Python Notes – Python Decision Making & Loops – Notes

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

Python Decision Making & Loops

1. If-Else Statement

- The `if` statement executes a block of code when the condition is **True**.
- The `else` block executes when the condition is **False**.

Syntax:

```
if condition:
    # code if true
else:
    # code if false
```

In [2]:
```
# Example: Check if a number is positive or negative
num = -5
if num >= 0:
    print("Positive number")
else:
    print("Negative number")
```

```
Negative number
```

2. Nested If-Else

- We can put an `if` statement inside another `if` or `else` block.

In [3]:
```
# Example: Check if a number is positive, negative or zero
num = 0
if num > 0:
    print("Positive")
else:
    if num < 0:
        print("Negative")
    else:
        print("Zero")
```

```
Zero
```

3. If-Elif-Else

- `elif` stands for **else if**
- Used when we have **multiple conditions**.

In [5]:
```
# Example: Find grade of a student
marks = 85
if marks >= 90:
    print("Grade A")
elif marks >= 75:
    print("Grade B")
elif marks >= 50:
    print("Grade C")
else:
    print("Fail")
```

```
Grade B
```

4. Match Statement (Python 3.10+)

- Used like a **switch-case** in other languages.

In [6]:
```
# Example: Simple calculator
operation = "+"
match operation:
    case "+":
```

```
        print("Addition")
    case "-":
        print("Subtraction")
    case "*":
        print("Multiplication")
    case _:
        print("Invalid Operation")
```

```
Addition
```

Loops in Python

5. While Loop

- Executes a block of code **as long as condition is True**.

In [7]:
```
# Example: Print numbers from 1 to 5
i = 1
while i <= 5:
    print(i)
    i += 1
```

```
1
2
3
4
5
```

6. For Loop

- Used to iterate over a sequence (list, tuple, string, range).

In [8]:
```
# Example: Print elements of a list
fruits = ["apple", "banana", "cherry"]
for fruit in fruits:
    print(fruit)
```

```
apple
banana
cherry
```

Q1. Write a program to check whether a number is even or odd.

In [9]:
```
num = 7
if num % 2 == 0:
    print("Even")
else:
    print("Odd")
```

```
Odd
```

Q2. Write a program to find the largest of three numbers using if-elif-else.

In [10]:
```
a, b, c = 10, 20, 15
if a >= b and a >= c:
    print("Largest:", a)
elif b >= a and b >= c:
    print("Largest:", b)
else:
    print("Largest:", c)
```

```
Largest: 20
```

Q3. Write a program using `while` loop to print the first 10 natural numbers.

In [11]:
```
i = 1
while i <= 10:
    print(i, end=" ")
    i += 1
```

```
1 2 3 4 5 6 7 8 9 10
```

Q4. Write a program using `for` loop to calculate the sum of numbers from 1 to 100.

In [12]:
```
total = 0
for i in range(1, 101):
```

```
    total += i
print("Sum of 1 to 100 =", total)
```

```
Sum of 1 to 100 = 5050
```

Q5. Use `match` statement to print day of week (1 → Monday, 7 → Sunday).

In [13]:
```
day = 3
match day:
    case 1:
        print("Monday")
    case 2:
        print("Tuesday")
    case 3:
        print("Wednesday")
    case 4:
        print("Thursday")
    case 5:
        print("Friday")
    case 6:
        print("Saturday")
    case 7:
        print("Sunday")
    case _:
        print("Invalid day")
```

```
Wednesday
```

Q6. Write a program to check if a given year is a leap year or not.

In [14]:
```
year = 2024
if (year % 400 == 0) or (year % 100 != 0 and year % 4 == 0):
    print(year, "is a Leap Year")
else:
    print(year, "is not a Leap Year")
```

```
2024 is a Leap Year
```

Q7. Write a program to check whether a number is positive, negative, or zero.

In [15]:
```
num = -8
if num > 0:
    print("Positive")
elif num < 0:
    print("Negative")
else:
    print("Zero")
```

```
Negative
```

Q8. Write a program using `match` statement to check vowels (a, e, i, o, u).

In [16]:
```
ch = "e"
match ch:
    case "a" | "e" | "i" | "o" | "u":
        print("Vowel")
    case _:
        print("Consonant")
```

```
Vowel
```

Q9. Write a program to print the multiplication table of a number using `while` loop.

In [17]:
```
num = 5
i = 1
while i <= 10:
    print(num, "x", i, "=", num * i)
    i += 1
```

```
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
```

Q10. Write a program to print the factorial of a number using `for` loop.

In [18]:
```
n = 5
fact = 1
for i in range(1, n + 1):
    fact *= i
print("Factorial of", n, "is", fact)
```

```
Factorial of 5 is 120
```

Q11. Write a program to reverse a number using while loop.

In [19]:
```
num = 1234
rev = 0
while num > 0:
    digit = num % 10
    rev = rev * 10 + digit
    num //= 10
print("Reversed number =", rev)
```

```
Reversed number = 4321
```

Q12. Write a program to count the number of digits in a number using while loop.

In [20]:
```
num = 98765
count = 0
while num > 0:
    count += 1
    num //= 10
print("Total digits =", count)
```

```
Total digits = 5
```

Q13. Write a program using for loop to print all even numbers between 1 and 50.

In [21]:
```
for i in range(2, 51, 2):
    print(i, end=" ")
```

```
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50
```

Q14. Write a program to calculate the sum of digits of a number using while loop.

In [22]:
```
num = 12345
s = 0
while num > 0:
    s += num % 10
    num //= 10
print("Sum of digits =", s)
```

```
Sum of digits = 15
```

Q15. Write a program to check if a number is prime using for loop.

In [24]:
```
num = 29
is_prime = True

if num <= 1:
    is_prime = False
else:
    for i in range(2, int(num ** 0.5) + 1):
        if num % i == 0:
            is_prime = False
            break

if is_prime:
    print(num, "is Prime")
else:
    print(num, "is Not Prime")
```

```
29 is Prime
```

Scenario-Based Questions & Solutions

Q1. Scenario: Check if a student passed or failed

You are a teacher. You want to write a program that asks the student for their marks.

- If marks are **greater than or equal to 40**, print "Pass".
- Otherwise, print "Fail".

In [25]:
```
# Solution
marks = int(input("Enter your marks: "))
if marks >= 40:
    print("Pass ✅")
else:
    print("Fail ❌")
```

```
Pass ✅
```

Q2. Scenario: Find the largest of two numbers

Imagine you are comparing the ages of two friends.
Write a program that takes **two numbers as input** and prints which one is larger.

In [26]:
```
# Solution
a = int(input("Enter first number: "))
b = int(input("Enter second number: "))

if a > b:
    print(a, "is greater")
elif b > a:
    print(b, "is greater")
else:
    print("Both are equal")
```

```
5 is greater
```

Q3. Scenario: Check voting eligibility

A government system needs to check if a person is eligible to vote.
Write a program that asks for **age**.

- If age is **18 or more** → Print "Eligible to vote".
- Otherwise → Print "Not eligible".

In [27]:
```
# Solution
age = int(input("Enter your age: "))
if age >= 18:
    print("✅ Eligible to vote")
else:
    print("❌ Not eligible to vote")
```

```
❌ Not eligible to vote
```

Q4. Scenario: Number lies in a range

You are building a game where the winning number is always between 10 and 50.
Write a program that takes a number from the user and checks:

- If the number is **between 10 and 50**, print "You are in the range".
- Otherwise, print "Out of range".

In [28]:
```
# Solution
num = int(input("Enter a number: "))
if num >= 10 and num <= 50:
    print(" You are in the range!")
else:
    print("❌ Out of range")
```

```
❌ Out of range
```

Q5. Scenario: Day of the week

Suppose you are creating a digital calendar.
The user enters a number (1–7). Print the corresponding day of the week using a `match` statement.

In [29]:
```
# Solution
day = int(input("Enter a number (1-7): "))

match day:
    case 1:
        print("Monday")
```

```
    case 2:
        print("Tuesday")
    case 3:
        print("Wednesday")
    case 4:
        print("Thursday")
    case 5:
        print("Friday")
    case 6:
        print("Saturday")
    case 7:
        print("Sunday")
    case _:
        print("Invalid input ✖")
```

```
Friday
```

Q6. Scenario: Print first N natural numbers

A teacher wants to print the first `N` natural numbers (1, 2, 3, … N).
Write a program using a `while` loop.

In [30]:
```
# Solution
n = int(input("Enter the value of N: "))
i = 1
while i <= n:
    print(i, end=" ")
    i += 1
```

```
1 2 3 4 5 6 7 8 9
```

Q7. Scenario: Sum of first N numbers

You are writing a program for a bank that needs to calculate the sum of the first `N` numbers.
For example, if `N = 5`, the sum is `1+2+3+4+5 = 15`.

In [31]:
```
# Solution
n = int(input("Enter the value of N: "))
total = 0
for i in range(1, n+1):
    total += i
print("The sum of first", n, "numbers is:", total)
```

```
The sum of first 7 numbers is: 28
```

Q8. Scenario: Multiplication table

A school project requires printing the multiplication table of a given number.
Example: If the input is `5`, print:

5 x 1 = 5

5 x 2 = 10

... up to 10

In [33]:
```
# Solution
num = int(input("Enter a number: "))
for i in range(1, 11):
    print(num, "x", i, "=", num * i)
```

```
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
3 x 10 = 30
```

Q9. Scenario: Reverse a number

You are building a mobile app that reverses numbers.
If input is `1234`, output should be `4321`.

In [35]:
```python
# Solution
num = int(input("Enter a number: "))
rev = 0
while num > 0:
    digit = num % 10
    rev = rev * 10 + digit
    num //= 10
print("Reversed number:", rev)
```

```
Reversed number: 321
```

Q10. Scenario: Simple Grade System

A university has a grading system:

- Marks >= 90 → Grade A
- Marks >= 75 → Grade B
- Marks >= 50 → Grade C
- Else → Fail

Write a program to calculate grade based on marks.

In [36]:
```python
# Solution
marks = int(input("Enter your marks: "))

if marks >= 90:
    print("Grade A ")
elif marks >= 75:
    print("Grade B ")
elif marks >= 50:
    print("Grade C ")
else:
    print("Fail ✖")
```

```
Fail ✖
```

Python Decision Making & Loops – MCQs

In [37]:
```
MCQ ="""
1. Which of the following is the correct syntax for an if statement in Python?
a) if (x > y) then:
b) if x > y:
c) if x > y then:
d) if (x > y): then

Answer: b) if x > y:

2. What will be the output of the following code?

x = 10
if x > 5:
    print("Hello")


a) Error
b) Nothing will print
c) Hello
d) 5

Answer: c) Hello


3. Which keyword is used when we have multiple conditions?
a) elseif
b) elif
c) else if
d) end if

Answer: b) elif


4. What will be the output?

x = 7
if x % 2 == 0:
    print("Even")
else:
    print("Odd")
```

```
a) Even
b) Odd
c) Error
d) Nothing

Answer: b) Odd



5. Nested if statements mean:
a) if inside another if
b) if followed by else
c) if without else
d) if with elif

Answer: a) if inside another if


6. Which Python keyword is used for the default case in match statement?
a) else
b) default
c) case _
d) pass

 Answer: c) case _



 7. What will be the output?

num = 3
match num:
    case 1:
        print("One")
    case 2:
        print("Two")
    case _:
        print("Other")


a) One
b) Two
c) Other
d) Error

 Answer: c) Other

8. Which loop runs until a condition becomes False?
a) for
b) while
c) do-while
d) repeat

Answer: b) while



9. What will be the output?

i = 1
while i < 4:
    print(i)
    i += 1


a) 1 2 3
b) 0 1 2
c) 1 2 3 4
d) Infinite loop

 Answer: a) 1 2 3


10. Which function is used to generate a sequence in a for loop?
a) range()
b) seq()
c) list()
d) loop()

Answer: a) range()


 11. What will be the output?
```

```
for i in range(3):
    print(i)
```

a) 1 2 3
b) 0 1 2
c) 0 1 2 3
d) Error

Answer: b) 0 1 2

12. What does break statement do in loops?
a) Skips current iteration
b) Stops the loop immediately
c) Restarts the loop
d) Does nothing

Answer: b) Stops the loop immediately

13. What does continue statement do in loops?
a) Skips rest of the code in current iteration
b) Stops the loop
c) Repeats the loop twice
d) Goes to else

Answer: a) Skips rest of the code in current iteration

14. Which of these is not a valid loop in Python?
a) while
b) for
c) do-while
d) nested for

Answer: c) do-while

15. What will be the output?

```
for i in range(5, 8):
    print(i)
```

a) 5 6 7
b) 5 6 7 8
c) 6 7 8
d) Error

Answer: a) 5 6 7

16. What is the output?

```
for i in range(2, 10, 3):
    print(i)
```

a) 2 5 8
b) 2 3 4
c) 3 6 9
d) 2 5 7

Answer: a) 2 5 8
Step = 3 → increments by 3.

17. In Python, while True: means
a) The loop never runs
b) Runs once only
c) Infinite loop
d) Syntax error

Answer: c) Infinite loop

18. Which of these is correct?
a) for i = 0; i < 5; i++
b) for i in range(5):
c) for (i in range(5)):

```
d) loop i in 5

Answer: b) for i in range(5):


19. Which operator is used to check equality in Python?
a) =
b) ==
c) :=
d) !=

 Answer: b) ==


 20. Which operator is used to check inequality?
a) <>
b) =!
c) !=
d) not

 Answer: c) !=


 21. What will this print?

x = 15
if x > 10 and x < 20:
    print("Yes")


a) Yes
b) No
c) Error
d) Nothing

Answer: a) Yes


22. What is the output?

name = "Python"
if "y" in name:
    print("Found")


a) Found
b) Not Found
c) Error
d) Nothing

 Answer: a) Found


 23. What is the result of 5 % 2?
a) 2
b) 1
c) 0
d) 2.5

Answer: b) 1


24. Which keyword ends an if block in Python?
a) endif
b) end
c) no keyword needed
d) stop

Answer: c) no keyword needed


25. Which loop is better when we don’t know how many times to run?
a) for
b) while
c) if
d) match

Answer: b) while


26. What will happen?

i = 1
```

```
while i < 3:
    print("Hi")


a) Hi Hi
b) Hi Hi Hi
c) Infinite Hi
d) Error

Answer: c) Infinite Hi
 Since i is never incremented.

27. What is the output?

for ch in "Hi":
    print(ch)


a) Hi
b) H i
c) Error
d) Nothing

 Answer: b) H i
 Loops through characters in a string.

28. Which of these is a valid match syntax?
a) match(x) { case 1: … }
b) match x: case 1: …
c) match(x): case 1: …
d) case x in match:

 Answer: b) match x: case 1: …

29. What will be the output?

for i in range(1, 6):
    if i == 3:
        break
    print(i)


a) 1 2 3 4 5
b) 1 2 3
c) 1 2
d) 3 4 5

Answer: c) 1 2
 Loop stops when i=3.

30. What will be the output?

for i in range(1, 6):
    if i == 3:
        continue
    print(i)


a) 1 2 3 4 5
b) 1 2 4 5
c) 2 3 4 5
d) 1 2 5

 Answer: b) 1 2 4 5
continue skips i=3.

"""
```

In [ ]: