**Day 1 Notebook 1**

**Poem**

# Define a string with a poem using escape characters

poem = "Roses are red,\n\tViolets are blue,\n\"Escape characters\" in Python,\n\tMake formatting fun for you!"

# Print the formatted poem

print(poem)

**Poem with fstrings**

# Define variables with parts of the poem

line1 = "Roses are red,"

line2 = "Violets are blue,"

line3 = "Escape characters"

line4 = "Make formatting fun for you!"

# Use an f-string to format the poem with escape characters

poem = f"{line1}\n\t{line2}\n\"{line3}\" in Python,\n\t{line4}"

# Print the formatted poem

print(poem)

**Day 1 Notebook 2**

**Type Conversion**

# Given values for conversion

string\_num = "123"

integer\_num = 456

string\_float = "78.90"

float\_num = 12.34

# Perform conversions using built-in functions

s\_to\_i = int(string\_num) # String to Integer

i\_to\_s = str(integer\_num) # Integer to String

s\_to\_f = float(string\_float) # String to Float

f\_to\_s = str(float\_num) # Float to String

# Print results

print(f"String to Integer: '{string\_num}' -> {s\_to\_i}")

print(f"Integer to String: {integer\_num} -> '{i\_to\_s}'")

print(f"String to Float: '{string\_float}' -> {s\_to\_f}")

print(f"Float to String: {float\_num} -> '{f\_to\_s}'")

# Given values for conversion

string\_num = "123"

integer\_num = 456

string\_float = "78.90"

float\_num = 12.34

**Using sep and end Options**

# Perform conversions using built-in functions

s\_to\_i = int(string\_num) # String to Integer

i\_to\_s = str(integer\_num) # Integer to String

s\_to\_f = float(string\_float) # String to Float

f\_to\_s = str(float\_num) # Float to String

# Print results using print statements with sep and end options

print("String to Integer:", string\_num, "->", s\_to\_i, sep=" ", end="\n\n")

print("Integer to String:", integer\_num, "->", i\_to\_s, sep=" ", end="\n\n")

print("String to Float:", string\_float, "->", s\_to\_f, sep=" ", end="\n\n")

print("Float to String:", float\_num, "->", f\_to\_s, sep=" ", end="\n\n")

**miles per gallon**

#!/usr/bin/env python3

# calculate miles per gallon

print("The Miles Per Gallon program")

print()

# get input from the user

miles\_driven= float(input("Enter miles driven:\t\t"))

gallons\_used = float(input("Enter gallons of gas used:\t"))

# calculate and round miles per gallon

mpg = miles\_driven / gallons\_used

mpg = round(mpg, 2)

# display the result

print()

print("Miles Per Gallon:\t\t" + str(mpg))

**Day 1 Notebook 3**

**Relational Operators**

# Define two numerical variables

num1 = 25

num2 = 30

# Use relational operators and print the results directly

print(f"{num1} is equal to {num2}: {num1 == num2}")

print(f"{num1} is not equal to {num2}: {num1 != num2}")

print(f"{num1} is greater than {num2}: {num1 > num2}")

print(f"{num1} is less than {num2}: {num1 < num2}")

print(f"{num1} is greater than or equal to {num2}: {num1 >= num2}")

print(f"{num1} is less than or equal to {num2}: {num1 <= num2}")

**Logical Operators**

# Define three numerical variables

num1 = 25

num2 = 30

num3 = 20

# Using logical operators to evaluate conditions

if num1 < num2 and num1 < num3:

print(f"{num1} is the smallest number.")

if num2 > num1 or num2 > num3:

print(f"{num2} is greater than at least one of the other numbers.")

if not (num3 > num1):

print(f"{num3} is not greater than {num1}.")

if num1 < num2 and not (num2 < num3):

print(f"{num1} is less than {num2} and {num2} is not less than {num3}.")

**string comparisons**

# Define two string variables

string1 = "apple"

string2 = "banana"

# Use relational operators and print the results directly

print(f"'{string1}' is equal to '{string2}': {string1 == string2}")

print(f"'{string1}' is not equal to '{string2}': {string1 != string2}")

print(f"'{string1}' is greater than '{string2}': {string1 > string2}")

print(f"'{string1}' is less than '{string2}': {string1 < string2}")

print(f"'{string1}' is greater than or equal to '{string2}': {string1 >= string2}")

print(f"'{string1}' is less than or equal to '{string2}': {string1 <= string2}")

**if statements**

#!/usr/bin/env python3

# shapes.py

# prompt user for sides and display corresponding shape

# constants

SIDES\_LINE=1

SIDES\_POLYLINE=2

SIDES\_TRIANGLE=3

SIDES\_RECTANGLE=4

SIDES\_PENTAGON=5

# get user input and convert to an integer

sides = input("Please enter a number of sides from 1 to 5: ")

sides = int(sides)

# print the shape associated with that number of sides

# or print an error if the number is out of range

if sides == SIDES\_LINE:

    print("1 side is a line")

elif sides == SIDES\_POLYLINE:

    print("2 sides is a polyline")

elif sides == SIDES\_TRIANGLE:

    print("3 sides is a triangle")

elif sides == SIDES\_RECTANGLE:

    print("4 sides is a rectangle - and maybe a square")

elif sides == SIDES\_PENTAGON:

    print("5 sides is a pentagon")

else:

    print("that number is out of my range")

**nested if statements**

#!/usr/bin/env python3

# nested-if-discounts.py

#

customer\_type = "w"

invoice\_total = 125

if customer\_type.lower() == "r":

    if invoice\_total < 100:

        discount\_percent = 0

    elif invoice\_total >= 100 and invoice\_total < 250:

        discount\_percent = .1

    elif invoice\_total >= 250:

        discount\_percent = .2

elif customer\_type.lower() == "w":

    if invoice\_total < 500:

        discount\_percent = .4

    elif invoice\_total >= 500:

        discount\_percent = .5

else:

    discount\_percent = 0

print("Total cost with discount is " +

      str(invoice\_total - (invoice\_total \* discount\_percent)))

**ternary**

# Define a numerical variable

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

# Use a ternary operator to check if the number is positive, negative, or zero

result = "Positive" if num > 0 else "Negative" if num < 0 else "Zero"

# Print the result

print(f"The number is {result}.")

**match**

# Prompt the user to enter a command

command = input("Enter a command (start, stop, pause, resume): ")

# Use match statement for the command

match command:

    case "start":

        print("Starting the process...")

    case "stop":

        print("Stopping the process...")

    case "pause":

        print("Pausing the process...")

    case "resume":

        print("Resuming the process...")

    case \_:

        print("Unknown command")