

Internship Weekly Report – Week 1

◆ Title Page

- **Name:** Sandeep Ravaji Patel
 - **Domain:** Data Science
 - **Week Number:** Week 1
-

◆ Task Description

Objective:

To get started with Python programming and develop a foundational understanding through basic syntax, control structures, functions, and data structures.

Tasks Completed:

1. Python Basics Practice:

- Data types (integers, strings, lists, dictionaries).
- Conditional statements (if, elif, else).
- Loops (for, while) for iteration.
- Functions: User-defined functions and function arguments.

2. Data Manipulation Tasks:

- Worked with lists and dictionaries to store and process structured data.
- Implemented tasks like counting word frequency, grouping data, filtering values, and performing lookups.

3. Notebook 1 – Basic Programming:

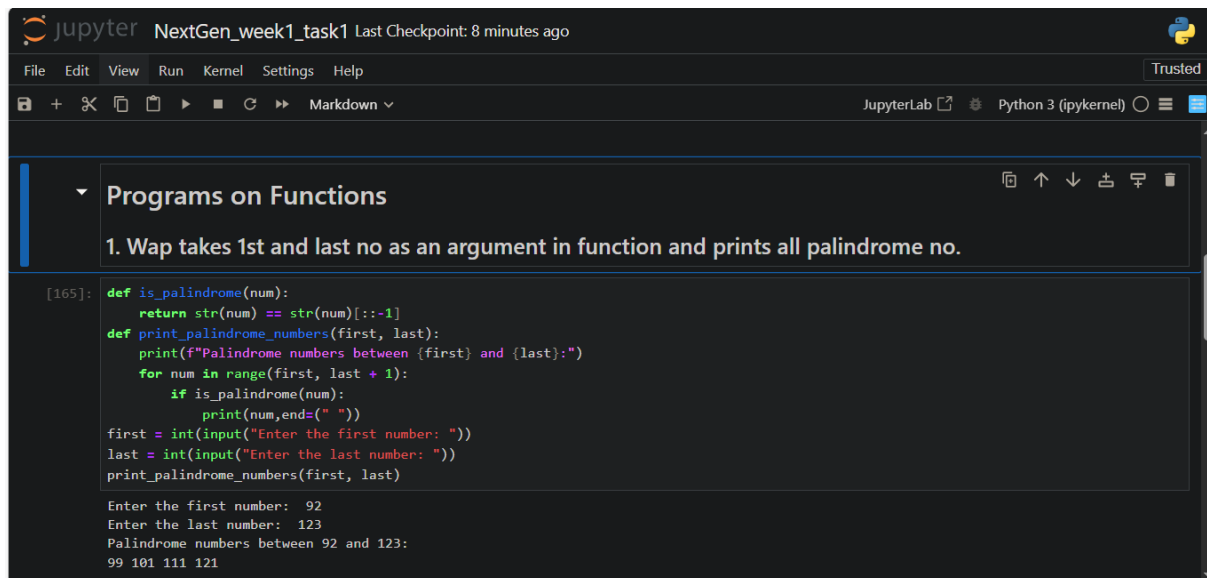
- Wrote programs to print palindromes, Armstrong numbers, and calculate power using functions.
- Practiced number analysis (sum/count of digits, etc.).

4. Notebook 2 – Data Manipulation:

- Created mini applications using lists and dictionaries.
 - Examples: Contact book, student marks analysis, employee grouping.
-

◆ Code Snippets / Design Screenshots

Example: Palindrome Number Function



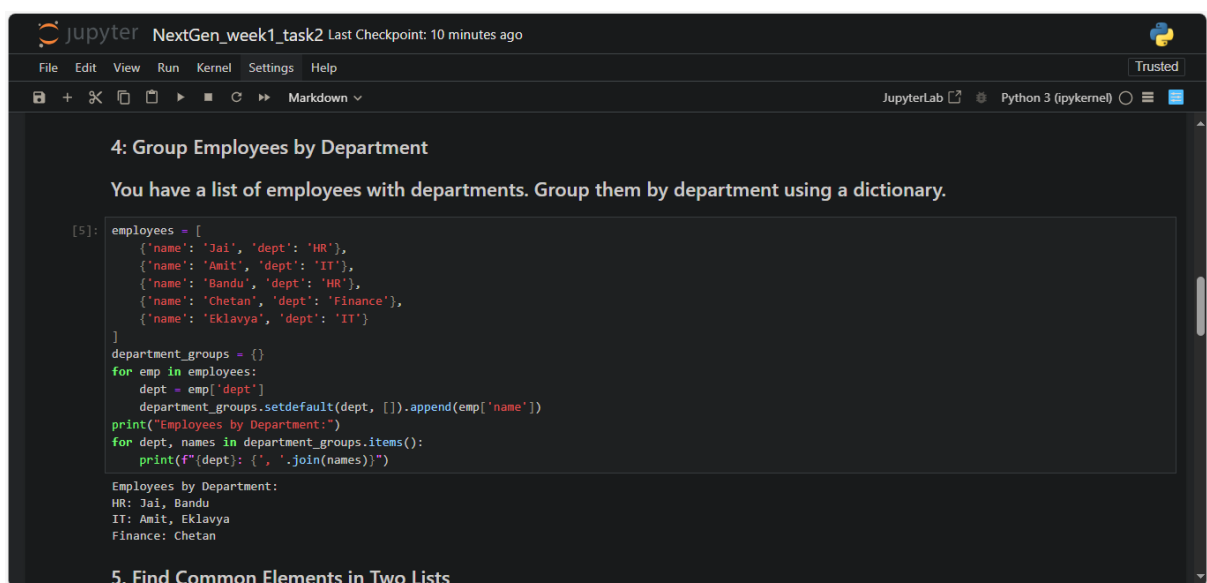
The screenshot shows a JupyterLab window titled 'NextGen_week1_task1' with a 'Trusted' status. The interface includes a menu bar (File, Edit, View, Run, Kernel, Settings, Help) and a toolbar with icons for file operations and execution. A sidebar on the left displays a file tree with 'Programs on Functions' expanded, showing a file named '1. Wap takes 1st and last no as an argument in function and prints all palindrome no.'. The main editor area contains a Python code cell with the following code:

```
[165]: def is_palindrome(num):  
        return str(num) == str(num)[::-1]  
def print_palindrome_numbers(first, last):  
    print(f"Palindrome numbers between {first} and {last}:")  
    for num in range(first, last + 1):  
        if is_palindrome(num):  
            print(num, end=" ")  
first = int(input("Enter the first number: "))  
last = int(input("Enter the last number: "))  
print_palindrome_numbers(first, last)
```

The output of the code is displayed below the cell:

```
Enter the first number: 92  
Enter the last number: 123  
Palindrome numbers between 92 and 123:  
99 101 111 121
```

Example: Data Grouping with Dictionary



The screenshot shows a JupyterLab window titled 'NextGen_week1_task2' with a 'Trusted' status. The interface includes a menu bar (File, Edit, View, Run, Kernel, Settings, Help) and a toolbar with icons for file operations and execution. A sidebar on the left displays a file tree with '4: Group Employees by Department' expanded, showing a file named 'You have a list of employees with departments. Group them by department using a dictionary.'. The main editor area contains a Python code cell with the following code:

```
[5]: employees = [  
    {'name': 'Jai', 'dept': 'HR'},  
    {'name': 'Amit', 'dept': 'IT'},  
    {'name': 'Bandu', 'dept': 'HR'},  
    {'name': 'Chetan', 'dept': 'Finance'},  
    {'name': 'Eklavya', 'dept': 'IT'}  
]  
department_groups = {}  
for emp in employees:  
    dept = emp['dept']  
    department_groups.setdefault(dept, []).append(emp['name'])  
print("Employees by Department:")  
for dept, names in department_groups.items():  
    print(f"{dept}: {' '.join(names)}")
```

The output of the code is displayed below the cell:

```
Employees by Department:  
HR: Jai, Bandu  
IT: Amit, Eklavya  
Finance: Chetan
```

Below the code cell, the next task is listed: '5. Find Common Elements in Two Lists'.

◆ Challenges Faced

- Initially struggled with proper indentation in Python loops and conditionals.
- Faced confusion distinguishing between mutable and immutable data types (e.g., lists vs tuples).
- Minor issues using Google Colab and Jupyter for the first time, especially with markdown formatting and cell execution.

How They Were Resolved:

- Referred to W3Schools and official Python documentation for syntax and examples.
 - Practiced writing short programs to reinforce understanding.
 - Learned by debugging and reviewing error messages.
-

◆ Learning Outcome

- Gained hands-on experience with core Python concepts.
 - Learned how to manipulate and process basic data structures.
 - Strengthened logical thinking and debugging skills.
 - Understood the difference between writing a script and designing small problem-solving applications.
-

◆ Next Steps

For **Week 2**, the focus will be on:

- Learning data handling using **Pandas** and **NumPy**.
- Performing operations like filtering, grouping, and sorting on datasets.
- Working with real-world CSV/Excel files.