

PYTHON TEST - 1.9 (STATEMENT FLOW CONTROL)

Total points 50/50 ?

Statement Flow Control in Python

STUDENT NAME *

VIVA

✓ 1. Which keyword is used for conditional branching in Python? *

1/1

- a) loop
- b) if
- c) case
- d) switch



✓ 2. What will be the output of if $5 > 2$: `print("Yes")`? *

1/1

- a) No output
- b) Yes
- c) Error
- d) $5 > 2$



✓ 3. Which keyword is used when multiple conditions are checked after if? * 1/1

- a) elseif
- b) elif
- c) else-if
- d) case



✓ 4. In Python, the else block is executed when: *

1/1

- a) If condition is true
- b) If condition is false
- c) Always
- d) Never



✓ 5. What will be the output? *

1/1

```
x = 10
```

```
if x > 15:
```

```
    print("A")
```

```
elif x > 5:
```

```
    print("B")
```

```
else:
```

```
    print("C")
```

a) A

b) B

c) C

d) None



✓ 6. Which is correct syntax for nested if? *

1/1

a) if x: if y: print("Hello")

b) if x then if y: print("Hello")

c) if (x && y): print("Hello")

d) None



✓ 7. Nested if statements in Python are: *

1/1

- a) Not allowed
- b) Allowed
- c) Only in loops
- d) Deprecated



✓ 8. What will be the output? *

1/1

```
x = 20
```

```
if x > 10:
```

```
    if x > 15:
```

```
        print("Yes")
```

- a) Yes
- b) No output
- c) Error
- d) 20



✓ 9. Which keyword is used for iteration in Python? *

1/1

- a) for
- b) loop
- c) repeat
- d) iterate



✓ 10. Which loop runs until condition is true? *

1/1

- a) for
- b) while
- c) do-while
- d) until



✓ 11. How many times will this loop run? *
for i in range(5):

1/1

print(i)

- a) 6
- b) 4
- c) 5
- d) Infinite



✓ 12. What is the output? *

1/1

```
i = 1
```

```
while i < 4:
```

```
    print(i)
```

```
    i += 1
```

a) 1 2 3

✓

b) 1 2 3 4

c) 1 2

d) Infinite loop

✓ 13. The function range(2, 10, 2) generates: *

1/1

a) 2,4,6,8

✓

b) 2,3,4,5,6,7,8,9

c) 10,8,6,4,2

d) Error

✓ 14. Which statement exits from a loop immediately? *

1/1

a) continue

b) break

✓

c) exit

d) stop

✓ 15. Which statement skips the current iteration and moves to next? * 1/1

- a) stop
- b) pass
- c) continue
- d) break



✓ 16. Which statement is used as a null operation * 1/1

- a) break
- b) continue
- c) skip
- d) pass



✓ 17. What will be the output? * 1/1

```
for i in range(5):
```

```
    if i == 3:
```

```
        break
```

```
    print(i)
```

- a) 0 1 2 3 4
- b) 0 1 2
- c) 0 1 2 3
- d) 3 4



✓ 18. Output?for i in range(5): *

1/1

```
if i == 2:  
    continue  
print(i)
```

- a) 0 1 2 3 4
- b) 0 1 3 4
- c) 0 1 4
- d) 2 3 4

✓

✓ 19. Output? * *

1/1

```
for i in range(3):  
    pass  
print("Done")
```

- a) Done
- b) Error
- c) Nothing
- d) 0 1 2

✓

✓ 20. Python allows else with: *

1/1

- a) if only
- b) while and for
- c) switch
- d) None



✓ 21. What will be the output? *

1/1

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

```
    print(i)
```

```
else:
```

```
    print("End")
```

- a) 0 1 2
- b) 0 1 2 End
- c) End only
- d) Error



✓ 22. In loops, else executes when: *

1/1

- a) Loop breaks
- b) Loop completes normally
- c) Both
- d) None



✓ 23. Output? *

1/1

```
for i in range(5):
```

```
    if i == 2:
```

```
        break
```

```
    else:
```

```
        print("No Break")
```

a) No Break

b) Nothing

c) Error

d) 0 1 2

✓

✓ 24. match-case is similar to: *

1/1

a) if-else

b) switch-case

c) loop

d) break

✓

✓ 25. Correct syntax of match-case: *

1/1

a) match x: case 1: print("One")

b) switch(x) { case 1: print("One") }

c) match-case(x): print("One")

d) None

✓



✓ 26. Output? *

1/1

x = 2

match x:

case 1:

print("One")

case 2:

print("Two")

case _:

print("Other")

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

✓

✓ 27. In match-case, _ represents: *

1/1

- a) Error
- b) Default case
- c) Null
- d) Break

✓

✓ 28. Which is NOT a flow control statement? *

1/1

- a) if
- b) for
- c) def
- d) while

✓

✓ 29. Indentation in Python is important for: *

1/1

- a) Loop only
- b) if only
- c) Defining block of code
- d) Variables

✓

✓ 30. Infinite loop happens when: *

1/1

- a) Condition always true
- b) Condition always false
- c) Break used
- d) Continue used

✓

✓ 31. Which is valid? *

1/1

- a) while True: break
- b) while True: continue
- c) while False: print("Hi")
- d) All



✓ 32. Which is used to exit from entire program? *

1/1

- a) stop()
- b) sys.exit()
- c) exit_loop()
- d) quit_loop()



✓ 33. What is the output? *

1/1

for i in "ABC":

print(i)

- a) A B C
- b) ABC
- c) Error
- d) None



✓ 34. Can for loop iterate over a list? *

1/1

a) Yes



b) No

✓ 35. Which loop is entry controlled? *

1/1

a) for

b) while

c) Both



d) None

✓ 36. Which keyword creates an infinite loop easily? *

1/1

a) while True



b) for True

c) loop forever

d) repeat



✓ 37. Output? *

1/1

```
x = 0
```

```
while x < 3:
```

```
    print(x)
```

```
    x += 1
```

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

✓

✓ 38. Which is true about Python loops? *

1/1

- a) for can iterate over sequences
- b) while works with condition
- c) Both a & b
- d) None

✓

✓ 39. Output? *

1/1

```
for i in range(2, 5):
```

```
    print(i)
```

a) 2 3 4

✓

b) 2 3 4 5

c) 3 4 5

d) 5

✓ 40. Which can be used to stop execution of program in middle? *

1/1

a) exit()

✓

b) break

c) continue

d) None

✓ 41. What will be the output? *

1/1

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

```
    print("Hi")
```

```
else:
```

```
    print("Bye")
```

- a) Hi Hi Hi
- b) Hi Hi Hi Bye
- c) Bye only
- d) Error

✓

✓ 42. Which one is invalid? *

1/1

- a) if True: print("Yes")
- b) if (5 > 3): print("Yes")
- c) if 5 > 3 print("Yes")
- d) if 5 > 3: print("Yes")

✓

✓ 43. continue statement is used in: *

1/1

- a) if only
- b) loops
- c) functions
- d) modules

✓

✓ 44. Which loop can be replaced by recursion? *

1/1

- a) for
- b) while
- c) Both
- d) None

✓

✓ 45. A flow control statement decides: *

1/1

- a) Execution order of program
- b) Data type
- c) Variable value
- d) None

✓

✓ 46. Output? *

1/1

```
x = 5
```

```
if x % 2 == 0:
```

```
    print("Even")
```

```
else:
```

```
    print("Odd")
```

- a) Even
- b) Odd
- c) Error
- d) None

✓



✓ 47. In Python, infinite loops can be dangerous because: *

1/1

- a) Program never stops
- b) High CPU usage
- c) Both
- d) None

✓

✓ 48. Which is faster for iteration? *

1/1

- a) for over list
- b) while with index

✓

✓ 49. Which is not a loop? *

1/1

- a) for
- b) while
- c) do-while
- d) None

✓

✓ 50. Which keyword is used to return control to caller? *

1/1

a) return



b) break

c) continue

d) pass

This content is neither created nor endorsed by Google. - [Contact form owner](#) - [Terms of Service](#) - [Privacy Policy](#).

Google Forms



