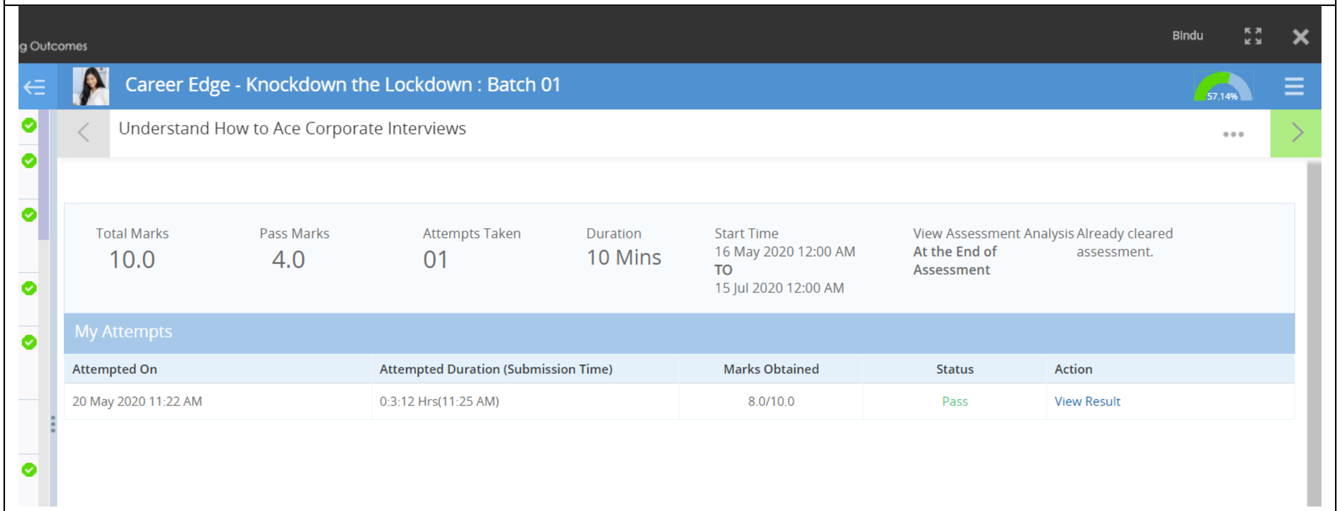
DAILY ASSESSMENT FORMAT

Date:	20-05-2020	Name:	BINDUSHRI
Course:	TCSion	USN:	4AL17EC011
Topic:	Ace corporate interviews Learn corporate Etiquette Write effective emails	Semester & Section:	6th A
Github Repository:	Bindushri		

FORENOON SESSION DETAILS



Career Edge - Knockdown the Lockdown : Batch 01

Understand How to Ace Corporate Interviews

Total Marks 10.0

Pass Marks 4.0

Attempts Taken 01

Duration 10 Mins

Start Time 16 May 2020 12:00 AM TO 15 Jul 2020 12:00 AM

View Assessment Analysis At the End of Assessment

Already cleared assessment.

My Attempts

Attempted On	Attempted Duration (Submission Time)	Marks Obtained	Status	Action
20 May 2020 11:22 AM	0:3:12 Hrs(11:25 AM)	8.0/10.0	Pass	View Result

20-05-2020 - May

Day7:

1) Ace ~~cop~~(co)rporate Interviews

Objectives :- explain the importance of an interview
Imbibe the skills and expertise an interview requires
Identify the engagement rules of a face to face interview
Know how to answer the interview questions effectively
List Do's & Donts of an effective interview

* Interview is a widely used process of screening applicants for jobs.

~~Learns~~

* Interview preparation includes : prepare, practice, present & participate.

* Preparing for a job interview include
  - good assesment of yourself
  - updating your resume
  - Researching the organization
  - understanding the venue details

Summary:-

* Be thoroughly prepared.
* Importance of interview
* Skills & expertise an interview requires
* Do's & Donts of an effective interview

2) Day8 - Learn Corporate Etiquette.

objective :

Importance of Business
Basic rules of Business
To follow cubicle Etiquette

Business Etiquette can be defined as a all the rules that one has to follow when in a Business environment

Language used

- using foul language is not appriciated
- Use golden words "please", "Thank you",
- ~~volume~~

volume

- keep voice low but audible.

Summary :-

- Business Etiquette is essential to build healthy professional relationships
- Basic rules should be followed

Day-9 write Effective emails

Email is the short form for Electronic Mail. It is the information sent electronically between two or more people over a network. It involves a sender and receiver/s

Structure :- 1) Address field
2) Subject
3) Salutation
4) Type the text
5) complimentary close
6) Signature.

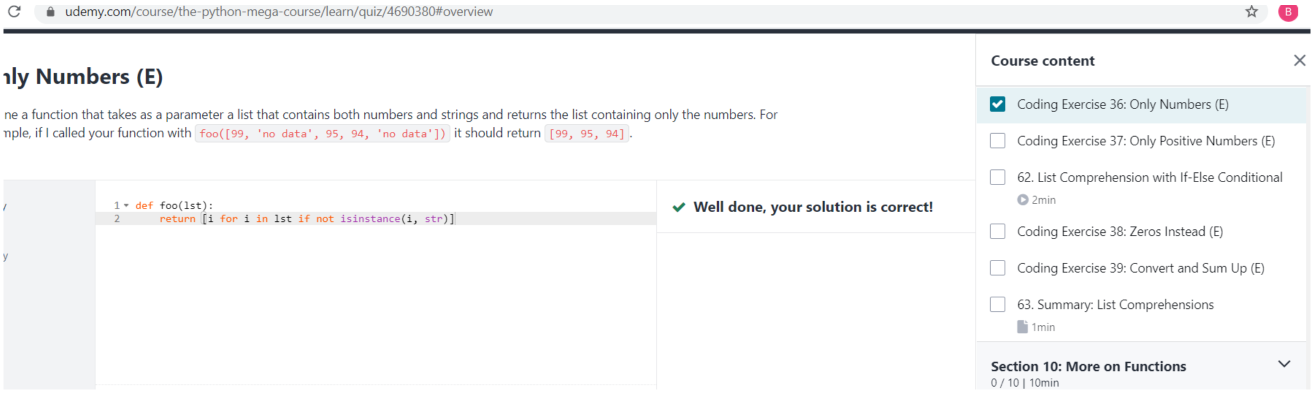
Summary

to write an email using the structure of mail
to write an effective subject line & text
to use correct spellings and grammar

Date:20may2020	**Name:Bindushri**
Course: python	**USN:4AL17EC011**
Topic:	**Sem&Sec:6th A**
Basics:sec-13	

AFTERNOON SESSION DETAILS

Image of session



udemy.com/course/the-python-mega-course/learn/quiz/4690380#overview

ıly Numbers (E)

ne a function that takes as a parameter a list that contains both numbers and strings and returns the list containing only the numbers. For nple, if I called your function with `foo([99, 'no data', 95, 94, 'no data'])` it should return `[99, 95, 94]`.

```python
def foo(lst):
    return [i for i in lst if not isinstance(i, str)]
```

✓ **Well done, your solution is correct!**

Course content

☑ Coding Exercise 36: Only Numbers (E)

□ Coding Exercise 37: Only Positive Numbers (E)

□ 62. List Comprehension with If-Else Conditional
2min

□ Coding Exercise 38: Zeros Instead (E)

□ Coding Exercise 39: Convert and Sum Up (E)

□ 63. Summary: List Comprehensions
1min

Section 10: More on Functions
0 / 10 | 10min

20-05-2020

Section13

Loading the json Data to the python.

-> In the terminal window.

```
>>> import json
>>> data = json.load(open("data.json"))
>>> type(data)
    <class 'dict'>
>>> print(data)
```

* Inorder to extract worn from data.

ex:- >>> datat ["rain"]

this will extract the word rain from data.

-> For Returning the Defination of a word:-

create a new file ex:- app1.py [in python]

~~1. import js~~

```
1. import json
2. data = json.load(open ("data json"))
3. def translate (word):
        return data [word]
4.     word = input ("Enter word:")
5.
6.     print(translate (word))
```

* to check o/p. open terminal window.

```
>>> python app1.py.
    Enter word: rain.
```

Note:- to check weather that word is there in the data. ex:-

```
>>> "rain" in data.
    True
>>> "asdasfd" in data
    False
```

So

```
import json
data = json.load(open("data.json"))
def translate(w):
    if w in data:
        return data[w]
    else:
        return "The word doesn't exist. Please ch
    word = input ("Enter word:")
    print(translate(word))
```

o/p :- Enter word : asdfasdsdf.

```
the word doesn't exist. Please check it
```

* and also.

```
Enter word : Ralu
-the word doesn't exist. please check
```

* So to over come the case sensitivity

```
import json
data = json.load(open ["data.json"])
def translate.(word):
        w = w.lower()
    if w in data:
        return data[w]
else:
    resdturn "the word doesnot exist. pl
                   double check it".
word = input [" Enter word:")
Print (translate(word))
```

o/p Enter word: rainn

```
word doesnot exist.
```

* So to over come this.

* Inorder to get the semilarity Ratio B/w two words such as raiun = rain.

```
>>> get_close_matches ("raeun", "pyiamid", "rain"])
['rain']
```

* In order to extract keys from dictionary.

```
>>> data.Keys()
```

```
>>> get_ close-matches ["raiun", data.key(), n=5)
['rain', 'train', 'rainy', 'grain', 'drain']
```

```
~~import json~~
~~data=json.load (open ("data.json"))~~
~~def translate(w):~~
~~    w=w.lower()~~
```

```
import json.
from difflib import get_close_matches
# data = json.load (open ("data.json"))
def translate(w):
    w = w.lower()
    if w in data(w)
    elif len(get_close_matches (w, data.keys()))>0:
        return "Did you mean %s instead?" get_close_matches
                        (w, data.keys()[0]
    else:
        return "The word doesn't exist, please
                double check it."
word = input ("Enter word:")
print (translate (word))
```

O/P Enter the word: rainn
did you mean a rain instead?

* Confirmation from the user

```
1. import json
2. from difflib import get_close_matches
3. data = json.load (open ("data.json"))
4. def translate (w):
5.   w = w.lower()
6.   if w in data:
7.   return data (w)
8.   elif len (get.. close_matches (w,data.keys()))
9.   yn = input ("Did u mean %s instead ?. Enter
10.            Y if yes, or N if No:" % get_close_match
11.            (w, data.keys ())[0])
12.   if yn == "Y":
13.     return data [get_close_matches(w,data.
Continue
18. else:
19.    return "The word doesn't exist. Please double
20. word = input ("Enter word:")
21. print (translate (word)).

14. elif yn = "N":
15. return "the word doesn't exist. Please
..           double check it."
16. else :
17.  return "we didn't understand your entry.
```

* to optimize the final output

```
21. output = input("Enter word:")
22. if type (output) == list:
23. for item in output:
24.     print (item)
~~24~~
25. else:
26.     print (output)
```