Python Practice Worksheet 4

Focus: Expanded list/set comprehension, Functions (in depth), Lambda, Files, terminal I/O, Class basics.

# List and Set Comprehensions

1. Use a list comprehension to create a list of squares for the numbers 1 through 5.

a.

numbers = [1, 2, 3, 4, 5]

squares = [n\*\*2 for n in numbers]

print(squares) # Output: [1, 4, 9, 16, 25]

2. Create a set comprehension that extracts all unique first letters from the list ["apple", "banana", "avocado", "cherry"].

a.

fruits = ["apple", "banana", "avocado", "cherry"]

first\_letters = {word[0] for word in fruits}

print(first\_letters) # Output: {'a', 'b', 'c'}

3. Make a list comprehension that creates a list of all even numbers from 0 to 10.

a.

evens = [n for n in range(11) if n % 2 == 0]

print(evens) # Output: [0, 2, 4, 6, 8, 10]

# Functions in Depth

4. Write a function called greet that takes a name as an argument and prints "Hello, [name]!".

a.

def greet(name):

print(f"Hello, {name}!")

greet("Alex") # Output: Hello, Alex!

5. Write a function that takes a list of numbers and returns a new list with each number increased by 10.

a.

def add\_ten(numbers):

return [n + 10 for n in numbers]

print(add\_ten([1, 2, 3])) # Output: [11, 12, 13]

6. Write a function that returns True if a number is prime, and False otherwise.

a.

def is\_prime(n):

if n < 2:

return False

for i in range(2, int(n\*\*0.5) + 1):

if n % i == 0:

return False

return True

print(is\_prime(7)) # Output: True

print(is\_prime(8)) # Output: False

# Lambda Functions

7. Use a lambda function with map to multiply every number in the list [2, 4, 6] by 3.

a.

numbers = [2, 4, 6]

result = list(map(lambda x: x \* 3, numbers))

print(result) # Output: [6, 12, 18]

8. Use a lambda with filter to select all numbers greater than 5 from the list [3, 6, 9, 2, 8].

a.

numbers = [3, 6, 9, 2, 8]

filtered = list(filter(lambda x: x > 5, numbers))

print(filtered) # Output: [6, 9, 8]

9. Use a lambda to sort a list of tuples by the second value in each tuple: [(1, 3), (2, 2), (4, 1)].

a.

tuples = [(1, 3), (2, 2), (4, 1)]

sorted\_tuples = sorted(tuples, key=lambda x: x[1])

print(sorted\_tuples) # Output: [(4, 1), (2, 2), (1, 3)]

# Working with Files (with Statement)

10. Write Python code to save the list ["dog", "cat", "rabbit"] to a text file called animals.txt, one animal per line.

a.

animals = ["dog", "cat", "rabbit"]

with open("animals.txt", "w") as f:

for animal in animals:

f.write(animal + "\n")

11. Write code to read the file animals.txt and print each line, capitalized.

a.

with open("animals.txt") as f:

for line in f:

print(line.strip().capitalize())

12. Write code to append "hamster" to animals.txt and then read the file to confirm the change.

a.

with open("animals.txt", "a") as f:

f.write("hamster\n")

with open("animals.txt") as f:

for line in f:

print(line.strip())

# Terminal/Program File Inputs (input() Function)

13. Write a program that uses input() to ask the user for their favorite food, and then prints "Yum! I like [food], too."

a.

food = input("What is your favorite food? ")

print(f"Yum! I like {food}, too.")

14. Write a program that asks the user for two numbers and prints their sum.

a.

a = int(input("Enter a number: "))

b = int(input("Enter another number: "))

print("The sum is:", a + b)

15. Write a program that asks the user for their name and age, then prints "Hello [name], you are [age] years old."

a.

name = input("What is your name? ")

age = input("How old are you? ")

print(f"Hello {name}, you are {age} years old.")

# Basics of Python Classes

16. Define a class called Book with a title attribute and a method called display() that prints "Title: [title]".

a.

class Book:

def \_\_init\_\_(self, title):

self.title = title

def display(self):

print(f"Title: {self.title}")

mybook = Book("Python 101")

mybook.display() # Output: Title: Python 101

17. Create a class Person with attributes name and age, and write a method to print "Hi, I am [name] and I am [age] years old."

a.

class Person:

def \_\_init\_\_(self, name, age):

self.name = name

self.age = age

def introduce(self):

print(f"Hi, I am {self.name} and I am {self.age} years old.")

p = Person("Sam", 15)

p.introduce() # Output: Hi, I am Sam and I am 15 years old.

18. Create a class called Rectangle with width and height. Add a method area() that returns width \* height.

a.

class Rectangle:

def \_\_init\_\_(self, width, height):

self.width = width

self.height = height

def area(self):

return self.width \* self.height

rect = Rectangle(3, 4)

print(rect.area()) # Output: 12

# Challenges: Test on learning

Complete at least 2/3s of the challenge questions and put your completed work into a separate file to show me. Let the president or VP know which you will be completing before completing them please!

19. Combine what you've learned: Write a program that asks the user for a list of book titles (separated by commas), saves them to a file, then reads the file and creates a Book object for each title. Each Book should use its display() method to print the title.

20. Write a function that takes a filename and returns a set of all unique letters used in that file. Use set comprehension and file reading techniques.

21. Write a program that asks the user to input several numbers separated by spaces, creates a list of those numbers, and then uses a lambda function to filter out and print only the odd numbers.

# Reflection

22. Which activity did you find most exciting or challenging? Why?

23. What new skills or concepts stood out to you as especially useful?

24. How could you use these Python techniques in your own creative projects?