

## ▼ HW 1a Python (Total Points - 5)

You have to submit two files for this part of the HW

1. FirstNameLastName\_Hw1a.ipynb (colab notebook)
2. FirstNameLastName\_Hw1a.pdf pdf file\*\*

- This is a HW to test your understanding of Python Basics.

### ▼ Q1: split() function - 0.25 Point

Split the sentence below into list of words (1 Point)

```
sentence = 'This is a basic python assignment'
```

```
sentence.split()
```

```
['This', 'is', 'a', 'basic', 'python', 'assignment']
```

```
# Do not RUN/MODIFY this cell. This tells you how your solution should look like
```

```
['This', 'is', 'a', 'basic', 'python', 'assignment']
```

### ▼ Q2 print using f-string literals - 0.25 Point

Given the variables:

```
course = 'Machine Learning'
```

```
lec_num = 1
```

Print the sentence

```
This is lecture 1 of Machine Learning
```

```
course = "Machine Learning"
```

```
lec_num = 1
```

```
print(f'This is lecture {lec_num} of {course}')
```

```
This is lecture 1 of Machine Learning
```

```
# Do not RUN/MODIFY this cell. This tells you how your solution should look like
```

```
This is lecture 1 of Machine Learning
```

### ▼ Q3 Indexing for nested list - 1.25 Points

Use indexing to get only the word `python` from nested list

```
nested_list = [10,9,[4,'hi'],['good',[10,99,['python is easy', 'c++ is hard']],45,15],12,107]
```

```
nested_list[3][1][2][0][0:6]
```

```
'python'
```

```
# Do not RUN/MODIFY this cell. This tells you how your solution should look like
```

```
'python'
```

#### ▼ Q4 Update Dictionary - 0.25 Point

In the dictionary below, change the course number of ML course from 6341 to 6341.ss1

```
course_dict = {'Course Name': ['NLP', 'ML', 'DL'], 'Course Number': ['6342', '6341', '6382']}
```

```
course_dict
```

```
{'Course Name': ['NLP', 'ML', 'DL'], 'Course Number': ['6342', '6341', '6382']}
```

```
ML_index = course_dict['Course Name'].index('ML')
```

```
course_dict['Course Number'][ML_index]='6341.ss1'
```

```
course_dict
```

```
{'Course Name': ['NLP', 'ML', 'DL'],  
 'Course Number': ['6342', '6341.ss1', '6382']}
```

# Do not RUN/MODIFY this cell. This tells you how your solution should look like

```
{'Course Name': ['NLP', 'ML', 'DL'],  
 'Course Number': ['6342', '6341.ss1', '6382']}
```

#### ▼ Q5 List Comprehension and Function - 1.5 Points

Create a function that get all the occurrences of the word python (account for capitalization) in a string. You are not allowed to use for loop. (Hint - Use list comprehension. Search how you can use if statement in list comprehension.)

```
def list_python(input_string):
```

```
    input_lower = input_string.lower()
```

```
    Target = "python"
```

```
    occurrences = [input_string[a:a+len(Target)] for a in range(len(input_lower)-len(Target)+1) if input_lower[a:a+len(Target)] ==  
    return occurrences
```

```
    print(occurrences)
```

```
# string to check your function
```

```
input_string = 'Python is wasy. I can learn python in one month'
```

```
# Run this cell and make sure that your solution matches teh solution shown below
```

```
list_python(input_string)
```

```
['Python', 'python']
```

# Do not RUN/MODIFY this cell. This tells you how your solution should look like

```
['python', 'python']
```

#### ▼ Q6 Function and If-elif-else block - 1.5 Points

You all are Pilots. You want to land a plane safely and the altitude required for landing a plane is 1000ft. If it is less than 1000 ft., tell(print) the pilot 'Safe to Land'. If its more than 1000 ft but less than 5000ft, tell (print) the pilot to "Bring down to 1000ft", else if it more than 5000ft tell (print) the pilot to "Go around and try later". The altitude is provided to you by the user (already added as the first line in function).

```
def land_or_not():
```

```
    altitude = int(input("Enter the altitude: "))
```

```
    while True:
```

```
        if altitude < 1000:
```

```
            print(f'Safe to Land')
```

```
        elif 1000 <= altitude < 5000:
```

```
            print(f'Bring down to 1000ft')
```

```
        elif altitude >= 5000:
```

```
            print(f'Go around and try later')
```

```
    break
```

```
# Run this cell to check your solution. Try different input values
land_or_not()
```

```
Enter the altitude: 100
Safe to Land
```

```
# Run this cell to check your solution. Try different input values
land_or_not()
```

```
Enter the altitude: 1200
Bring down to 1000ft
```

```
# Run this cell to check your solution. Try different input values
land_or_not()
```

```
Enter the altitude: 30000
Go around and try later
```

```
# Do not RUN/MODIFY this cell. This tells you how your solution should look like.
land_or_not()
```

```
Enter the altitude: 100
Safe to land
```

```
# Do not RUN/MODIFY this cell. This tells you how your solution should look like.
land_or_not()
```

```
Enter the altitude: 1200
Bring down to 1000 ft!
```

```
# Do not RUN/MODIFY this cell. This tells you how your solution should look like.
land_or_not()
```

```
Enter the altitude: 30000
Go around and try later
```