

DAILY ASSESSMENT REPORT

Date:	24 July 2020	Name:	Gagan M K
Course:	How to develop Pythonic coding – Logic Perspective	USN:	4AL17EC032
Topic:	<ul style="list-style-type: none"> Day 4 	Semester & Section:	6 th sem & 'A' sec
GitHub Repository:	Alvas-education-foundation/Gagan-Git		

FORENOON SESSION DETAILS

Image of session

The screenshot shows a Google Meet interface during a presentation. The main window displays a Google Colab notebook titled "Pythonic workshop Day 4 Session 1.ipynb". The notebook content includes a comment "The following code is harmful" followed by a code block:

```
[ ] book_info = ' The Three Musketeers: Alexandre Dumas'
formatted_book_info = book_info.strip()
formatted_book_info = formatted_book_info.upper()
formatted_book_info = formatted_book_info.replace(':', ' by')
formatted_book_info
```

Below the code block, the output is displayed: "MUSKETEERS by ALEXANDRE DUMAS'". A notification at the bottom left states "Princia Dsouza has left the meeting". The right sidebar shows a grid of participants, including Badhusha Mohideen, Khazi Moin, Persis, Ritika kulkarni_4AL19CS..., Jagadeesha Hegde, anusha k, Ashmitha Mendon, and RASHMITHA POOJARY. The top of the interface shows the presenter "Badhusha Mohideen is presenting" and the time "11:21".

Report – Report can be typed or hand written for up to two pages.

Python:

Writing Pythonic code

The following code is harmful

```
result_list = ['True', 'False', 'File not found']
result_string = ''
for result in result_list:
    result_string += result
result_string
```

```
TrueFalseFile not found
```

Using join function as per PEP 8 rule

Saved successfully!

The following code is idiomatic

```
result_list = ['True', 'False', 'File not found']
result_string = ''.join(result_list)
result_string
```

```
TrueFalseFile not found
```

Chain string functions

The following code is harmful

```
book_info = ' The Three Musketeers: Alexandre Dumas'
formatted_book_info = book_info.strip()
formatted_book_info = formatted_book_info.upper()
```

<https://colab.research.google.com/drive/1bg0z-Z3VfgyGYGUoJHel0dD9eAibbo4H#scrollTo=yL3ufitqJtx6&printMode=true>

1/9

24/07/2020

Pythonic workshop Day 4 Session 1.ipynb - Colaboratory

```
formatted_book_info = formatted_book_info.replace(':', ' by')
formatted_book_info
```

```
THE THREE MUSKETEERS by ALEXANDRE DUMAS
```

The following code is idiomatic

```
book_info = ' The Three Musketeers: Alexandre Dumas'
formatted_book_info = book_info.strip().upper().replace(':', ' by')
formatted_book_info
```

```
THE THREE MUSKETEERS by ALEXANDRE DUMAS
```

Removing Duplicates from a List

Saved successfully!

```
ints_list = [1, 2, 3, 4, 3, 2]
temp = []
for x in ints_list:
    if x not in temp:
        temp.append(x)
ints_list = temp
print(f'Updated List after removing duplicates = {temp}')
```

```
Updated List after removing duplicates = [1, 2, 3, 4]
```

▼ Map() function with lamda in Python

It is used to call the specified function for each item of an iterable (such as string, list, tuple or dictionary) and returns a list of results.

```
def square(x):
    return x*x

numbers=[1, 2, 3, 4, 5]
sqrList=map(square, numbers)
print(list(sqrList))

# use lamda function to have direct values
2, 3, 4])
[1, 4, 9, 16, 25]
[1, 4, 9, 16]
```

In the above example, the map() function applies to each element in the numbers[] list.

▼ Slicing operations in string

Program for exchanging first and last characters of a string using function

```
def change(string):
    return string[-1] + string[1:-1] + string[0]
string=input("Enter string:")
print("Modified string:",change(string))
print('original string :',string)
```

<https://colab.research.google.com/drive/1bg0zZ3VfyGYGUoLHlElOtD9sAibb04H#scrollTo=yL3uffitqJtx&printMode=true>

4/9

24/07/2020

Pythonic workshop Day 4 Session 1.ipynb - Colaboratory

```
Enter string:vinay
Modified string: yinav
original string : vinay
```

▼ Removing one character from a string

```
def remove(string, n):
    first = string[:n]
    last = string[n+1:]
    return first + last
string=input("Enter the string:")
n=int(input("Enter the index of the character to remove:"))
```

Saved successfully!

```
Enter the string:ramajayam
Enter the index of the character to remove:4
Modified string:
ramaayam
```

▼ Sorting in Python using function

```
def myFunc(e):
    return len(e)
#myfunc=lambda x : len(x)
cars = ['Ford', 'Mitsubishi', 'BMW', 'VW']
cars.sort(key=myFunc)
```

- A common neologism in the Python community is pythonic, which can have a wide range of meanings related to program style.
- To say that code is pythonic is to say that it uses Python idioms well, that it is natural or shows fluency in the language. Likewise, to say of an interface or language feature that it is pythonic is to say that it works well with Python idioms, that its use meshes well with the rest of the language.
- Python scripts can put the system into different states, set configurations, and test all sorts of real-world use cases. Python can also be used to receive embedded system data that can be stored for analysis.
- Programmers can then use Python to develop parameters and other methods of analyzing that data. There are certain things you can do with all sequence types.
- These operations include indexing, slicing, adding, multiplying, and checking for membership. In addition, Python has built-in functions for finding the length of a sequence and for finding its largest and smallest elements.
- One of the special concepts in Python is the idea of writing idiomatic code that is most aligned with the language features and ideals. In Python, we call this idiomatic code Pythonic.
- While this idea is easy to understand, it turns out to be fairly hard to make concrete. This course will take you on a tour of over 50 of the more popular and useful code examples demonstrating examples of Pythonic code.
- In the examples, you'll first see non-Pythonic code and then the more natural Pythonic version. Topics covered include the expansive use of dictionaries, hacking Python's memory usage via slots, using generators, comprehensions, and generator expressions, creating subsets of collections via slices (all the way to the database) and more.
- Several of these are Python 3 features so you'll have even more reason to adopt Python 3 for your next project.

Certificate:

