1.) Describe the difference between mutable and immutable types in Python. Provide examples.

Understanding Mutable :

mutable data types are which value can be changed after creation

List ,dictionaries and sets are mutable data types in Python

Eg code:

# List (mutable)

my\_list = [1, 2, 3]

print("Original List:", my\_list) # [1, 2, 3]

my\_list.append(4)

print("Modified List:", my\_list) # [1, 2, 3, 4]

Immutable Types :

in which data types modification cannot be done are caled immutable datatypes

tuples are immutable

Eg code:

# Tuple (immutable)

my\_tuple = (1, 2, 3)

print("Original Tuple:", my\_tuple) # (1, 2, 3)

# Trying to modify a tuple directly

try:

my\_tuple[0] = 10 # Attempt to change the first element

except TypeError as e:

print("Error:", e) # Will raise a TypeError

2.) Write a Python program to check if a number is a palindrome.

def is\_palindrome(string):

string = string.lower().replace(" ", "")

return string == string[::-1]

user\_input = input("Enter a word or phrase: ")

if is\_palindrome(user\_input):

print("It is a palindrome.")

else:

print("It is not a palindrome.")

"3.) What are Python decorators? Provide an example of how to use one. "

Answer: A decorator is a function that takes another function as input and extends or modifies its behavior without changing the original function.

Syntax:

@decorator\_name

def some\_function():

pass

COde:

def decorator(func):

return func

@decorator

def say\_hello():

print("Hello!")

say\_hello()

"4. Explain the purpose of \_\_init\_\_ in Python classes."

In Python, the \_\_init\_\_ method is like the setup function for an object when its attributes (its properties or features) right when it is made.

"5. What is the difference between a list and a tuple? When would you use each?"

|  |  |  |
| --- | --- | --- |
|  | List | Tuple |
| Syntax | my\_list = [1, 2, 3] | my\_tuple = (1, 2, 3) |
| Performance | Slower | Faster |
| Muttability | Yes | No |