Do you know Google Colab? Free GPUs to run ML models ;) https://colab.research.google.com/drive/1W-PoyOGCXgzDkac46pwdqQy_8eUaKebv?usp=sharing

1. How to get indices of N maximum values in a NumPy array?

```
import numpy as np

x = np.array([1, 3, 2, 5, 0, -1, 10])
print(f"x: {x}")
```

```
x: [ 1  3  2  5  0 -1 10]
```

```
N = 3
max_x = np.argsort(-x)[:N]
print(max_x)
```

```
[6 3 1]
```

2. Mention the use of // operator in Python?

```
10 / 3
```

```
3.3333333333333335
```

```
10 // 3
```

```
3
```

3. What is the difference between a list and a tuple?

```
mylist = [1, 2, 3, 4]
print(f"mylist: {mylist}")
print(f"type(mylist): {type(mylist)}")
```

```
mylist: [1, 2, 3, 4]
type(mylist): <class 'list'>
```

```
mytuple = (1, 2, 3, 4)
print(f"mytuple: {mytuple}")
print(f"type(mytuple): {type(mytuple)}")
```

```
mytuple: (1, 2, 3, 4)
type(mytuple): <class 'tuple'>
```

```
mylist[1] = 5
print(f"mylist: {mylist}")
```

```
mylist: [1, 5, 3, 4]
```

```
mytuple[1] = 5
```

```
---------------------------------------------------------------------------
--
TypeError                                 Traceback (most recent call
last)
<ipython-input-8-eb494d40ecde> in <module>()
----> 1 mytuple[1] = 5

TypeError: 'tuple' object does not support item assignment
```

4. What would be the output of the following?

```
a = [1,2,3]
```

```
b = a
c = [1,2,3]


print(a == b) # True
print(a == c) # True
```

```
True
True
```

```
print(a is b) # True => a and b point to the same object
print(a is c) # False => a and c do not point to the same object
```

```
True
False
```

```
A0 = dict(zip(('a', 'b', 'c', 'd', 'e'),
              (1, 2, 3, 4, 5)))
A1 = range(10)
A2 = sorted([i for i in A1 if i in A0])
A3 = sorted([A0[s] for s in A0])
A4 = [i for i in A1 if i in A3]
A5 = {i:i*i for i in A1}
A6 = [[i, i*i] for i in A1]


list(range(10) )
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
print(f"A0={A0}\nA1={A1}\nA2={A2}\nA3={A3}\nA4={A4}\nA5={A5}\nA6={A6}\n")
```

```
A0={'a': 1, 'b': 2, 'c': 3, 'd': 4, 'e': 5}
A1=range(0, 10)
```

5. Define a class named car with 2 attributes, “color” and “speed”. Then create an instance and return speed.

```
class car():
    def __init__(self, color: str, speed: float):
        self.color = color
        self.speed = speed

ferrari = car(color='red', speed=300)
print(ferrari.speed)
```

```
300
```

6. Write a regular expression that will accept an email id. Use the re module.

```
import re
regex = '^[a-z0-9]+[\._]?[a-z0-9]+[@]\w+[.]\w+$'

def check(email):

    # pass the regular expression
    # and the string in search() method
    if(re.search(regex,email)):
        print("Valid Email")

    else:
        print("Invalid Email")


check(email="guillaume.simo@hotmail.fr")
```

```
Valid Email
```

```
check(email="invalidmailahotmail.fr")
```

```
Invalid Email
```

```
check(email="invalidmail@hotmailfr")
```

```
Invalid Email
```

```
check(email="invalidmailhotmail.fr")
```

```
Invalid Email
```

7. If you have to choose between a list, set, and a dictionary to store 10 million integers, what will you use? Bear in mind that you would later like to query the frequency of a number within the dataset.

```
import time
start = time.time()
mylist = list(range(10**6)) + [1]
from collections import Counter
freq_count = Counter(mylist)
print(freq_count[1])
print(freq_count[4])
print('This costs {:.4f} s.'.format(time.time() - start))
```

```
2
1
This costs 0.1350 s.
```