1. **Print a simple message**

```
python
# Expected Output: Hello, World!
```

2. **Print multiple items in one statement**

```
python
# Example: Print your name, age, and city
# Expected Output: Name: John, Age: 25, City: New York
```

3. **Print using escape characters**

```
python
# Print a multi-line message using \n
# Expected Output:
# Welcome to Python!
# Let's learn coding.
```

4. **Print quotes inside a string**

```
python
# Expected Output: He said, "Python is amazing!"
```

5. **Print a pattern using stars**

```
python
# Expected Output:
# *
# **
# ***
# ****
# *****
```

□ Intermediate Output Exercises

6. **Formatted string output using f-strings**

```
python
# Take a name and age as variables and print:
# Hello, my name is John and I am 25 years old.
```

7. **Print a table**

```
python
# Print a 5x5 multiplication table in a clean format
```

8. **Align text with padding**

```
python
```

```
# Output three words in a column of width 10
# Example:
# Apple     Banana    Cherry
```

9. **Output the result of a simple calculation**

```
python
# Print: 5 + 7 = 12
```

10. **Print a formatted receipt**

```
python
# Use print statements to simulate a store receipt with:
# - Item names
# - Quantity
# - Price
# - Total
```

✅ Basic Keyword Exercises (1–10)

1. **Print a list of some common Python keywords.**
   (Manually type and print at least 10 keywords)
2. **Print three keywords used for conditional statements in Python.**
   (Expected: `if`, `else`, `elif`)
3. **Print three keywords used in loops.**
   (Expected: `for`, `while`, `break`, `continue`, etc.)
4. **Print the Boolean keywords in Python.**
   (Expected: `True`, `False`, `and`, `or`, `not`, `is`)
5. **Manually sort and print five keywords alphabetically.**
6. **Print the keywords that represent constant values.**
   (Expected: `True`, `False`, `None`)
7. **Print keywords used for flow control.**
   (Expected: `break`, `continue`, `pass`)
8. **Create a print statement that uses the word `if` as part of the text.**

   Example Output: `The keyword 'if' is used to check conditions.`

9. **Make a table using print that shows 2 columns: Keyword | Use**

   Example:

```
pgsql
CopyEdit
Keyword     | Use
----------- | ----------------------
if          | Conditional check
while       | Loop
break       | Exit loop
```

10. **Print all capital letter versions of 5 keywords.**

    e.g., `IF`, `ELSE`, `WHILE`, etc.

🧠 Intermediate Conceptual Challenges (11–15)

11. **Make a list (manually) of 10 Python keywords and print it.**

12. **Print 5 keywords that cannot be used as variable names.**

    Example: `if`, `while`, `for`, `class`, `def`

13. **Print a sentence that uses 3 keywords (in plain text).**

    Example Output: `Python uses if, else, and for to control the flow of code.`

14. **Group and print keywords into 3 categories: Conditional, Looping, Boolean.**

15. **Write a short explanation (using print) of why we can’t use keywords as variable names.**

    Example Output: `We can't use Python keywords as variable names because they have special meanings.`

1. **Take the user's name as input and print a welcome message.**
   Example:
   ```
   Input: John
   Output: Hello, John!
   ```

2. **Take two numbers as input and print their sum.**

3. **Ask the user for their favorite color and print it back.**

4. **Take the user's age and print: "You are X years old."**

5. **Ask the user for their city and country, then print: "You live in [City], [Country]."**

6. **Take two numbers as input and print their difference.**

7. **Take a word as input and print it in uppercase.**

8. **Take a sentence from the user and print how many characters it has.**

9. **Take your school name as input and print a message: "I study at [School Name]."**

10. **Take the user's first name and last name and print the full name.**

11. **Take the user's age and print how old they will be after 5 years.**

12. **Ask the user for two numbers and print their product.**

13. **Take a name and a hobby from the user and print a sentence: "Alex likes painting."**

1. **Convert a string `"10"` to an integer and add 5 to it.**

   Expected Output: `15`

2. **Take a float number (like `7.9`) as a string and convert it to a float.**

3. **Convert an integer `5` into a string and print `"The number is 5"` using string concatenation.**

4. **Convert a float `3.14` to an integer and print the result.**

5. **Convert an integer `1` to a boolean and print the result.**

   Hint: `bool(1)` returns `True`

6. **Convert the string `"False"` to a boolean. What happens?**

7. **Take a number from the user as a string, convert it to an integer, and multiply it by 2.**

8. **Take a floating-point number input, convert it to an integer, and print both the original and converted values.**

9. **Convert a number to a string and check its type using `type()`.**

---

11. **Take two numbers from the user and add them as integers (not strings).**

12. **Take a string input like `"25.5"` and convert it into both `float` and `int`. Print both results.**

13. **Convert a string like `"123abc"` to an integer. What error do you get?**

14. **Take a string input, check if it's numeric using `.isnumeric()`, and then convert to int if it is.**

15. **Convert a float (e.g., `9.99`) to int and then to string. Print each step.**

1. **Print a string literal that says "Learning Python is fun!"**
2. **Create and print an integer literal and a float literal.**

   Example: `10`, `3.14`

3. **Create a string literal using single quotes and another using double quotes. Print both.**
4. **Create a boolean literal and print it.**

   Example: `True`, `False`

5. **Create a variable with the value `None` and print it.**
6. **Print the type of the following literals using `type()`:**

- `"Hello"`
- `10`
- `3.5`
- `True`
- `None`

7. **Use a string literal to print: `Python's simplicity is powerful.`**
8. **Use string concatenation to join two string literals and print the result.**
9. **Create a list of mixed literals: a number, a string, and a boolean. Print it.**
10. **Print a multi-line string using triple quotes (string literal).**

12. **What is the output of: `print(None == 0)` and `print(None == False)`? Try and explain.**