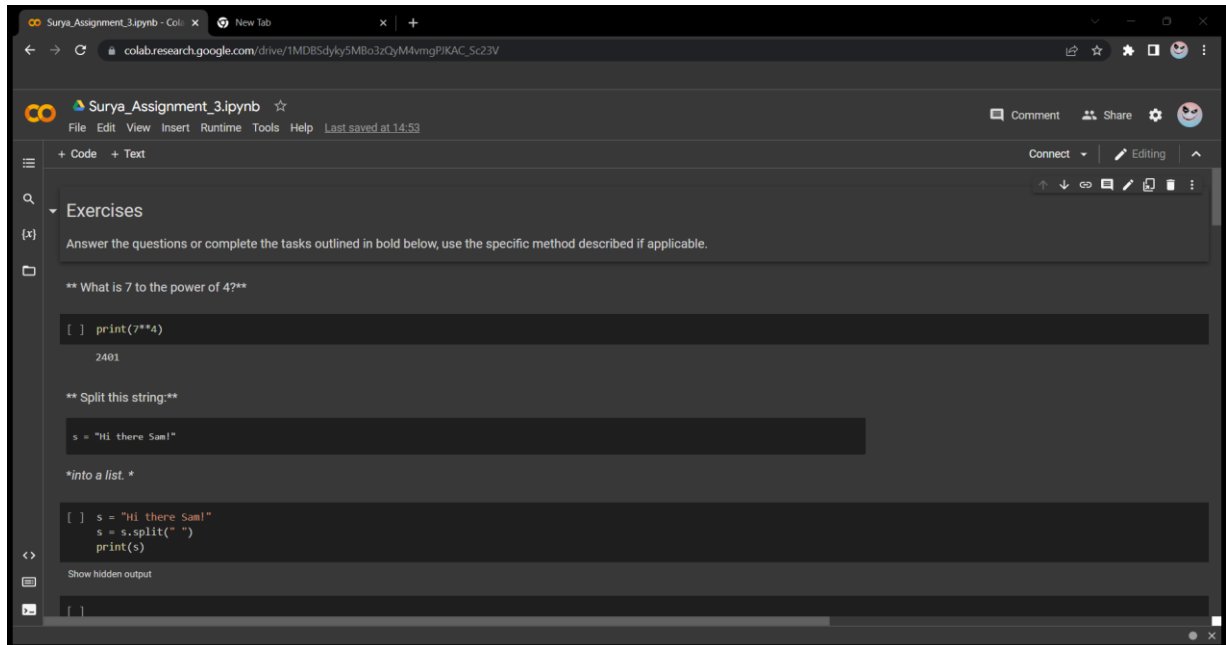


Name: Surya L

Roll No.: 727619BIT036



Screenshot of a Google Colab notebook titled "Surya_Assignment_3.ipynb". The notebook is open in a web browser at the URL `colab.research.google.com/drive/1MDBSdyky5MBo3zQyM4vmgPIKAC_Sc23V`. The interface shows a menu bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". The "Code" tab is selected, and the notebook content is displayed in a dark theme. The exercises section contains the following text and code:

```
Exercises

Answer the questions or complete the tasks outlined in bold below, use the specific method described if applicable.

** What is 7 to the power of 4? **

[ ] print(7**4)

2401

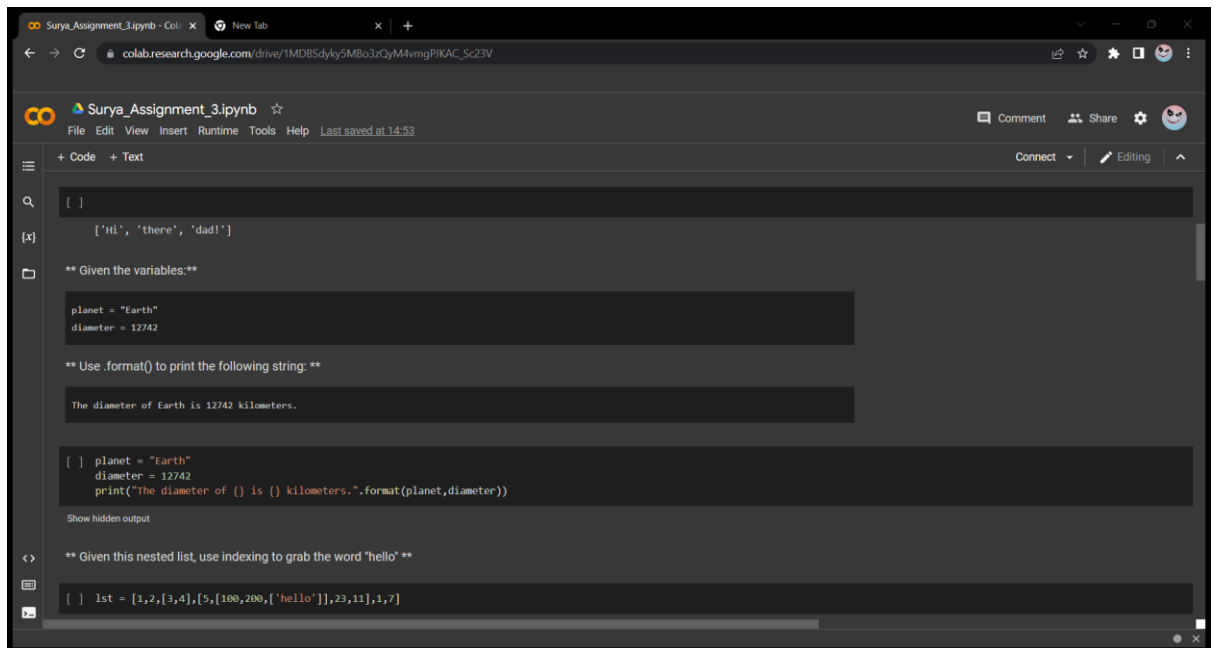
** Split this string: **

s = "Hi there Sam!"

*into a list.*

[ ] s = "Hi there Sam!"
    s = s.split(" ")
    print(s)
```

The output of the code shows the result of the string split operation: `['Hi', 'there', 'Sam!']`.



Screenshot of a Google Colab notebook titled "Surya_Assignment_3.ipynb". The notebook is open in a web browser at the URL `colab.research.google.com/drive/1MDBSdyky5MBo3zQyM4vmgPIKAC_Sc23V`. The interface shows a menu bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". The "Code" tab is selected, and the notebook content is displayed in a dark theme. The exercises section contains the following text and code:

```
[ ]

['Hi', 'there', 'dad!']

** Given the variables: **

planet = "Earth"
diameter = 12742

** Use .format() to print the following string: **

The diameter of Earth is 12742 kilometers.

[ ] planet = "Earth"
    diameter = 12742
    print("The diameter of {} is {} kilometers.".format(planet,diameter))

Show hidden output

** Given this nested list, use indexing to grab the word 'hello' **

[ ] lst = [1,2,[3,4],[5,[100,200,['hello']],23,11],1,7]
```

The output of the code shows the result of the string formatting operation: `The diameter of Earth is 12742 kilometers.`

```
[ ] lst = [1,2,[3,4],[5,[100,200,['hello']],23,11],1,7]

[ ] print(lst[3][1][2][0])

hello

** Given this nest dictionary grab the word 'hello'. Be prepared, this will be annoying/tricky **

[ ] d = {'k1':{1,2,3,{'tricky':{'oh','man','inception',{'target':[1,2,3,'hello']}}}}}

[ ] print(d["k1"][3]["tricky"][3]["target"][3])

hello

** What is the main difference between a tuple and a list? **

[ ] print("List and Tuple in Python are the classes of Python Data Structures. The list is dynamic, whereas the tuple has static characteristics. This means that lists can be modified whr")

Show hidden output

** Create a function that grabs the email website domain from a string in the form: **

user@domain.com
```

```
So for example, passing "user@domain.com" would return: domain.com

[ ] def email(email):
    return str(email).split("@")[1]

[ ] email_var = "user@domain.com"
print(email(email_var))

domain.com

** Create a basic function that returns True if the word 'dog' is contained in the input string. Don't worry about edge cases like a punctuation being attached to the word dog, but do account for capitalization. **

[ ] def word_found(word):
    return True if "dog" in str(word) else False

[ ] word = "Don't worry about edge cases like a punctuation being attached to the word dog, but do account for capitalization"
print(word_found(word))

True

** Create a function that counts the number of times the word "dog" occurs in a string. Again ignore edge cases. **
```

Surya_Assignment_3.ipynb - Colab

colabresearch.google.com/drive/1MDBSdyky5MB03rQyM4vmgPIKAC_Sc23V

Surya_Assignment_3.ipynb

File Edit View Insert Runtime Tools Help Last saved at 14:53

+ Code + Text

Connect Editing

** Create a function that counts the number of times the word "dog" occurs in a string. Again ignore edge cases. **

```
[ ] def word_found_counts(word):
    return word.count("dog")

[ ] word = "Don't worry about edge cases like a punctuation being attached to the word dog, but do account for capitalization dogs"
print(word_found_counts(word))

2
```

Problem

*You are driving a little too fast, and a police officer stops you. Write a function to return one of 3 possible results: "No ticket", "Small ticket", or "Big Ticket". If your speed is 60 or less, the result is "No Ticket". If speed is between 61 and 80 inclusive, the result is "Small Ticket". If speed is 81 or more, the result is "Big Ticket". Unless it is your birthday (encoded as a boolean value in the parameters of the function) – on your birthday, your speed can be 5 higher in all cases. *

```
[ ] def caught_speeding(speed, is_birthday):

    if is_birthday:
        speeding = speed + 5
    else:
        speeding = speed

    if speeding <= 80 :
        return "No Ticket"
    elif speeding <= 85:
        return "Small Ticket"
    else:
        return "Big Ticket"
```

Surya_Assignment_3.ipynb - Colab

colabresearch.google.com/drive/1MDBSdyky5MB03rQyM4vmgPIKAC_Sc23V

Surya_Assignment_3.ipynb

File Edit View Insert Runtime Tools Help Last saved at 14:53

+ Code + Text

Connect Editing

```
[ ] if speeding <= 80 :
    return 'Small Ticket'
elif speeding <= 85:
    return 'No Ticket'
else:
    return 'Big Ticket'

[ ] print(caught_speeding(81,True))

Big Ticket

[ ] print(caught_speeding(80,False))

Small Ticket
```

Create an employee list with basic salary values(at least 5 values for 5 employees) and using a for loop retrieve each employee salary and calculate total salary expenditure.

```
[ ] employee = [{"Muqsid",800000}, {"Bhavna",700000}, {"Rithika",600000}, {"Adhish",500000}, {"surya",400000}]
total = 0
for i in employee:
    print(i[1])
    total += i[1]

print(total)
```

Surya_Assignment_3.ipynb - Colab

colabresearch.google.com/drive/1MDBSdyky5MB03rQyM4vmgPIKAC_Sc23V

Surya_Assignment_3.ipynb

File Edit View Insert Runtime Tools Help Last saved at 14:53

+ Code + Text

Connect Editing

```
[ ] print(caught_speeding(80,False))
```

Small Ticket

Create an employee list with basic salary values(at least 5 values for 5 employees) and using a for loop retrieve each employee salary and calculate total salary expenditure.

```
employee = [{"Empid":1,"Empname":"Surya", "Basicpay": 80000.00}, {"Empid":2,"Empname":"Rithika", "Basicpay": 60000.00}, {"Empid":3,"Empname":"Adhish", "Basicpay": 50000.00}, {"Empid":4,"Empname":"Surya", "Basicpay": 40000.00}, {"Empid":5,"Empname":"Surya", "Basicpay": 30000.00}]
total = 0
for i in employee:
    print(i[1])
    total += i[2]

print(total)
```

800000
700000
600000
500000
400000
3000000

Create two dictionaries in Python:
First one to contain fields as Empid, Empname, Basicpay

Surya_Assignment_3.ipynb - Colab

colabresearch.google.com/drive/1MDBSdyky5MB03rQyM4vmgPIKAC_Sc23V

Surya_Assignment_3.ipynb

File Edit View Insert Runtime Tools Help Last saved at 14:53

+ Code + Text

Connect Editing

First one to contain fields as Empid, Empname, Basicpay

Second dictionary to contain fields as DeptName, DeptId.

Combine both dictionaries.

```
[ ] basic_emp = {
    "Empid": 1,
    "Empname": "Surya",
    "Basicpay": 80000.00
}

second_dict = {
    "DeptName": "Information Technology",
    "DeptId": 7276
}

combine = {
    "Empid": basic_emp["Empid"],
    "Empname": basic_emp["Empname"],
    "Basicpay": basic_emp["Basicpay"],
    "DeptName": second_dict["DeptName"],
    "DeptId": second_dict["DeptId"]
}

print(combine)
```

```
{'Empid': 1, 'Empname': 'Surya', 'Basicpay': 80000.0, 'DeptName': 'Information Technology', 'DeptId': 7276}
```