



Adamson University  
College of Engineering  
Computer Engineering Department



Linear Algebra

Laboratory Activity No. 1

---

# Getting acquainted with Python

---

*Coded and submitted by:*

Vallarta, Troy Joaquin G.

CPE 201

*Instructor:*

Engr. Dylan Josh D. Lopez

September 23, 2020

---

## I. Objectives

This laboratory activity aims to implement the principles and techniques of python, especially list and list comprehension.

## II. Methods

- To be acquainted with python ○ It teaches the student to learn the difference of python in other programming languages
- To provide adept understanding in python especially about lists ○ How did you achieve them?

## III. Results

The first activity is the best way to be familiarized with python and how the list works in this language. Two lists need to extract one variable and pair it to the other variable of the list. The programmer's experience in this exercise is that the programmer stumbled upon the zip function. What it does is pairing the two lists shown below. With this knowledge, It can pair up with other lists.

The second exercise is a very challenging exercise to make it in one line. It needs to have an understanding of list comprehension, to make it in one line. As seen in Figure 2, it sorts the highest or the lowest level of a certain Pokemon. The programmer has no idea where to start while reading exercise two, then the programmer researched list comprehension. He read about that and suddenly, eureka moment! The programmer can do it in one line by using list comprehension and getting the key to the sorted levels. Using the new walrus operator that is new to many python language users, that means that the programmer dived onto new learning. The walrus operator assigns a value and also returns it.

In the last exercise, it seems to be easy to do but very hard to make it in one line. Using every bit of knowledge in programming and still, I can't do it in one line. Therefore, the programmer settled not doing many lines. What the exercise is all about is that It needs to copy the list of the party list in Figure 1 and appending one Pokemon in the party. An intriguing puzzle that will make programmers think for days. A fun exercise to get accounted with the Python program

```
In [1]: party = ['Charmander', 'Pidgey', 'Sandshrew', 'Rattata', 'Abra']
levels = [15, 11, 18, 5, 14]
for i,j in zip(party,levels): print(f'{i} at leve {j}')
# Using for loop to print all the variable in the list
# the zip() is where it pairs the iterator. Just like in the program it pairs the first then the next.

Charmander at leve 15
Pidgey at leve 11
Sandshrew at leve 18
Rattata at leve 5
Abra at leve 14
```

Figure 1 Exercise#1

```
In [2]: reserves = [
    ('Onix',10),
    ('Slowpoke',18),
    ('Dialga', 2),
    ('Magikarp', 32),
    ('Feebas', 22),
    ('Swablu', 19),
    ('Regigigas', 3),
    ('Unown', 50)
]

(candidates := [k[0] for k in sorted(reserves, key =lambda x:x[1])][0:3])
#using the walrus operator to have a value of the candidates
#I've used the list comprehension to make the list and sort the key as its level

Out[2]: ['Dialga', 'Regigigas', 'Onix']
```

Figure 2 Exercise#2

```
In [30]: def create_party(party, candidates):
    suggested_parties = party.copy()
    for l in range(3):suggested_parties.append(candidates[l]);print(suggested_parties);suggested_parties.pop()
    return suggested_parties
create_party(party, candidates)
#please sir dylan give me some hint how to make this one liner

['Charmander', 'Pidgey', 'Sandshrew', 'Rattata', 'Abra', 'Dialga']
['Charmander', 'Pidgey', 'Sandshrew', 'Rattata', 'Abra', 'Regigigas']
['Charmander', 'Pidgey', 'Sandshrew', 'Rattata', 'Abra', 'Onix']
```

Figure 3 Exercise#3

## IV. Conclusion

In this Laboratory Activity, I've learned a lot in terms of programming. I've become more comfortable in using python and learning its nuances. For example, the use of lists is very helpful and informative. Lists give many variables and the programmer can modify it the way they want to achieve. In this laboratory activity, it taught us a lot of lessons in programming.

1.In your perspective, what is the difference between Python and C++ or other languages you have used before? (30 words)

In my perspective, the difference between c++ and Python is that Python uses more logical operators like and, not, or, and many more. Python is much easier to read codes

2. Enumerate and briefly discuss the functions you have used in the laboratory exercise, please cite their usage using their respective documentations. (200 words)

For loop [Fig. 1 & 3], execute the code until it satisfies then stops. Zip [Fig. 1], use to pair up lists. List comprehension [Fig 2], it defines one list into another list. Sorted [Fig. 2], to sort in values in ascending order if the programmer wants to sort in descending order reverse = True inside the sorted function. key in sorted [Fig. 2], this defines the key was to sort. lambda [Fig. 2], the use of lambda is being an anonymous function with expression right after the colon. in [Fig. 1, 2, & 3] "in" are used in loops for filtering what is inside your variable. append [Fig. 3] the append adds another value to a list. Copy [Fig. 3], a copy is used when you want to duplicate a list. Pop [Fig. 3] is when you want to want to remove the last action that you did, its property is from the stack. Wherein its abbreviation is FILO, First in Last Out. walrus operator [Fig. 2] what it does is that it declares and returns the value, its main uses are shortening the coding lines, making it into one line. This is what I've used in the Laboratory report.

3. In your perspective, what are the advantages and disadvantages of using Python and Jupyter Notebooks? (100 words)

The advantage of Jupyter in my opinion is the design of the notebook is easier to understand unlike the other python IDE's. For me, this is also more efficient in reading codes, it separates the cells to understand if efficiently The last thing is that if your code is jumbled and running in different cells, Jupyter can move the cell up or down. This is my only problem in Jupyter, when I restart the kernel I need to run every single cell to call the variable in a different cell. That is a hassle if I'm working with many cells.