Day 1 of Training at Ansh Info Tech

The first day of training at Ansh Info Tech covered the basics of Python programming with a focus on using Google Colab as the development environment. The session began with an introduction to Python's role in artificial intelligence and machine learning. Key Python concepts were then introduced, including variables, data types, and operators. The training also provided numerous practical examples and exercises to reinforce these concepts.

Topics Covered

Introduction to Python in AI:

- Overview of Python's significance in artificial intelligence and machine learning.
- Discussed how Python's simplicity and extensive libraries make it ideal for AI projects.

Introduction to Google Colab:

- Explained the features and benefits of using Google Colab for Python programming.
- Demonstrated how to set up and run Python code in Google Colab.

Variables in Python:

- Introduction to declaring and using variables.
- Examples included basic variable assignments and updates.

Data Types in Python:

- Covered the various data types available in Python: integers, floats, booleans, complex numbers, and tuples.
- Illustrated how to check data types using the type() function.

Operators in Python:

- Arithmetic Operators: Addition, subtraction, multiplication, division, floor division, modulus, and exponentiation.
- Relational Operators: Used for comparison between values.

Practical Exercises:

- A series of practice questions to apply the learned concepts. Examples included:
- Calculating the new population of lions in a zoo.
- Adjusting the temperature during a heatwave.

- Tracking the growth of a plant over a week.
- Determining the final velocity of a space mission.
- Distributing pizza slices among friends.
- Calculating the period of oscillation for a pendulum.
- Updating bug counts in a software codebase.
- Simulating gem collection in a game.
- Improving HRV index after a relaxation session.
- Modeling bacterial colony growth over a day.

The exercises provided practical scenarios for using Python to solve real-world problems, reinforcing the day's learning objectives.

```
print("Hello World")
a = 5
b = 7.5
c = True
d = 4.5j
e = 4,5
# if = 5
# print(if)
print(type(e))
print(type(d))
print(2+5) #Addition
print(5-5) #subtraction\
print(5*5) # Multiplicaton
print(5/2) # Division
print(5//2) #Floor Division
print(5%2) #Remainder
print(2**3) #Exponential
print(type(a))
print(type(b))
print(type(c))

print(2+10*(10+3))
wins = 6
draws = 1
losses = 2

print(wins*3+draws*1+losses*0)
#Precedence ->
# ()
# **
# *, /, %, //
# +, -

raisins = 50
almonds = 60
apricots = 20
total = raisins + almonds + apricots

print("Raisin Percentage: ", raisins/total*100)
print("Almond Percentage: "+ str(almonds/total*100))
print("Apricots Percentage: "+ str(apricots/total*100))



#Python Basics
# Variables, Operators, Google Colab Introduction,type()
```

```
Hello World
<class 'tuple'>
<class 'complex'>
7
0
25
2.5
2
1
8
<class 'int'>
<class 'float'>
<class 'bool'>
132
19
Raisin Percentage:  38.46153846153847
Almond Percentage: 46.15384615384615
Apricots Percentage: 15.384615384615385
```

Double-click (or enter) to edit

1. *You're creating a program to manage a zoo's animal population. Declare a variable lion_population with an initial value of 10. The zoo welcomes 5 new lion cubs. Update the lion_population variable and print the total lion population.*

```
lion_population = 10
lion_population += 5
print(lion_population)
```

```
15
```

You're developing a weather monitoring system. Declare a variable temperature with a value of 25.5 degrees Celsius. Due to a sudden heatwave, the temperature increases by 8 degrees. Update and print the new temperature.*

```
temp = 25.5
temp += 8
print(temp)
```

```
33.5
```

A science experiment involves tracking the growth of a plant. Declare a variable plant_height with an initial value of 15 centimeters. Over a week, the plant grows 2.5 centimeters taller each day. Update and print the final height of the plant after the week.

```
plant_height = 15
growth_each_day = 2.5
final_plant_height = plant_height + 7*growth_each_day
print(final_plant_height)
```

```
32.5
```

```
Start coding or generate with AI.
```

You're designing a space mission trajectory. Declare variables initial_velocity and acceleration with values 3000 meters per second and 500 meters per second squared respectively. Calculate and print the final velocity after 10 seconds.*

```
initial_velocity = 3000
acc = 500
time = 10
final_velocity = initial_velocity + acc*time
print(final_velocity)
```

```
8000
```

A group of friends is sharing a pizza. Declare a variable pizza_slices with a value of 8. Each friend wants to have an equal number of slices, and there are 5 friends. Calculate and print the maximum number of slices each friend can have without cutting the pizza.

```
pizza_slices = 8
num_friends = 5
max_slices_per_friend = pizza_slices // num_friends
print(max_slices_per_friend)
```

*You're modeling the movement of a pendulum. Declare a variable pendulum_length with a value of 1.2 meters. Calculate and print the period of oscillation (time taken for one complete swing) using the formula (T = 2pi * sqrt{{L}/{g}}), where pi (approximately 3.14159) and (g) is the acceleration due to gravity (approximately (9.81) meters per second squared).*

```
pendulum_length = 1.2  # meters
pi = 3.14159
gravity = 9.81  # meters per second squared
```

A software company is tracking the number of bugs in their codebase. Declare a variable bug_count with an initial value of 100. After a round of debugging, 35 bugs are fixed. Update and print the new bug_count.

```
bug_count = 100
bug_count -= 35
print(bug_count)
```

```
65
```

You're building a game where players collect gems. Declare a variable gem_count with an initial value of 50. Each time a player finds a gem, 5 gems are added to their collection. Update and print the new gem_count.

```
gem_count = 50
gem_collected = int(input("Enter the Number of Gems Collected: "))
new_gem_count = gem_count + gem_collected*5
print("New Gem Count: ", new_gem_count)
```

```
Enter the Number of Gems Collected: 5
New Gem Count:  75
```

A fitness tracker is monitoring a user's heart rate variability (HRV). Declare a variable hrv_index with an initial value of 80. After a relaxation session, the user's HRV improves by 10 points. Update and print the new hrv_index.

```
hrv_index = 80
hrv_index += 10
print(hrv_index)
```

```
90
```