🐍 Python Functions & Logic - Day 2 Notes

Theme: Building Logic with Simple Python Functions

📚 Table of Contents

---

1. Using External Modules

🔍 What are External Modules?

External modules are pre-written code packages that add extra features to Python.

Example: QR Code Generator

```python
# First install the module
!pip install qrcode

# Then use it
import qrcode
img = qrcode.make('https://www.google.com')
img.save('myqr.png')
```

💡 Real-life Use:

- Generate QR codes for your college website
- Create QR codes for UPI payments
- Make QR codes for social media links

---

2. Palindrome Check

🔍 What is a Palindrome?

A word that reads the same forwards and backwards.

Examples: nitin, malayalam, radar

Simple Method:

python

```python
word = 'nitin'
if word == word[::-1]:
    print(f'{word} is a Palindrome')
else:
    print(f'{word} is NOT a Palindrome')
```

Better Method (Using Function):

python

```python
def check_palindrome(word):
    word = word.upper()
    if word == word[::-1]:
        return 'Palindrome'
    else:
        return 'Not a Palindrome'

print(check_palindrome('Ajay'))   # Not a Palindrome
print(check_palindrome('nitin'))  # Palindrome
```

🔑 Key Point:

`[::-1]` is a magic trick to reverse any string!

---

3. Fibonacci Series

🔍 What is Fibonacci?

Each number is the sum of the previous two numbers.

Pattern: 0, 1, 1, 2, 3, 5, 8, 13...

Simple Code:

python

```
def give_fibo(n):
    fibo = [0, 1]
    for i in range(n - 2):
        next_num = fibo[-1] + fibo[-2]
        fibo.append(next_num)
    return fibo

print(give_fibo(7))  # [0, 1, 1, 2, 3, 5, 8]
```

💡 Real-life Use:

- Stock market analysis
- Nature patterns (flower petals, shells)
- Computer algorithms

---

4. Prime Number Check

🔍 What is a Prime Number?

A number that can only be divided by 1 and itself.

Examples: 2, 3, 5, 7, 11, 13...

Simple Code:

```python
def check_prime(number):
    for i in range(2, number):
        if number % i == 0:
            return 'Not a Prime Number'
    return 'Prime Number'

print(check_prime(5))   # Prime Number
print(check_prime(12))  # Not a Prime Number
```

🔑 Key Point:

We check if any number from 2 to (number-1) can divide it evenly.

---

5. Pattern Printing

Left Aligned Stars:

python

```python
n = 5
for i in range(1, n+1):
    print('* ' * i)
```

Output:

```
*
* *
* * *
* * * *
* * * * *
```

Right Aligned Stars:

python

```python
n = 5
for i in range(1, n+1):
    print(' ' * (n-i) + '* ' * i)
```

Output:

```
    *
   * *
  * * *
 * * * *
* * * * *
```

💡 Logic:

- Left: Just print stars
- Right: Add spaces before stars

---

6. Working with Lists

Negative Indexing:

python

```python
data = [34, 65, 654, 76, 856]
print(data[-1])  # 856 (last element)
print(data[-2])  # 76 (second last)
```

🔑 Key Point:

- Positive index: Start from beginning (0, 1, 2...)
- Negative index: Start from end (-1, -2, -3...)

---

7. Basic Math Functions

Sum of Natural Numbers:

python

```python
def sum_of_n_natural_numbers(n):
    result = 0
    for i in range(1, n+1):
        result += i
    return result

print(sum_of_n_natural_numbers(10))  # 55
```

Factorial:

python

```python
def factorial(n):
    result = 1
    for i in range(1, n+1):
        result *= i
    return result

print(factorial(5))  # 120
```

💡 Logic:

- Sum: Keep adding numbers
- Factorial: Keep multiplying numbers

---

8. Function Arguments

Using *args (Multiple Values):

python

```python
def total_sales(*args):
    result = 0
    for i in args:
        result += i
    return result

print(total_sales(100, 200, 300))  # 600
```

Using **kwargs (Named Arguments):

python

```python
def student_info(**kwargs):
    for key, value in kwargs.items():
        print(f"{key}: {value}")

student_info(name="Rahul", age=20, city="Delhi")
```

🔑 Key Points:

- `*args`: For unlimited regular arguments
- `**kwargs`: For unlimited named arguments

---

9. Mini Assignments

📝 Practice Problems:

1. **Even or Odd Function**
   - Write a function to check if a number is even or odd
   - Hint: Use `%` operator
2. **Pyramid Pattern**
   - Create a pyramid using stars
   - Make it centered
3. **Hashtag Generator**
   - Add `#` before startup names
   - Example: "Flipkart" → "#Flipkart"
4. **User Input Factorial**
   - Ask user for a number
   - Calculate and display factorial
5. **Student Records**
   - Store multiple student data using `**kwargs`

- Display in a nice format

---

10. Key Takeaways

🎯 Important Points:

1. **External Modules** expand Python's power
   - Use `pip install` to get new features
2. **String Slicing** is powerful
   - `[::-1]` reverses any string
3. **Functions** make code reusable
   - Write once, use many times
4. **Logic Building** is key
   - Start with simple examples
   - Build complexity gradually
5. **Real-world Applications**
   - Every concept has practical uses
   - Practice with real examples

🔧 Quick Tips:

- Always test your code with different inputs
- Use meaningful variable names
- Comment your code for clarity
- Practice daily with small programs

📖 Study Plan

Today's Focus:

- Practice writing simple functions
- Understand logic building
- Work on pattern problems

Tomorrow's Preview:

- File handling
- Error handling
- More advanced functions

Remember: Programming is like learning to ride a bike - practice makes perfect! 🚴

Happy Coding! 🎉