Introduction

Hello guys! Nene Rahul Medapati | So welcome to the Python notes. Ee notes lo oka full course mi kosame chesa. I hope these notes will help you to understand Python better. Let's dive into the first topic!

Topic 1: Comments

Print (“Hello ,world !”)

#output :- Hello ,world!

1 .Comments in Python

Symbols vacchesi (“#” , “”” ”””)

Comments ante neeku and verevallaki code ento chadavadaniki easy ga cheppadaniki help chestayi. Python lo comments rayadaniki rendu ways unnayi: single-line comments and multi-line comments.

1. **Single-Line Comments**:
    - Syntax: # Mee comment ikkada

Example:

```python
# Idi oka single-line comment
print("Hello, World!")  # Idi oka greeting message ni print chestundi
```

2. **Multi-Line Comments**:
    - Syntax:

```
"""
Idi oka multi-line comment.
Neeku chala lines lo comments rayavachu.
"""
```

Example:

```
"""
Idi oka multi-line comment.
Idi code yela pani chesthundo leda notes ichhi explain cheyyali ante use chestaru.
"""
print("Hello, World!")  # Idi oka greeting message ni print chestundi
```

Explanation :

1. **Single-Line Comments**:
   - # use chesi quick note rayandi. # taruvatha ochina prathi vishayam comment ga consider chestaru. Think of it like sticky note meeda quick reminder rayadam laga.
2. **Multi-Line Comments**:
   - Triple quotes """ use chesi ekkuva explanation rayandi. Idi code yela pani chesthundo detailed ga explain cheyyadam laga undi. Idi code neat ga and understandable ga chese help chestundi.

Of course! Here's a refined explanation for Topic 2 on data types and variables:

Topic 2: Data Types and Variables

Data Types in Python

Python lo data types ante different types of data ne manage cheyyadam. Python chala varieties of data types support chestundi. Chuddam:

1. **Integers (int)**:
   - Numbers without decimal points. Example: 5, 100, -20
   - Example usage:

```
age = 25  # Age is an integer
```

2. **Floats (float)**:

- Numbers with decimal points. Example: 5.0, 3.14, -0.5
- Example usage:

```
height = 5.9  # Height is a float
```

3. **Strings (str)**:
   - Text data enclosed in quotes. Example: "Hello", 'Python', "123"
   - Example usage:

```
name = "Rahul"  # Name is a string
```

4. **Booleans (bool)**:
   - Logical values True or False. Used for decision making.
   - Example usage:

```
is_student = True  # is_student is a boolean
```

5. **Lists**:
   - Ordered collection of items that can be of different types. Enclosed in square brackets.
   - Example: [1, 2, 3], ['apple', 'banana']
   - Example usage:

```
fruits = ["apple", "banana", "cherry"]  # fruits is a list
```

6. **Dictionaries**:
   - Unordered collection of key-value pairs. Enclosed in curly braces.
   - Example: {'name': 'Rahul', 'age': 25}
   - Example usage:

```
person = {"name": "Rahul", "age": 25}  # person is a dictionary
```

Variables in Python

Variables are like containers where you can store your data. Meeku data ni store chesi, malli use cheyali ante variables use chestam.

1. **Creating Variables**:
    - Syntax: variable_name = value
    - Example:

```
x = 10  # x is a variable that holds the integer 10
```

2. **Variable Naming Rules**:
    - Variable names should start with a letter or an underscore (_).
    - They can contain letters, numbers, and underscores, but not start with a number.
    - Example: my_variable, _value, value1
3. **Updating Variables**:
    - You can change the value of a variable anytime.
    - Example:

```
x = 10
x = 20  # Now x holds the value 20
```

Explanation :

1. **Data Types**:
    - Python lo different types of data handle cheyyadaniki different data types use chestaru. Example, numbers, text, boolean values, etc.
        - **Integers**: Plain numbers, no decimal.
        - **Floats**: Numbers with decimal.
        - **Strings**: Text data.
        - **Booleans**: True or False.
        - **Lists**: Collection of items, ordered.
        - **Dictionaries**: Key-value pairs, unordered.
2. **Variables**:
    - Variables are like boxes where you can keep different types of data. You can name them and use them to store values.

- **Creating**: Assign value using variable_name = value.
- **Naming**: Should start with a letter or _, and can have letters, numbers, or _.
- **Updating**: Change the value whenever needed.

Topic 1: Conditional Statements in Python

Conditional Statements

Conditional statements help to run different blocks of code based on different conditions. In Python, you use if, elif, and else for this.

1. if Statement

if statement condition ni check chestundi, condition true aina appudu code block execute avutundi.

Example : -

```
x = 10
if x > 5:
    print("x is greater than 5")
```

Explanation: if x > 5 ante x 5 kanna pedda kada ani check chestundi. x 10 undi, so condition true, "x is greater than 5" ani print chestundi.

2. elif Statement

elif (else if) valla munde if condition false aina appudu maroka condition check chesthundi.

Example: -

```
x = 10
if x > 15:
    print("x is greater than 15")
elif x > 5:
    print("x is greater than 5 but less than or equal to 15")
```

Explanation: elif x > 5 ante x 15 kanna takkuva, kani 5 kanna pedda undi ani check chestundi. x 10 undi, so "x is greater than 5 but less than or equal to 15" ani print chestundi.

3. else Statement

else valla if and elif conditions anni false aithe, default ga koddi block execute avutundi.

Example: -

```
x = 3
if x > 5:
    print("x is greater than 5")
elif x > 0:
    print("x is positive but less than or equal to 5")
else:
    print("x is not positive")
```

Explanation: else block anedi if and elif conditions anni false aithe run avutundi. x 3 undi, so x > 5 condition false, kani x > 0 condition true. So, "x is positive but less than or equal to 5" ani print chestundi. Leda, else block run avvachu.

Topic 4: Looping Statements in Python

Looping statements help to execute a block of code multiple times based on certain conditions. Python lo for and while loops use chestaru.

1. for Loop

for loop valla iterable object lo elements ni iterate chesi, prathi element ki code block execute chestundi.

Example:

```
for i in range(5):
    print(i)
```

2.while loop

while loop condition true aina appudu loop repeat avuthundi. Condition false aithe loop terminate avutundi.

Example: -

```
count = 0
while count < 5:
    print(count)
    count += 1
```

Explanation: while count < 5 anedi count 5 kanna takkuva undi appudu loop run avuthundi. count value ni print chesi, count ni 1 increment chestundi. Output: 0, 1, 2, 3, 4.

Topic 5: Jumping Statements in Python

Jumping statements use chesi loop execution ni control cheyochu. Python lo main jumping statements break, continue, and pass.

1. break Statement

break statement loop ni immediate ga terminate chestundi.

Example:

```
for i in range(5):
    if i == 3:
        break
    print(i)
```

Explanation: if i == 3 condition true aithe break statement loop ni terminate chestundi. So, loop i 0, 1, 2 values ni print chestundi, kani 3 value vachinappudu loop stop avutundi.

2. continue Statement

continue statement current iteration ni skip chesi, next iteration start chestundi.

Example:

```
for i in range(5):
    if i == 3:
        continue
    print(i)
```

Explanation: if i == 3 condition true aithe, continue statement current iteration ni skip chestundi. So, 3 value print avvadu, remaining values 0, 1, 2, 4 print avuthayi.

3. pass Statement

pass statement anedi placeholder laga use chestundi. Koni code blocks lo, temporary ga emina execute cheyyakunda undaali anukunte use chestaru.

Example:

```
for i in range(5):
    if i == 3:
        pass
    else:
        print(i)
```

Explanation: if i == 3 condition true aithe pass statement execute avvadu, just placeholder laga undi. i values 0, 1, 2, 4 print avuthayi.

Topic 6: Strings in Python

Strings are sequences of characters used to store and manipulate text. In Python, strings are created using single quotes ('), double quotes ("), or triple quotes ''' or """).

1. Creating Strings

Strings create cheyadam chala simple. Single quotes, double quotes, leda triple quotes use chesi strings create cheyochu.

Examples:

```
single_quote_str = 'Hello'
double_quote_str = "World"
triple_quote_str = '''Python is awesome!'''
```

double_quote_str = "World"

triple_quote_str = '''Python is awesome!'''

Explanation: single_quote_str anedi single quotes use chesi string create chestundi. double_quote_str double quotes use chestundi. triple_quote_str multiple lines span chese strings ki use chestaru.

2. Accessing Characters

Strings lo specific character access cheyali ante indexing use cheyochu. Indexing 0 nundi start avutundi.

Examples:

```
text = "Python"
first_char = text[0]   # 'P'
last_char = text[-1]   # 'n'
```

Explanation: text[0] ante text string lo first character P ni access chestundi. text[-1] ante last character n ni access chestundi.

3. Slicing Strings

String slicing use chesi, string ni part-by-part access cheyochu. Syntax: string[start:end].

Examples:

```
text = "Python Programming"
substring = text[7:18]  # 'Programming'
```

Explanation: text[7:18] anedi text string lo 7th index nundi 18th index varaku characters ni slice chestundi.

4. String Methods

Python strings ki chala methods unnayi, vatini use chesi text manipulate cheyochu.

Examples:

- **upper()** - Converts all characters to uppercase.

```
text = "hello"
upper_text = text.upper()  # 'HELLO'
```

- **lower()** - Converts all characters to lowercase.

```
text = "WORLD"
lower_text = text.lower()  # 'world'
```

- **strip()** - Removes whitespace from the beginning and end of the string.

```
text = "  space  "
stripped_text = text.strip()  # 'space'
```

- **replace()** - Replaces a specified phrase with another phrase.

```
text = "Hello World"
new_text = text.replace("World", "Python")  # 'Hello Python'
```

- **split()** - Splits the string into a list based on a specified delimiter.

```
text = "Python is fun"
words = text.split()  # ['Python', 'is', 'fun']
```

- **join()** - Joins elements of a list into a single string with a specified delimiter.

```
words = ['Python', 'is', 'fun']
sentence = ' '.join(words)  # 'Python is fun'
```

Explanation:

- upper() method anedi string ni uppercase lo convert chestundi.
- lower() method string ni lowercase lo convert chestundi.
- strip() method string start and end lo unna extra spaces ni remove chestundi.
- replace() method valla specified word ni maroka word tho replace chestundi.

- split() method string ni delimiter (space by default) use chesi list lo convert chestundi.
- join() method valla list elements ni specified delimiter use chesi string lo join chestundi.

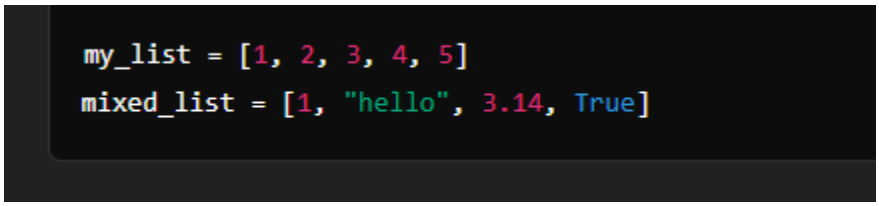
Topic 7: Lists in Python

Lists are mutable sequences used to store multiple items in a single variable. They can hold items of different data types.

1. Creating Lists

Lists create cheyadam chala easy. Square brackets [] use chesi items ni store cheyochu.

Examples:

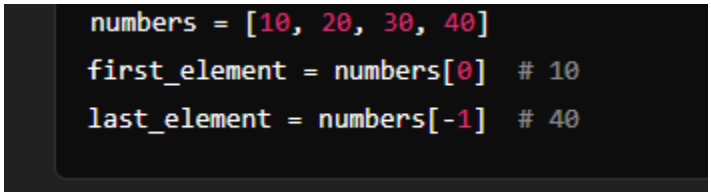


Explanation: my_list anedi integer values store chestundi. mixed_list anedi different data types (int, string, float, boolean) store chestundi.

2. Accessing List Elements

List elements access cheyali ante indexing use cheyochu. Indexing 0 nundi start avutundi.

Examples:



Explanation: numbers[0] ante first element 10 ni access chestundi. numbers[-1] ante last element 40 ni access chestundi.

3. List Methods

Lists ki chala methods unnayi, vatini use chesi list ni modify cheyochu.

Examples:

- **append()** - Adds an item to the end of the list.

```
my_list = [1, 2, 3]
my_list.append(4)  # [1, 2, 3, 4]
```

- **remove()** - Removes the first occurrence of a specified value.

```
my_list = [1, 2, 3, 2]
my_list.remove(2)  # [1, 3, 2]
```

- **sort()** - Sorts the list in ascending order.

```
my_list = [3, 1, 4, 1, 5]
my_list.sort()  # [1, 1, 3, 4, 5]
```

- **reverse()** - Reverses the order of elements in the list.

```
my_list = [1, 2, 3]
my_list.reverse()  # [3, 2, 1]
```

Topic 8: Tuples in Python

Tuples are immutable sequences used to store multiple items in a single variable. Once created, their elements cannot be changed.

1. Creating Tuples

Tuples create cheyali ante round brackets () use chesi items ni store cheyochu.

Examples:

```
my_tuple = (1, 2, 3, 4)
mixed_tuple = (1, "hello", 3.14, True)
```

Explanation: my_tuple anedi integer values store chestundi. mixed_tuple anedi different data types (int, string, float, boolean) store chestundi.

2. Accessing Tuple Elements

Tuple elements access cheyali ante indexing use cheyochu. Indexing 0 nundi start avutundi.

Examples:

```
values = (10, 20, 30, 40)
first_element = values[0]  # 10
last_element = values[-1]  # 40
```

Explanation: values[0] ante first element 10 ni access chestundi. values[-1] ante last element 40 ni access chestundi.

3. Tuple Methods

Tuples ki methods chala limited unnayi. Common methods **count()** and **index()** use chestaru.

Examples:

- **count()** - Counts the number of occurrences of a specified value.

```
my_tuple = (1, 2, 2, 3)
count = my_tuple.count(2)  # 2
```

- **index()** - Returns the index of the first occurrence of a specified value.

```
my_tuple = (1, 2, 3, 2)
index = my_tuple.index(2)  # 1
```

Topic 9: Dictionaries in Python

Dictionaries are mutable mappings used to store key-value pairs. Each key must be unique.

1. Creating Dictionaries

Dictionaries create cheyali ante curly braces {} use chesi key-value pairs ni store cheyochu.

Examples:

```
my_dict = {"name": "John", "age": 30, "city": "New York"}
```

Explanation: my_dict lo "name", "age", and "city" keys valla corresponding values store chestundi.

2. Accessing Dictionary Values

Dictionary values access cheyali ante key use chesi value retrieve cheyochu.

Examples:

```
person = {"name": "Alice", "age": 25}
name = person["name"]  # Alice
```

3. Dictionary Methods

Dictionaries ki methods like **keys()**, **values()**, and **items()** use chesi data access cheyochu.

Examples:

- **keys()** - Returns a view object displaying a list of all keys.

```
my_dict = {"name": "John", "age": 30}
keys = my_dict.keys()  # dict_keys(['name', 'age'])
```

- **values()** - Returns a view object displaying a list of all values.

```
my_dict = {"name": "John", "age": 30}
values = my_dict.values()  # dict_values(['John', 30])
```

- **items()** - Returns a view object displaying a list of key-value tuple pairs.

```
my_dict = {"name": "John", "age": 30}
items = my_dict.items()  # dict_items([('name', 'John'), ('age', 30)])
```

Topic 10: Sets in Python

Sets are unordered collections of unique items. They are useful for membership testing and eliminating duplicate entries.

1. Creating Sets

Sets create cheyali ante curly braces {} use chesi unique items store cheyochu.

Examples:

```
my_set = {1, 2, 3, 4, 5}
```

Explanation: my_set anedi unique integers ni store chestundi.

2. Set Operations

Sets lo common operations like **union()**, **intersection()**, and **difference()** perform cheyochu.

Examples:

- **union()** - Returns a set containing all items from both sets.

```
set1 = {1, 2, 3}
set2 = {3, 4, 5}
union_set = set1.union(set2)  # {1, 2, 3, 4, 5}
```

- **intersection()** - Returns a set containing items that are in both sets.

```
set1 = {1, 2, 3}
set2 = {2, 3, 4}
intersection_set = set1.intersection(set2)  # {2, 3}
```

- **difference()** - Returns a set containing items that are only in the first set.

```
set1 = {1, 2, 3}
set2 = {2, 3, 4}
difference_set = set1.difference(set2)  # {1}
```

Topic 11: Functions in Python

Functions are blocks of code that perform a specific task and can be reused. They help in organizing and managing code efficiently.

1. Defining Functions

Function create cheyali ante def keyword use chesi function ni define cheyochu.

Examples:

```
def greet(name):
    return f"Hello, {name}!"
```

Explanation: def greet(name) anedi greet function ni define chestundi, name parameter ni accept chesi "Hello, {name}!" message return chestundi.

2. Calling Functions

Function ni call cheyali ante function name use chesi arguments pass cheyochu.

Examples:

```
message = greet("Alice")  # "Hello, Alice!"
```

Explanation: greet("Alice") function ni call chestundi, "Alice" argument pass chestundi, output "Hello, Alice!" avuthundi.

3. Function Parameters

Functions ki default values, keyword arguments, and arbitrary arguments use cheyochu.

Examples:

- **Default Parameters**

```
def multiply(a, b=2):
    return a * b
```

- **Keyword Arguments**

```python
def person_info(name, age):
    return f"{name} is {age} years old."
```

- **Arbitrary Arguments**

```python
def add_numbers(*args):
    return sum(args)
```

Explanation:

- Default parameters valla function call lo parameters pass cheyyakunda default value use chestundi.
- Keyword arguments valla function ki arguments names use chesi values pass cheyochu.
- Arbitrary arguments (*args) valla unknown number of arguments pass cheyochu.

Topic 12: File Handling in Python

File handling allows you to read from and write to files. Python provides functions for working with files.

1. Opening Files

Files open cheyali ante open() function use chesi file ni specify chesina mode lo open cheyochu.

Examples:

```python
file = open('example.txt', 'r')  # Read mode
```

Explanation: 'example.txt' file ni 'r' mode lo open chestundi (read mode).

2. Reading Files

File contents read cheyali ante methods like **read()**, **readline()**, and **readlines()** use cheyochu.

Examples:

- **read()**

```
content = file.read()
```

- **readline()**

```
line = file.readline()
```

- **readlines()**

```
lines = file.readlines()
```

Explanation:

- read() anedi entire file content ni single string ga read chestundi.
- readline() anedi file lo next line read chestundi.
- readlines() anedi file lo unna prathi line ni list of strings ga read chestundi.

3. Writing to Files

File lo data write cheyali ante write() or writelines() methods use cheyochu.

Examples:

- **write()**

```
file.write("Hello, World!")
```

- **writelines()**

```
lines = ["Line 1\n", "Line 2\n"]
file.writelines(lines)
```

Explanation:

- write() anedi string ni file lo write chestundi.
- writelines() anedi list of strings ni file lo write chestundi.

4. Closing Files

File work complete aithe close() method use chesi file ni close cheyochu.

Examples:

```
file.close()
```

Explanation: file.close() anedi open chesina file ni close chestundi.

Topic 13: Error Handling in Python

Error handling is used to manage and respond to runtime errors in your code. Python uses try, except, else, and finally blocks for this.

1.Try and Except

try block lo code execute chestundi, error vachinappudu except block run avutundi.

Examples:

```
try:
    result = 10 / 0
except ZeroDivisionError:
    print("Cannot divide by zero!")
```

Explanation: try block lo zero division error jariginappudu, except ZeroDivisionError block run avuthundi and error message print chestundi.

2.Else Block

else block try block lo error lekapothe execute avuthundi.

Examples:

```python
try:
    result = 10 / 2
except ZeroDivisionError:
    print("Cannot divide by zero!")
else:
    print("Division successful!")
```

Explanation: try block lo error lekapothe else block execute avuthundi and "Division successful!" message print chestundi.

3. Finally Block

finally block always execute avuthundi, error unna or lekapothe.

Examples:

```python
try:
    file = open('example.txt', 'r')
finally:
    file.close()
```

Explanation: finally block always file.close() execute chestundi, file open chesaka close chestundi.

Topic 14: Object-Oriented Programming (OOP) in Python

OOP is a programming paradigm based on objects and classes. It helps in organizing code in a modular way.

1. Classes and Objects

Class define cheyali ante class keyword use chesi, object create cheyali ante class ni instantiate cheyochu.

Examples:

```python
class Dog:
    def __init__(self, name):
        self.name = name

    def bark(self):
        return f"{self.name} says Woof!"

my_dog = Dog("Buddy")
```

Explanation: Dog class __init__ method use chesi name attribute set chestundi. bark method valla dog bark chese message return chestundi. my_dog anedi Dog class instance.

2. Inheritance

Inheritance use chesi existing class functionalities ni extend cheyochu.

Examples:

```python
class Animal:
    def speak(self):
        return "Animal sound"

class Dog(Animal):
    def bark(self):
        return "Woof!"
```

Explanation: Dog class Animal class ni inherit chestundi, so speak method ni Dog class lo kuda use cheyochu.

3. Encapsulation

Encapsulation valla class lo data hiding chesi, public methods through data access cheyochu.

Examples:

```python
class Person:
    def __init__(self, name):
        self.__name = name

    def get_name(self):
        return self.__name
```

Explanation: __name private attribute ga define chestundi, get_name method valla access cheyochu.

4. Polymorphism

Polymorphism valla same method different classes lo different implementations use cheyochu.

Examples:

```python
class Cat:
    def speak(self):
        return "Meow!"

class Dog:
    def speak(self):
        return "Woof!"

def make_speak(animal):
    print(animal.speak())
```

Explanation: Cat and Dog classes lo speak method different implementations chestayi. make_speak function polymorphism use chesi speak method call chestundi.

Topic 15: Web Scraping

Web scraping is the process of extracting data from websites. Python provides various libraries to facilitate this task.

1. Introduction to Web Scraping

Web scraping ante websites nundi data extract cheyadam. Python lo web scraping cheseydaniki chala libraries unnayi.

Examples:

- **Libraries:** BeautifulSoup, requests, Scrapy

Explanation: BeautifulSoup and requests use chesi simple scraping tasks perform cheyochu, Scrapy advanced scraping and crawling tasks ki use chestaru.

2. Using BeautifulSoup

BeautifulSoup use chesi web pages nundi data extract cheyadam chala easy. First requests library use chesi web page download cheyali, then BeautifulSoup use chesi parse cheyochu.

Examples:

```
import requests
from bs4 import BeautifulSoup

response = requests.get('https://example.com')
soup = BeautifulSoup(response.text, 'html.parser')

# Extracting title
title = soup.title.string
print(title)
```

Explanation: requests.get method use chesi web page download chestundi. BeautifulSoup use chesi HTML parse chesi data extract cheyochu.

3. Using Scrapy

Scrapy framework use chesi large scale scraping and data extraction cheseydam easy.

Examples:

- **Creating a Spider:**

```python
import scrapy

class MySpider(scrapy.Spider):
    name = 'myspider'
    start_urls = ['https://example.com']

    def parse(self, response):
        yield {
            'title': response.css('title::text').get(),
        }
```

Explanation: Scrapy framework lo spider create chesi, web pages nundi data extract cheyochu.

4. Handling Dynamic Content

Websites lo JavaScript use chesi dynamic content generate chestaru. Selenium or Playwright use chesi dynamic content handle cheyochu.

Examples:

```python
from selenium import webdriver

driver = webdriver.Chrome()
driver.get('https://example.com')
content = driver.page_source
driver.quit()
```

Explanation: Selenium use chesi dynamic content loaded page ni access chesi data extract cheyochu.

5. Ethics and Legal Considerations

Web scraping chesetappudu legal and ethical guidelines follow cheyyali. Websites robots.txt file lo scraping rules specify chestundi.

Examples:'/bsite allows scraping and what parts of the site are off-limits.

Explanation: Web scraping chesetappudu website owner rules follow cheyyadam and website ni overload cheyakunda scraping cheyyadam important.

Topic 16: Data Science

Data Science involves extracting insights and knowledge from data through various techniques and methods.

1. Introduction to Data Science

Data Science ante data nundi insights and knowledge extract cheyadam. It involves data collection, analysis, and visualization.

Examples:

- **Libraries:** Pandas, NumPy, Matplotlib, Seaborn

Explanation: Data manipulation and analysis ki Pandas and NumPy use chestaru, data visualization ki Matplotlib and Seaborn use chestaru.

2. Data Collection

Data collection chesetappudu various sources nundi data gather cheyyali – files, databases, web scraping, etc.

Examples:

- **Loading Data from CSV:**

```
import pandas as pd

df = pd.read_csv('data.csv')
```

Explanation: Pandas library use chesi CSV files nundi data load cheyochu.

3. Data Cleaning and Preparation

Raw data ni clean cheyali ante missing values handle cheyyali, duplicates remove cheyyali, and data types correct cheyyali.

Examples:

- **Handling Missing Values:**

```
df = df.dropna()  # Drop rows with missing values
```

Explanation: dropna() method use chesi missing values unna rows ni remove cheyochu.

4. Data Analysis

Data analysis lo statistical methods and techniques use chesi data patterns and trends identify cheyyali.

Examples:

- **Descriptive Statistics:**

```
summary = df.describe()
```

Explanation: describe() method use chesi data summary statistics generate cheyochu.

5. Data Visualization

Data visualization lo graphs and charts use chesi data ni visual form lo represent cheyyali.

Examples:

- **Creating a Plot:**

```
import matplotlib.pyplot as plt

df['column'].plot(kind='hist')
plt.show()
```

Explanation: Matplotlib use chesi histograms, bar charts, and other visualizations create cheyochu.

6. Machine Learning Basics

Machine Learning lo models train chesi data predictions and classifications cheyali.

Examples:

- **Simple Linear Regression:**

```
from sklearn.linear_model import LinearRegression

model = LinearRegression()
model.fit(X_train, y_train)
predictions = model.predict(X_test)
```

Explanation: scikit-learn use chesi machine learning models train and predictions generate cheyochu.

Topic 17: Logical Questions

Logical questions are designed to test your problem-solving and critical-thinking skills. They often involve puzzles, riddles, and scenarios that require logical reasoning.

1. Introduction to Logical Questions

Logical questions ante problem-solving and reasoning skills ni test cheyyadaniki use chese questions. Ivanni puzzles and scenarios lo solve cheyyali.

Examples:

- **Classic Puzzles:** "Two people are on opposite sides of a river. How can they both get to the other side?"

Explanation: Logical questions involve analyzing a problem and finding a solution through reasoning.

2. Types of Logical Questions

Logical questions ki various types unnayi:

- **Puzzles:** Problems that require creative thinking to solve.
- **Pattern Recognition:** Identifying patterns and sequences.
- **Syllogisms:** Drawing conclusions based on given premises.

Examples:

- **Pattern Recognition:** "What comes next in the sequence 2, 4, 8, 16?"

Explanation: Logical questions can involve numerical patterns, word puzzles, or hypothetical scenarios.

3. Problem-Solving Strategies

Logical questions solve cheyali ante effective strategies use cheyyali:

- **Break Down the Problem:** Divide the problem into smaller parts and analyze each part.
- **Look for Patterns:** Identify patterns or rules that can help in solving the problem.
- **Eliminate Impossible Solutions:** Narrow down the possibilities by excluding options that don’t fit.

Examples:

- **Eliminating Options:** "If a train travels from A to B in 2 hours, how long will it take to travel from B to A?"

Explanation: By eliminating or validating options, you can find the correct answer.

4. Practice and Application

Logical questions lo proficiency penchali ante regular practice chesi different types of problems solve cheyyali.

Examples:

- **Daily Practice:** Solve puzzles and logical problems regularly to improve skills.

Explanation: Regular practice helps in developing a strong problem-solving mindset and enhances logical reasoning abilities.