sum of all the numbers in the list. Print the final result to screen. Do it in two ways:

b1. by using one for loop

b2. by using a nested loop

Your loops should work on an arbitrary list of lists - do not hardcode.

Activity: more summing loop

In a script called **A12-sum-conc.py** do the following:

a. Define dictionary **dsquare**, where the keys are strings naming the squares, and the associated values are the sides of the squares in meter.

```
dsquare= {"square1" : 5, "square2" : 3, "square3" : 2,
"square4" : 10}
```

b. Loop over the keys of dsquare and build this string: `'re1re2re3re4'`
inside the for loop. You should select the last 3 characters of each key by using slicing and concatenate them within the loop – it is like a summing loop structure, but in this case the + will concatenate

c. Calculate the total area of the squares (you need to square each value and add the squares). Do not hardcode. Use a summing loop structure.

Submit to A12:

- **A12-loop1.py**
- **A12-loop2.py**
- **A12-isc.py**
- **A12-while1.py**
- **A12-patients.py**

- **Optional**
- **A12-random.py**
- **A12-nested-lists.py**
- **A12-sum-conc.py**