Prepared by: **Dhananjay Kumar**
□□ Trainer | Java Full-Stack & Python | Contact: 7023577968
LinkedIn: Dhananjay Kumar

Introduction to Python Programming

1. Python Environment Setup

Python is an interpreted, high-level programming language. To run Python programs:

- Install Python (latest version recommended).
- Use IDEs like PyCharm, VS Code, or **Jupyter Notebook** for interactive coding.
- Run code in terminal with: `python filename.py`.

In Jupyter, you can run code cell-by-cell interactively.

---

2. Identifiers

- Identifiers are names used to identify variables, functions, classes, etc.
- Rules:
  - Must start with a letter (a–z, A–Z) or underscore (_).
  - Followed by letters, digits (0–9), or underscore.
  - Cannot be a reserved word.

```
# Example of valid identifiers
name = "Dhananjay"
_age = 28
rollNo123 = 101
print(name, _age, rollNo123)
```

---

3. Reserved Words

Python has reserved keywords that cannot be used as identifiers.

```
import keyword
print(keyword.kwlist)  # List of all Python reserved words
```

---

4. Lines and Indentation

- Python uses **indentation** instead of `{}` braces.
- Default indentation: 4 spaces.

```
if True:
    print("This is indented properly")
```

✖ Wrong:

```
if True:
print("This will cause IndentationError")
```

---

5. Multi-line Statements

- Use `\` for line continuation.
- Use parentheses `()`, brackets `[]`, braces `{}` to break lines naturally.

```
x = 10 + 20 + 30 + \
    40 + 50

nums = [1, 2, 3,
        4, 5, 6]
print(x, nums)
```

---

6. Comments

- **Single-line comment**: starts with `#`
- **Multi-line comment**: use triple quotes `'''` or `"""`

```
# This is a single-line comment
'''
This is a
multi-line comment
'''
```

---

7. Quotation

- Strings can be defined using `'`, `"`, or `'''`.

```
s1 = 'Hello'
s2 = "World"
s3 = '''This is
multi-line string'''
print(s1, s2, s3)
```

---

8. Input / Output

- Input: `input()` always returns **string**.
- Output: `print()`.

```
name = input("Enter your name: ")
age = int(input("Enter your age: "))
print("Hello", name, "You are", age, "years old")
```

---

9. Output Formatting

- **Using % formatting**
- **Using `format()` method**
- **Using f-strings (Python 3.6+)**

```
num = 123
text = "Python"

print("Number is %d" % num)
print("Number is {}".format(num))
print(f"Number is {num} and text is {text}")
```

---

10. Variables

a) Simple Variables

```
x = 10
y = "Hello"
print(x, y)
```

b) Assigning Multiple Values

```
a, b, c = 1, 2, 3
print(a, b, c)
```

c) One Value to Multiple Variables

```
x = y = z = 100
print(x, y, z)
```

d) Output Variables

```
name = "Dhananjay"
print("My name is " + name)
```

e) Global and Local Variables

```python
x = "Global Variable"

def myFunc():
    x = "Local Variable"
    print("Inside function:", x)

myFunc()
print("Outside function:", x)
```

Practice Questions with Solutions

Q1. Write a Python program to input two numbers and print their sum.

```python
a = int(input("Enter first number: "))
b = int(input("Enter second number: "))
print("Sum is:", a+b)
```

Q2. Show an example of multiple assignments.

```python
x, y, z = 5, 10, 15
print("Values:", x, y, z)
```

Q3. Demonstrate difference between local and global variable.

```python
x = "Global"

def test():
    x = "Local"
    print("Inside function:", x)

test()
print("Outside function:", x)
```

Q4. Write a program to format output using f-strings.

```python
name = input("Enter name: ")
age = int(input("Enter age: "))
print(f"Hello {name}, you are {age} years old")
```

Q5. Write a program to show usage of comments, indentation, and multi-line statement.

```python
# This program calculates total price
price = 100 + 200 + \
        300 + 400  # multi-line statement

if price > 500:
    print("High price")
else:
    print("Low price")
```

Student Practice Exercises

1. Write a program to input your name and print it in upper and lower case.
2. Assign one value to 3 variables and print them.
3. Print the list of all Python reserved words.
4. Show difference between single quotes, double quotes, and triple quotes.
5. Take user input for 3 numbers and print their average.
6. Write a program to check indentation error by removing spaces.
7. Create a global variable and modify it inside a function using `global` keyword.
8. Print numbers in formatted style: `Number: 005` (using formatting).
9. Show example of concatenating string with integer using `str()` conversion.
10. Write a program to demonstrate multi-line comments.

Summary

- Python uses indentation instead of braces.

- Identifiers follow naming rules, keywords are reserved.
- `input()` for taking user input, `print()` for output.
- Strings can be enclosed in single, double, or triple quotes.
- Variables can be global or local, multiple assignment is supported.
- Output formatting can be done with `%`, `format()`, or f-strings.

✅ Student Practice Exercises with Solutions

1. Write a program to input your name and print it in upper and lower case.

```
name = input("Enter your name: ")
print("Uppercase:", name.upper())
print("Lowercase:", name.lower())
```

---

2. Assign one value to 3 variables and print them.

```
x = y = z = 50
print(x, y, z)
```

---

3. Print the list of all Python reserved words.

```
import keyword
print(keyword.kwlist)
```

---

4. Show difference between single quotes, double quotes, and triple quotes.

```
s1 = 'Hello'
s2 = "World"
s3 = '''This is a
multi-line string'''
print(s1)
print(s2)
print(s3)
```

---

5. Take user input for 3 numbers and print their average.

```
a = float(input("Enter first number: "))
b = float(input("Enter second number: "))
c = float(input("Enter third number: "))

average = (a + b + c) / 3
print("Average is:", average)
```

---

6. Write a program to check indentation error by removing spaces.

❌ Wrong (this will cause error):

```
# if True:
# print("No indentation")
```

✅ Correct:

```
if True:
    print("Proper indentation")
```

---

7. Create a global variable and modify it inside a function using `global` keyword.

```
x = 10

def change():
    global x
    x = 20
    print("Inside function, x =", x)

change()
print("Outside function, x =", x)
```

---

8. Print numbers in formatted style: `Number: 005` (using formatting).

```
num = 5
print("Number: %03d" % num)  # using % formatting
print("Number: {:03d}".format(num))  # using format()
```

---

9. Show example of concatenating string with integer using `str()` conversion.

```
age = 25
print("I am " + str(age) + " years old")
```

---

10. Write a program to demonstrate multi-line comments.

```
'''
This is a multi-line comment.
It can be used to explain code in detail.
It will not be executed by Python.
'''

print("Program executed with multi-line comments!")
```

MCQs on Python Basics (Jupyter Notebook Format)

Below are **40 multiple-choice questions (MCQs)** with answers and explanations based on the introduction topics.

Q1. Which of the following is a valid identifier in Python?

A) 1name
B) _value
C) for
D) name@123

```
# Correct Answer: B) _value
# Explanation: Identifiers cannot start with digits, cannot use reserved words, and cannot contain special characters.
```

Q2. Which of the following is NOT a Python reserved keyword?

A) while
B) global
C) function
D) pass

```
# Correct Answer: C) function
# Explanation: 'function' is not a reserved keyword; 'def' is used for defining functions.
```

Q3. How does Python indicate a block of code?

A) Curly braces `{}`
B) Parentheses `()`
C) Indentation
D) Semicolon `;`

```
# Correct Answer: C) Indentation
```

Q4. Which of these will NOT raise an error?

A) x = 10 y = 20
B) x, y = 10, 20
C) x = y = 30
D) a, b, c = 1, 2

```
# Correct Answer: B and C
```

Q5. Which is used for multi-line comments in Python?

A) // comment
B) /* comment */
C) ''' comment '''
D)

 # Correct Answer: C

Q6. What is the output?

```
print("Number is %03d" % 7)
```

A) Number is 7
B) Number is 007
C) Number is 0007
D) Error

 # Correct Answer: B

Q7. What will be the output?

```
x = "Global"

def show():
    x = "Local"
    print("Inside:", x)

show()
print("Outside:", x)
```

A) Inside: Global, Outside: Global
B) Inside: Local, Outside: Global
C) Inside: Local, Outside: Local
D) Error

 # Correct Answer: B

Q8. What is the data type of `input()`?

A) int
B) float
C) str
D) depends on input

 # Correct Answer: C

Q9. Which is correct f-string usage?

A) print("Value of x = {x}")
B) print(f"Value of x = {x}")
C) print("f Value of x = {x}")
D) print(f'Value of x = x')

 # Correct Answer: B

Q10. Which statement is correct about Python?

A) Ignores indentation
B) Case-insensitive
C) Variables are created when assigned
D) Requires type declaration

 # Correct Answer: C

Q11. Which is a valid multi-line string?

A) s = 'Hello World

B) s = "Hello
C) s = '"Hello World"'
D) s = (Hello)

```
# Correct Answer: C
```

Q12. What is the output?

```
x = 5
print("Value:", x)
```

A) Error
B) Value: x
C) Value: 5
D) None

```
# Correct Answer: C
```

Q13. What will this print?

```
print(type(input("Enter something: ")))
```

A) <class 'int'>
B) <class 'str'>
C) depends
D) Error

```
# Correct Answer: B
```

Q14. Which is invalid assignment?

A) x, y = 1, 2
B) a = b = c = 10
C) 1x = 5
D) name = "John"

```
# Correct Answer: C
```

Q15. Which symbol is used for single-line comments?

A) //
B) #
C) <!--
D) ;

```
# Correct Answer: B
```

Q16. Which output formatting method is newest in Python?

A) % formatting
B) format()
C) f-strings
D) None

```
# Correct Answer: C
```

Q17. What is the result?

```
x, y = 5, 10
print(x, y)
```

A) 5 10
B) (5, 10)
C) Error
D) 10 5

```
# Correct Answer: A
```

Q18. Which is invalid identifier?

A) name1
B) _abc
C) while
D) roll_no

```
# Correct Answer: C
```

Q19. Which prints `007`?

A) print("%03d" % 7)
B) print("{:03d}".format(7))
C) print(f"{7:03d}")
D) All of the above

```
# Correct Answer: D
```

Q20. Which is correct?

```
x = 5
print("x = " + x)
```

A) Prints x = 5
B) Error
C) Prints x = x
D) None

```
# Correct Answer: B
# Need str(x) for concatenation
```

Q21. Which keyword is used to define a function?

A) function
B) def
C) fun
D) define

```
# Correct Answer: B
```

Q22. What is the output?

```
x = 100
y = 200
print(x, end=" ")
print(y)
```

A) 100200
B) 100 200
C) Error
D) None

```
# Correct Answer: B
```

Q23. What does `keyword.kwlist` return?

A) List of identifiers
B) List of keywords
C) List of variables
D) None

```
# Correct Answer: B
```

Q24. Which is correct for taking integer input?

A) x = input()
B) x = int(input())
C) x = str(input())

D) Both A and B

```
# Correct Answer: B
```

Q25. Which of the following prints multi-line string?

A) print('Hello\nWorld')
B) print("Hello\nWorld")
C) print('''Hello\nWorld''')
D) All

```
# Correct Answer: D
```

Q26. Which is correct output?

```
x = "10"
y = 20
print(x + str(y))
```

A) 30
B) 1020
C) Error
D) None

```
# Correct Answer: B
```

Q27. Which is global variable?

```
x = 5

def f():
    print(x)

f()
```

A) x inside function
B) x outside function
C) Both
D) None

```
# Correct Answer: B
```

Q28. Which keyword allows modifying global variable inside function?

A) global
B) local
C) public
D) static

```
# Correct Answer: A
```

Q29. What will this print?

```
x = 5
y = 10
x, y = y, x
print(x, y)
```

A) 5 10
B) 10 5
C) Error
D) None

```
# Correct Answer: B
```

Q30. Which is NOT correct string definition?

A) s = 'Hello'
B) s = "Hello"
C) s = '''Hello'''

D) s = Hello

```
# Correct Answer: D
```

Q31. Which character continues a statement in Python?

A) /
B) \
C) ;
D) ,

```
# Correct Answer: B
```

Q32. What is the result?

```
print("{} {}".format(10, 20))
```

A) 10 20
B) 20 10
C) Error
D) None

```
# Correct Answer: A
```

Q33. Which is correct way to assign multiple values?

A) a, b, c = 1, 2, 3
B) a = b = c = 1
C) a, b = 1, 2, 3
D) Both A and B

```
# Correct Answer: D
```

Q34. Which is invalid?

A) _x = 5
B) __y = 10
C) class = 20
D) var1 = 100

```
# Correct Answer: C
```

Q35. What is printed?

```
print(f"{5+5}")
```

A) {5+5}
B) 5+5
C) 10
D) Error

```
# Correct Answer: C
```

Q36. Which keyword defines local scope?

A) local
B) def
C) scope
D) None

```
# Correct Answer: B (functions create local scope)
```

Q37. Which function is used to display output?

A) display()
B) echo()
C) print()

D) show()

```
# Correct Answer: C
```

Q38. Which of the following is NOT allowed?

A) var = 10
B) VAR = 20
C) var = Var
D) var-name = 30

```
# Correct Answer: D (hyphen not allowed)
```

Q39. What is the default value of an uninitialized variable?

A) 0
B) None
C) Error
D) Undefined

```
# Correct Answer: C (variable must be assigned before use)
```

Q40. Which prints the average of 3 numbers correctly?

A) avg = a+b+c/3
B) avg = (a+b+c)/3
C) avg = (a+b)/c/3
D) avg = a+b+c//3

```
# Correct Answer: B
```

In [ ]: