**LESSON PLAN**

**Grade 8**

**Lesson 1: Introduction to Python**

**Lesson Objectives (What you will learn):**

* Understand what Python is
* Learn why we use it
* See how it’s used in real-world applications

**1. What is Python?**

* Python is a programming language used to make computers follow instructions.
* It helps us build games, apps, websites, and even control robots!
* It's known for being simple and easy to read — great for beginners.

Python works with:

* **Text/Commands** – to tell the computer what to do
* **Data** – to store names, numbers, answers
* **Logic** – to make decisions in a program

**2. Real Examples of Python**

* **Games** like Minecraft mods or simple 2D games
* **Calculators** and other small programs
* **Robotics** – controlling motors and sensors
* **Websites** like Instagram and YouTube use Python behind the scenes

**3. Try This Example**

print("Hello, Python!")

This line tells the computer to **print** the message.

Try changing the message inside the quotes.

**4. Exercises**

**Exercise 1: Change the Message**  
Change this part of the code to say something new:

print("Your message here!")

**Exercise 2: Say Your Name**  
Write a script that shows your name:

print("Hello, my name is [Your Name]!")

**Exercise 3: Ask for a Name**  
This script asks the user for their name and says hello:

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

print("Hello, " + name + "! Welcome to Python.")

**5. Lesson Summary**

* **Python** is a beginner-friendly programming language used in games, apps, websites, and robotics.

You learned how to:

* + Use print() to show a message
  + Use input() to ask the user for something
  + Join text and variables using +

**Quiz Questions**

**1. What is Python used for?**  
a) Drawing pictures  
b) Making computers do tasks  
c) Playing music  
d) Writing stories

**2. Which of the following is a real-world example of Python?**  
a) Making music in a band  
b) Typing in Microsoft Word  
c) Controlling a robot  
d) Posting on Facebook manually

**3. Which code is correct to show a message in Python?**  
a) echo("Hello!")  
b) say("Hello!")  
c) print("Hello!")  
d) message("Hello!")

**4. What does this Python code do?**

print("Welcome to Python!")

a) Opens a window  
b) Makes a sound  
c) Shows a message  
d) Does nothing

**5. How do you ask someone to enter their name in Python?**  
a) ask("Enter your name")  
b) prompt("Enter your name")  
c) input("Enter your name")  
d) name("Enter your name")

**Answer Key:**

1 – b  
2 – c  
3 – c  
4 – c  
5 – c

**Lesson 2: Simple Math and Input in Python**

**Recap of Lesson 1:**  
**Introduction to Python**

* Python is a simple programming language.
* Used in games, apps, websites, and robotics.
* You learned:  
  • print() – show messages  
  • input() – ask the user for input  
  • + – join text and variables

**Lesson Goals (What you will learn):**  
• Use math operations in Python  
• Get numbers from the user  
• Do basic calculations with input

**1. Math in Python**  
• + Add  
• - Subtract  
• \* Multiply  
• / Divide

**Example:**

python

Copy code

a = 8

b = 2

print("Sum:", a + b)

print("Divide:", a / b)

**2. Input with Numbers**  
When you ask for input, it’s a string. To do math, turn it into a number using int().

**Example:**

python

Copy code

num1 = int(input("Enter a number: "))

num2 = int(input("Enter another number: "))

print("Result:", num1 + num2)

**Try This**

# Try using Python as a calculator

print(10 + 5)

print(12 - 3)

print(4 \* 2)

print(16 / 4)

**Try with variables:**

x = 8

y = 2

print("Sum:", x + y)

print("Difference:", x - y)

print("Product:", x \* y)

print("Quotient:", x / y)

**And with user input:**

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

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

print("The total is", a + b)

**3. Exercises**  
**Exercise 1: Multiply Two Numbers**  
Ask the user for two numbers and multiply them.

Solution:

num1 = int(input("Enter a number: "))

num2 = int(input("Enter another number: "))

print("Result:", num1 \* num2)

**Exercise 2: Subtract and Show Result**  
Let the user enter two numbers. Subtract the second from the first and show the answer.

Solution:

num1 = int(input("Enter the first number: "))

num2 = int(input("Enter the second number: "))

print("Result:", num1 - num2)

**Gamification Challenge:**  
Ask the user for their test score out of 50. Show how many points they need to reach 50.  
Stars earned: 1 star for using input, 1 star for correct math!

Solution:

score = int(input("Enter your test score out of 50: "))

points\_needed = 50 - score

print("You need", points\_needed, "more points to reach 50.")

**Summary**  
• Use +, -, \*, / for math  
• Use int(input(...)) to get numbers  
• You can do calculations with user input

**Quiz**

1. What does \* do in Python?  
   a) Add  
   b) Divide  
   c) Multiply  
   d) Subtract
2. Which code asks for a number and adds 10?  
   a) input("number") + 10  
   b) int(input("Enter: ")) + 10  
   c) ask("number") + 10  
   d) print("number") + 10
3. What will this print?

python

Copy code

a = 5

b = 3

print(a - b)

a) 2  
b) 8  
c) a - b  
d) Error

1. What is the result of 10 / 2?  
   a) 5  
   b) 20  
   c) 0  
   d) 2

**Answer Key:**  
1 – c, 2 – b, 3 – a, 4 – a

**Lesson 3: Conditional Statements (if, elif, else)**

**Recap:**  
In the previous lesson, we learned how to perform basic math operations, and we saw how to ask the user for numbers using input(). We also used int() to convert input into numbers for calculations.

**Lesson Goals (What you will learn):**  
• Understand how to make decisions in Python using if, elif, and else  
• Learn how to compare values using logical operators  
• Practice writing simple conditional statements to control program flow

**1. What Are Conditional Statements?**  
Conditional statements let us make decisions based on conditions (true or false). They help us control the flow of a program.  
• if checks a condition  
• elif checks another condition if the first is false  
• else is used if all previous conditions are false

**Example:**

age = 18

if age >= 18:

print("You are an adult.")

else:

print("You are a minor.")

**2. Using elif for Multiple Conditions**  
You can check multiple conditions using elif. This helps if you want to make more than two decisions.

**Example:**

temperature = 25

if temperature > 30:

print("It's really hot outside.")

elif temperature > 20:

print("The weather is nice.")

else:

print("It's a bit cold.")

**3. Exercises**  
**Exercise 1: Age Checker**  
Write a program that checks if a person is a child (under 12), a teenager (12-17), or an adult (18+).

age = int(input("Enter your age: "))

if age < 12:

print("You are a child.")

elif age < 18:

print("You are a teenager.")

else:

print("You are an adult.")

**Exercise 2: Number Checker**  
Ask the user to enter a number. If the number is even, print "Even". If it's odd, print "Odd".

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

if num % 2 == 0:

print("Even")

else:

print("Odd")

**Gamification Challenge:**  
Ask the user for their age and give them a message based on their age group:

* If they are younger than 12, say "You are young!"
* If they are between 12 and 18, say "You are a teenager!"
* If they are older than 18, say "You are an adult!"  
  *Stars earned: 1 star for each correct condition used!*

**Summary**  
• if checks if a condition is true  
• elif checks another condition if the if is false  
• else runs if none of the conditions are true  
• We use conditional statements to make decisions in our program

**Quiz**

1. What does the if statement do in Python?  
   a) Checks a condition  
   b) Prints a message  
   c) Loops through a list  
   d) Asks for user input
2. What is the correct syntax for checking if a number is greater than 10?  
   a) if number > 10:  
   b) if number > 10  
   c) if (number > 10)  
   d) if number >= 10:
3. What will this code print?

age = 15

if age >= 18:

print("Adult")

else:

print("Minor")

a) Adult  
b) Minor  
c) Nothing  
d) Error

1. What does elif do in Python?  
   a) It checks another condition if the if condition is false  
   b) It checks if a condition is true  
   c) It is used to loop through a list  
   d) It runs code after all conditions
2. How do you check if a number is odd in Python?  
   a) if num % 2 == 0:  
   b) if num % 2 != 0:  
   c) if num / 2:  
   d) if num == 1:

**Answer Key:**  
1 – a, 2 – a, 3 – b, 4 – a, 5 – b

**LESSON PLAN**

**Grade 9**

**Lesson 1: Python Functions**

**Lesson Objectives (What you will learn):**

* Understand what a function is in Python
* Learn why we use functions
* See how functions are used in real life
* Write simple Python functions

**1️. What is a Python Function?**

* A function is a block of code that does something specific.
* Think of it like a **recipe**: you write the steps once, and use it many times.

**2️. Real-Life Examples of Functions**

* **Greeting Users** – Show a welcome message when someone opens a program
* **Math Calculators** – Add or multiply numbers
* **Menus** – Choose different actions (like in a game or app)
* **Robotics** – Run the same motor code again and again

**3️. Examples**

**Example 1 – Say Hello**

def say\_hello():

print("Hello, welcome to Python!")

say\_hello()

**Activity:** Change the message to: "Good morning, Grade 9 learners".

**Example 2 – Show a Custom Message**

def show\_message(message):

print(message)

show\_message("Welcome to Python!")

**Example 3 – Multiply Two Numbers**

def multiply\_numbers(a, b):

result = a \* b

print("The result is:", result)

multiply\_numbers(5, 6)

**4. Exercises**

**Exercise 1 – Add Two Numbers**

Create a function that adds 5 + 10 and shows the result.

def add\_numbers():

num1 = 5

num2 = 10

sum = num1 + num2

print(sum)

add\_numbers()

**Exercise 2 – Greet Someone by Name**

def greet(name):

print("Hello, " + name + "!")

greet("John")

**Gamification:**

**Challenge of the Day:**  
Create 3 functions about yourself:  
• introduce() – prints your name  
• favourite\_color() – prints your favorite color  
• hobby() – prints your favorite hobby

Earn 1 star for each function you complete correctly

**5. Lesson Summary**

* **What is a Function?**  
  A function is reusable code that performs a specific task.
* **Where Are Functions Used?**  
  To greet users, do calculations, handle repeated actions, etc.
* **Examples You Learned:**  
  Showing messages, multiplying numbers, and using parameters.

**Quiz: Python Functions**

**1. What is the primary purpose of a function in Python?**  
a) To draw images  
b) To reuse specific tasks in code  
c) To create websites  
d) To write stories

**2. Which of the following is a real-world example of using a Python function?**  
a) Printing your name once  
b) Repeating the same greeting in several places  
c) Saving a Word document  
d) Changing font size

**3. How do you define a function in Python?**  
a) function greet() {}  
b) def greet():  
c) greet = function()  
d) create function greet()

**4. What will this code do?**

def show\_message(message):

print(message)

show\_message("Welcome to Python!")

a) Multiply numbers  
b) Show a message  
c) Show an error  
d) Change background color

**5. What is the output of this code?**

def add():

print(5 + 10)

add()

a) 10  
b) 15  
c) Error  
d) Nothing

**Answer Key:**

1 – b  
2 – b  
3 – b  
4 – b  
5 – b

**Lesson 2: Parameters and Return Values**

**Lesson Goals (What you will learn)**

• Understand what **parameters** are in Python  
• Learn how to **pass values** to functions  
• Understand what a **return value** is  
• Write functions that return results

**Recap**

**What You Learned:**  
• A function is a reusable block of code  
• Functions start with the def keyword  
• You can call a function using its name  
• Some functions can accept **inputs** (parameters)

**Example Recap:**

def greet(name):

print("Hello, " + name + "!")

greet("John")

**1. What Are Parameters and Return Values?**

• **Parameters** are placeholders for values you pass to a function  
• **Arguments** are the actual values you pass when calling the function  
• **Return values** let a function send back a result

**Example: Function With Parameters**

def greet(name):

print("Hi " + name + "!")

You can now greet anyone by changing the argument.

**Example: Function That Returns a Value**

def add(a, b):

return a + b

result = add(5, 3)

print(result)

The function sends back the result using return, and we store it in result.

**2. Real-Life Examples of Return Values**

• A **calculator app** returns the result of your sum  
• A **game** might return your score after a level  
• A **form** might check if your input is correct and return "Yes" or "No"

**3. Examples**

**Example 1 – Add Two Numbers**

def add(a, b):

return a + b

print(add(4, 6)) # Outputs: 10

**Example 2 – Square a Number**

def square(n):

return n \* n

result = square(7)

print(result) # Outputs: 49

**Example 3 – Greet With a Return**

def get\_greeting(name):

return "Hello, " + name + "!"

message = get\_greeting("Grade 9")

print(message)

**4. Activities**

**Activity 1 – Add and Return** Create a function add\_numbers() that adds two numbers and returns the result. Store it in a variable and print it.

Solution:

def add\_numbers():

num1 = 10

num2 = 15

return num1 + num2

result = add\_numbers()

print(result) # Output: 25

**Activity 2 – Multiply and Store** Write a function multiply(x, y) that returns the product. Use it with print() to display the result.

Solution:

def multiply(x, y):

return x \* y

print(multiply(4, 5)) # Output: 20

**5. Exercises**

**Exercise 1 – Return a Welcome Message**

def welcome(name):

return "Welcome to class, " + name + "!"

Call it with your name and print the result.

Solution:

def welcome(name):

return "Welcome to class, " + name + "!"

message = welcome("Lerato")

print(message) # Output: Welcome to class, Lerato!

**Exercise 2 – Return the Square of a Number** Write a function square\_it(num) that returns the square of the number.

Solution:

def square\_it(num):

return num \* num

print(square\_it(9)) # Output: 81

**Gamification:**

**Challenge of the Day:**  
Write a function my\_info(name, age) that returns this sentence:  
**"My name is [name] and I am [age] years old."**

**Earn 3 stars:**  
• 1 star for correct function  
• 1 star for using return  
• 1 star for printing the result

**Lesson Summary**

• Functions can accept inputs called **parameters**  
• You pass values to those parameters when you call the function  
• Functions can **return** results using the return keyword  
• Returned values can be stored and used later

**Quiz:**

**1. What are parameters in a function?**  
A. The result of the function  
B. Values you get from input  
C. Placeholders in the function definition  
D. Comments in the code

**2. What does the return keyword do?**  
A. Repeats a function  
B. Sends back a value from the function  
C. Ends the program  
D. Adds numbers

**3. What is printed by this code?**

def add(a, b):

return a + b

print(add(2, 3))

A. a + b  
B. 23  
C. 5  
D. Nothing

**4. What happens if you don’t use return in a function?**  
A. The function won’t run  
B. It returns an error  
C. It returns nothing (None)  
D. It prints automatically

**5. Which of the following is correct?**  
A. def add(a + b):  
B. def add(a, b): return a + b  
C. add = def(a, b) return a + b  
D. def add() = a + b

**Answers**

1 – C  
2 – B  
3 – C  
4 – C  
5 – B

**Lesson 3: Conditional Statements (if, elif, else)**

**Lesson Goals (What you will learn)**

• Learn how to make decisions in code using if, elif, and else  
• Understand how comparisons work in Python  
• Practice writing functions that use conditions  
• Use real-life examples like age checks, test results, and game decisions

**Recap**

**What You Learned:**  
• Functions can take **parameters** and return **values**  
• return sends back a result  
• You can store returned results in variables

**Example Recap:**

def add(a, b):

return a + b

result = add(3, 7)

print(result) # Output: 10

**1. What Are Conditional Statements?**

• Conditional statements let your code **make decisions**  
• Use if, elif, and else to control what your program does  
• A condition checks if something is **True** or **False**

**2. Real-Life Examples of Conditions**

• If a student gets 50% or more, they pass  
• If it's raining, bring an umbrella  
• In a game, if your score is above 100, you level up  
• If a user is under 13, show a "too young" message

**3. Examples**

**Example 1 – Pass or Fail**

def check\_pass(mark):

if mark >= 50:

print("You passed!")

else:

print("You failed.")

**Example 2 – Age Checker**

def check\_age(age):

if age < 13:

print("You are too young.")

elif age < 18:

print("You're a teenager.")

else:

print("You're an adult.")

**Example 3 – Even or Odd**

def even\_or\_odd(number):

if number % 2 == 0:

print("Even")

else:

print("Odd")

**4. Activities**

**Activity 1 – Grade Checker**  
Write a function that takes a mark and:  
• If mark ≥ 80: print "Excellent"  
• If mark ≥ 50: print "Good job"  
• Else: print "Keep trying"

Solution:

def grade\_checker(mark):

if mark >= 80:

print("Excellent")

elif mark >= 50:

print("Good job")

else:

print("Keep trying")

grade\_checker(85) # Output: Excellent

grade\_checker(60) # Output: Good job

grade\_checker(40) # Output: Keep trying

**Activity 2 – Number Checker**  
Write a function that checks if a number is positive, negative, or zero, and prints the correct message.

Solution:

def check\_number(num):

if num > 0:

print("Positive")

elif num < 0:

print("Negative")

else:

print("Zero")

check\_number(5) # Output: Positive

check\_number(-2) # Output: Negative

check\_number(0) # Output: Zero

**5. Exercises**

**Exercise 1 – Check If a Person Can Vote**  
Write a function that checks if a person is 18 or older and prints "You can vote!" Otherwise, print "Too young to vote."

Solution:

def can\_vote(age):

if age >= 18:

print("You can vote!")

else:

print("Too young to vote.")

can\_vote(20) # Output: You can vote!

can\_vote(16) # Output: Too young to vote.

**Exercise 2 – Game Score Ranker**  
Create a function that takes a score and:  
• If score ≥ 100: print "Gold rank"  
• If score ≥ 50: print "Silver rank"  
• Else: print "Bronze rank"

Solution:

def rank\_score(score):

if score >= 100:

print("Gold rank")

elif score >= 50:

print("Silver rank")

else:

print("Bronze rank")

rank\_score(120) # Output: Gold rank

rank\_score(70) # Output: Silver rank

rank\_score(30) # Output: Bronze rank

**Gamification for this Lesson**

**Challenge of the Day:**  
Create a function student\_result(mark) that returns:  
• "Distinction" for 80 and above  
• "Pass" for 50–79  
• "Fail" for below 50

**Earn 3 stars:**  
• 1 star for using if  
• 1 star for using elif and else  
• 1 star for using return instead of print

**Lesson Summary**

• Use if, elif, and else to make choices in your code  
• Conditions compare values using operators like ==, >, <, >=, and <=  
• You can use conditionals inside functions to control what they do  
• Real programs use conditions all the time—games, apps, websites, and more!

**Quiz:**

**1. What does if do in Python?**  
A. Loops the code  
B. Runs code only when a condition is true  
C. Stores information  
D. Stops the program

**2. What is the output of this code?**

def test():

if 5 > 3:

print("Yes")

else:

print("No")

test()

A. Yes  
B. No  
C. Error  
D. Nothing

**3. Which keyword checks for the next condition?**  
A. else  
B. then  
C. elif  
D. check

**4. What does this code print?**

def check\_number(num):

if num == 0:

print("Zero")

else:

print("Not zero")

check\_number(0)

A. Zero  
B. Not zero  
C. Error  
D. Nothing

**5. What is wrong with this code?**

def check():

if age >= 18:

print("Adult")

A. print should be in all caps  
B. age is not defined  
C. if must be last  
D. def cannot be used here

**Answers**

1 – B  
2 – A  
3 – C  
4 – A  
5 – B

**LESSON PLAN**

**Grade 10**

**Lesson 1: Working with Lists and Dictionaries in Python**

**Lesson Objectives (What you will learn):**

* Understand what lists and dictionaries are in Python
* Learn how to store and manage data using these structures
* Practice accessing and modifying lists and dictionaries
* Use real-life examples to build programs with loops and data structures

**1. What is a List in Python?**

* A **list** is a collection of values stored in one variable.
* Lists are ordered and can be changed (mutable).

**Example:**

fruits = ["Apple", "Banana", "Cherry"]

print(fruits[1]) # Output: Banana

**Adding to a list:**

fruits.append("Orange")

print(fruits)

**2. What is a Dictionary in Python?**

* A **dictionary** stores data in **key-value** pairs.
* It’s like a mini-database: you label each piece of data with a name (key).

**Example:**

person = {

"name": "John",

"age": 30,

"job": "Developer"

}

print(person["name"]) # Output: John

**Adding a new key-value pair:**

person["hobby"] = "Reading"

**3. Real-Life Examples**

**Example 1 – List of Fruits**

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

fruits.append("orange")

for fruit in fruits:

print(fruit)

**Example 2 – Book Dictionary**

book = {

"title": "Python for Beginners",

"author": "Jane Doe",

"year": 2021

}

book["genre"] = "Programming"

for key, value in book.items():

print(key + ":", value)

**Example 3 – List of Student Dictionaries**

students = [

{"name": "John", "age": 16},

{"name": "Jane", "age": 15},

{"name": "Bob", "age": 17}

]

students.append({"name": "Alice", "age": 18})

for student in students:

print(student["name"], "is", student["age"], "years old.")

**4. Exercises**

**Exercise 1 – Color List**

Create a list of 5 colors. Add one more. Use a loop to display all colors.

colors = ["red", "blue", "green", "yellow", "orange"]

colors.append("purple")

for color in colors:

print(color)

**Exercise 2 – Favorite Movie Dictionary**

Create a dictionary with title, director, and year. Add genre. Print all details.

movie = {

"title": "Inception",

"director": "Christopher Nolan",

"year": 2010

}

movie["genre"] = "Science Fiction"

for key, value in movie.items():

print(key + ":", value)

**Gamification**

**Challenge of the Day:** Create a mini student database using a list of 3 dictionaries. Each dictionary should include:

* name
* age
* subject

**Earn 1 star** for each student you add!

**5. Lesson Summary**

* **List:** A collection of items. You can add more using .append() and loop through them.
* **Dictionary:** A group of labeled values using key-value pairs. You can add and access them using keys.

**Quiz: Lists and Dictionaries in Python**

**1. What is a list in Python?**  
a) A collection of key-value pairs  
b) A collection of unordered values  
c) A collection of ordered values  
d) A text message

**2. How do you access the first element in a list?**  
a) list[1]  
b) list(0)  
c) list.first()  
d) list[0]

**3. What is a dictionary in Python?**  
a) A function that performs a task  
b) A collection of key-value pairs  
c) A list of numbers  
d) A type of loop

**4. What does the append() method do?**  
a) Removes an item from a list  
b) Adds an item to the end of a list  
c) Updates a dictionary key  
d) Clears the list

**5. What will this code display?**

car = {

"make": "Tesla",

"model": "Model 3",

"year": 2021

}

car["color"] = "blue"

print(car["color"])

a) blue  
b) undefined  
c) Model 3  
d) Tesla

**6. How can you loop through a list of dictionaries?**  
a) Using a for loop  
b) Using while  
c) Using forEach()  
d) All of the above

**7. Which is true about lists in Python?**  
a) Lists are immutable  
b) Lists are indexed by strings  
c) Lists can store different data types  
d) Lists only store numbers

**8. What does this code print?**

colors = ["red", "blue"]

colors.append("green")

print(colors[2])

a) red  
b) blue  
c) green  
d) Error

**9. How do you access the value in a dictionary with key "name"?**  
a) dictionary(name)  
b) dictionary.name  
c) dictionary["name"]  
d) dictionary.get(name)

**10. What is the output of this code?**

students = [

{"name": "Alice", "age": 17},

{"name": "Bob", "age": 18}

]

students.append({"name": "Charlie", "age": 16})

print(students[2]["name"])

a) Charlie  
b) Bob  
c) Alice  
d) Error

**Answer Key:**

1 – c  
2 – d  
3 – b  
4 – b  
5 – a  
6 – a  
7 – c  
8 – c  
9 – c  
10 – a

**Lesson 2: Looping Through Lists and Dictionaries in Python**

**Lesson Goals (What you will learn)**

• Understand how to use for loops with lists and dictionaries  
• Learn how to access and manipulate list items using loops  
• Use loops to display and update dictionary contents  
• Apply loops to real-life programming situations

**Recap**

What You Learned:  
• Lists store ordered values  
• Dictionaries store labeled data in key-value pairs  
• You can use .append() to add to lists and square brackets [] to access dictionary values

**1. Looping Through a List**

You can use a for loop to go through each item in a list.

Example:

colors = ["red", "blue", "green"]

for color in colors:

print(color)

**2. Using range() and Indexes**

Use range() to loop through a list by index.

Example:

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

for i in range(len(fruits)):

print("Fruit", i + 1, "is", fruits[i])

**3. Looping Through a Dictionary**

Use .items() to loop through both keys and values.

Example:

person = {

"name": "Alice",

"age": 17,

"city": "Cape Town"

}

for key, value in person.items():

print(key + ":", value)

**4. Looping Through a List of Dictionaries**

This is useful when working with groups of people or objects.

Example:

students = [

{"name": "Lerato", "age": 16},

{"name": "Sipho", "age": 17},

{"name": "Thando", "age": 15}

]

for student in students:

print(student["name"], "is", student["age"], "years old.")

**5. Activities**

**Activity 1 – Looping Through a List of Cities**  
Create a list of 5 cities. Use a for loop to print:  
“I want to visit [city] one day.”

Solution:

cities = ["New York", "Paris", "Tokyo", "London", "Sydney"]

for city in cities:

print("I want to visit", city, "one day.")

**Activity 2 – Loop Through a Student Dictionary**  
Create a dictionary with your name, age, grade, and favorite subject. Use a loop to print each key and its value.

Solution:

student = {

"name": "Lara",

"age": 15,

"grade": "10th",

"favorite\_subject": "Math"

}

for key, value in student.items():

print(key + ":", value)

**6. Exercises**

**Exercise 1 – Multiplying Numbers in a List**  
Create a list of 5 numbers. Use a loop to multiply each number by 2 and print the result.

Solution:

numbers = [2, 4, 6, 8, 10]

for number in numbers:

print(number \* 2)

**Exercise 2 – Display Books from a List of Dictionaries**  
Create a list with 3 book dictionaries. Each should have a title and author. Use a loop to display:  
“[title] was written by [author]”

Solution:

books = [

{"title": "1984", "author": "George Orwell"},

{"title": "To Kill a Mockingbird", "author": "Harper Lee"},

{"title": "The Great Gatsby", "author": "F. Scott Fitzgerald"}

]

for book in books:

print(book["title"], "was written by", book["author"])

**7. Gamification:**

**Challenge of the Day:**  
Create a program that asks the user to enter 3 friends’ names, stores them in a list, and prints:  
“Hi, [name]!” for each friend.

* Earn 1 star for using a list
* Earn 1 star for using a loop

**8. Lesson Summary**

• Use for loops to go through lists and dictionaries  
• Use .items() to loop through key-value pairs in dictionaries  
• Use range() when you need index-based access  
• Loops help you repeat actions and work with data efficiently

**Quiz: Looping Through Lists and Dictionaries**

1. What does a for loop do in Python?  
   A. Repeats code  
   B. Creates variables  
   C. Stops the program  
   D. Deletes data
2. What does range(len(list)) allow you to do?  
   A. Add to the list  
   B. Loop with index numbers  
   C. Skip items  
   D. Break the list
3. Which method is used to loop through both keys and values of a dictionary?  
   A. .keys()  
   B. .values()  
   C. .items()  
   D. .all()
4. What is the output of this code?

names = ["Ava", "Ben", "Cara"]

for name in names:

print("Hi", name)

A. Error  
B. Hi names  
C. Hi Ava, Ben, Cara  
D. Hi Ava (then Hi Ben, Hi Cara on next lines)

1. What is the correct way to print all ages from this list?

people = [

{"name": "Neo", "age": 16},

{"name": "Lira", "age": 17}

]

A. print(people.age)  
B. for age in people["age"]  
C. for person in people: print(person["age"])  
D. for people in age: print(age)

**Answers**

**1. A**

**2. B**

**3. C**

**4. D**

**5. C**

**LESSON PLAN**

**Grade 10**  
**Lesson 3: Conditional Statements in Python**

**Lesson Objectives (What you will learn):**

* Understand what conditional statements are
* Learn how to use if, elif, and else to control program flow
* Practice writing conditions using comparisons
* Use real-world examples to make decisions in code

**Recap**

**What You Learned in Lesson 2:**

* How to create and use for loops to go through lists and dictionaries
* Looping through each item in a list
* Looping through key-value pairs in a dictionary

**1. What is a Conditional Statement?**

* A **conditional statement** lets your program make decisions.
* You can run different code depending on whether something is **true** or **false**.

**Keywords:**

* if: Runs a block of code if a condition is true
* elif: Checks another condition if the first is false
* else: Runs if none of the conditions are true

**Example:**

age = 16

if age >= 18:

print("You are an adult.")

else:

print("You are not an adult.")

**2. Comparison Operators**

| **Operator** | **Meaning** | **Example** |
| --- | --- | --- |
| == | Equal to | 5 == 5 |
| != | Not equal to | 5 != 3 |
| > | Greater than | 7 > 4 |
| < | Less than | 3 < 10 |
| >= | Greater or equal | 10 >= 9 |
| <= | Less or equal | 5 <= 5 |

**3. Real-Life Examples**

**Example 1 – Grade Checker**

grade = 75

if grade >= 50:

print("You passed!")

else:

print("You failed.")

**Example 2 – Temperature Alert**

temperature = 35

if temperature > 30:

print("It's a hot day!")

elif temperature > 20:

print("It's a nice day.")

else:

print("It's a bit cold.")

**Example 3 – Even or Odd Number**

number = 7

if number % 2 == 0:

print("Even number")

else:

print("Odd number")

**4. Activities**

**Activity 1 – Pass or Fail**  
Create a program that checks if a student score is 50 or more. Print “Pass” or “Fail”.

Solution:

score = int(input("Enter your score: "))

if score >= 50:

print("Pass")

else:

print("Fail")

**Activity 2 – Check Driving Age**  
Ask for someone’s age and check:

* If age >= 18: print "You can drive."
* Else: print "You are too young to drive."

Solution:

age = int(input("Enter your age: "))

if age >= 18:

print("You can drive.")

else:

print("You are too young to drive.")

**5. Exercises**

**Exercise 1 – Number Comparison**  
Ask the user for a number.  
Check if it's **greater than 10**, **equal to 10**, or **less than 10**, and print a message for each case.

Solution:

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

if number > 10:

print("The number is greater than 10.")

elif number == 10:

print("The number is equal to 10.")

else:

print("The number is less than 10.")

**Exercise 2 – Grade Feedback** Check the value of a grade and give feedback:

* 80 and above: “Excellent”
* 60–79: “Good”
* 50–59: “Fair”
* Below 50: “Fail”

Solution:

grade = int(input("Enter your grade: "))

if grade >= 80:

print("Excellent")

elif grade >= 60:

print("Good")

elif grade >= 50:

print("Fair")

else:

print("Fail")

**Gamification:**

**Challenge of the Day:**  
Create a program that asks the user to enter a number.  
If the number is:

* Even: print “Even number”
* Odd: print “Odd number”  
  (Bonus: Let them try again if input is not a number)

**Earn Stars:**  
• 1 star for using if/else  
• 1 star for checking even or odd

**Lesson Summary**

* Conditional statements use if, elif, and else to control program decisions
* Use comparison operators to test conditions
* You practiced checking grades, age, numbers, and more

**Quiz: Conditional Statements in Python**

**1. What keyword is used to start a conditional in Python?**  
A. for  
B. when  
C. if  
D. loop

**2. What does != mean in Python?**  
A. Is equal to  
B. Is not equal to  
C. Assignment  
D. Is divided by

**3. Which of these is a valid if statement?**  
A. if x == 5  
B. if (x = 5)  
C. if x !=  
D. if then x = 5

**4. What will this code output?**

x = 3

if x > 5:

print("Greater")

else:

print("Smaller or Equal")

A. Greater  
B. Error  
C. Smaller or Equal  
D. Nothing

**5. Which of these is NOT a comparison operator?**  
A. ==  
B. !=  
C. //  
D. <

**Answer Key:**

1 – C  
2 – B  
3 – A  
4 – C  
5 – C