

PYTHON TEST - 3.6 (RETURNING VALUES)

Total points 50/50 ?

Returning Values from Functions in Python

STUDENT NAME *

VIVA

✓ 1. What keyword is used to return a value from a Python function? * 1/1

- a) yield
- b) return
- c) break
- d) pass



✓ 2. What does a function return if no return statement is used? *

1/1

- a) 0
- b) None
- c) Error
- d) Empty string



✓ 3. What will be the output of: *

1/1

```
def func():  
    return  
print(func())
```

- a) Error
- b) 0
- c) None
- d) Blank line



✓ 4. Can a Python function return multiple values? *

1/1

- a) No
- b) Yes, using tuples
- c) Yes, using lists
- d) Yes, using both tuples and lists



✓ 5. What is returned if multiple values are separated by commas in a return statement? *1/1

- a) List
- b) Set
- c) Tuple
- d) Dictionary



✓ 6. Output of: * 1/1
def calc(a, b):

```
    return a+b, a-b
```

```
x = calc(5, 2)
```

```
print(x)
```

- a) [7, 3]
- b) (7, 3)
- c) 7, 3
- d) Error



✓ 7. Can return appear more than once in a function? * 1/1

- a) Yes
- b) No



✓ 8. What happens if a return is encountered in a function? *

1/1

- a) Function execution stops immediately
- b) Function continues after return
- c) Function restarts
- d) None

✓

✓ 9. A function can return: *

1/1

- a) Only one value
- b) Only one object
- c) Any object, including multiple values as a tuple
- d) Only strings

✓

✓ 10. Which type does Python implicitly return if no return? *

1/1

- a) int
- b) str
- c) NoneType
- d) float

✓

✓ 11. What will `type(func())` be in Q3? *

1/1

- a) int
- b) NoneType
- c) str
- d) tuple



✓ 12. Output of: *

1/1

```
def f():  
    return [1,2,3]  
print(type(f()))
```

- a) list
- b) tuple
- c) int
- d) set



✓ 13. Output of: *

1/1

```
def f():
```

```
    return "Hello"
```

```
print(f())
```

- a) None
- b) Hello
- c) "Hello"
- d) Error



✓ 14. Output of: *

1/1

```
def f():
```

```
    return 5+3
```

```
print(f())
```

- a) 53
- b) 8
- c) (5,3)
- d) Error



✓ 15. Output of: *

1/1

```
def f():
```

```
    return "Python", 3.11
```

```
print(f())
```

- a) [Python, 3.11]
- b) ("Python", 3.11)
- c) Python 3.11
- d) Error

✓

✓ 16. Can a function return a function in Python? *

1/1

- a) Yes
- b) No

✓

✓ 17. What does this print? *

1/1

```
def outer():  
    def inner():  
        return "Hello"  
    return inner  
f = outer()  
print(f())
```

- a) Hello
- b) inner
- c) outer
- d) Error

✓

✓ 18. Can return return an expression result? *

1/1

- a) Yes
- b) No

✓

✓ 19. What happens if return is written without value? *

1/1

- a) Error
- b) Returns None
- c) Returns ""
- d) Returns 0

✓

✓ 20. Which statement is true? *

1/1

- a) Function can return only integers
- b) Function can return only one object
- c) Function can return different types based on conditions
- d) Function cannot return strings

✓

✓ 21. Difference between return and print? *

1/1

- a) return ends function and sends value back; print only displays output
- b) Both are same
- c) print ends function, return displays
- d) None

✓

✓ 22. Output: *

1/1

```
def f():  
    return 10  
print(f())
```

- a) 10
- b) None
- c) Error
- d) Blank

✓



✓ 23. Output: *

1/1

```
def f():
```

```
    print(10)
```

```
print(f())
```

- a) 10
- b) None
- c) 10 followed by None
- d) Error

✓

✓ 24. What is returned here? *

1/1

```
def f():
```

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

```
f()
```

- a) Hi
- b) None
- c) Error

✓

✓ 25. What is returned here? *

1/1

```
def f():
```

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

```
print(f())
```

- a) Hi
- b) None
- c) Hi then None
- d) Error

✓

✓ 26. Can a function return inside a loop? *

1/1

- a) Yes
- b) No

✓

✓ 27. What will happen? *

1/1

```
def f():
```

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

```
        return i
```

```
print(f())
```

- a) 0
- b) 0 1 2
- c) Error
- d) None

✓

✓ 28. Output: *

1/1

```
def f():  
    s = 0  
    for i in range(3):  
        s += i  
    return s  
print(f())
```

- a) 0
- b) 3
- c) 6
- d) Error

✓

✓ 29. Can we return from inside a conditional? *

1/1

- a) Yes
- b) No

✓

✓ 30. What happens if no return is reached in function? *

1/1

- a) Error
- b) Returns None
- c) Skips function

✓

✓ 31. What will this return? *

1/1

```
def f():  
    return {1,2,3}  
  
print(f())
```

- a) [1,2,3]
- b) {1,2,3}
- c) (1,2,3)
- d) Error

✓

✓ 32. Function returning dict: *

1/1

```
def f():  
    return {"a":1}  
  
print(f())
```

- a) {"a":1}
- b) [a:1]
- c) (a,1)
- d) Error

✓

✓ 33. Can a function return a generator? *

1/1

- a) Yes
- b) No

✓

✓ 34. Which of the following is valid? *

1/1

- a) return a,b,c
- b) return (a,b,c)
- c) return [a,b,c]
- d) All of these



✓ 35. What is returned here? *

1/1

```
def f():
```

```
    return [i*i for i in range(3)]
```

```
print(f())
```

- a) [0,1,4]
- b) (0,1,4)
- c) {0,1,4}
- d) Error



✓ 36. Which is correct? *

1/1

- a) return can be used outside a function
- b) return cannot be used outside function



✓ 37. Output of: *

1/1

```
def f():
```

```
    return 1
```

```
    return 2
```

```
print(f())
```

- a) 1
- b) 2
- c) (1,2)
- d) Error



✓ 38. Output: *

1/1

```
def f(x):
```

```
    if x>0:
```

```
        return "Positive"
```

```
    else:
```

```
        return "Negative"
```

```
print(f(-5))
```

- a) Positive
- b) Negative
- c) Error



✓ 39. Which function call is valid?def f(): *

1/1

return

- a) f()
- b) x = f()
- c) Both
- d) None

✓

✓ 40. If function has return type None, then: *

1/1

- a) It returns nothing
- b) It returns None object
- c) It throws error

✓

✓ 41. Can a function return itself? *

1/1

- a) Yes
- b) No

✓

✓ 42. Output: *

1/1

```
def f():
```

```
    return f
```

```
print(f())
```

- a) f()
- b) Function object
- c) None
- d) Error

✓

✓ 43. Output: *

1/1

```
def f(x):
```

```
    return x*2
```

```
y = f(f(2))
```

```
print(y)
```

- a) 2
- b) 4
- c) 8
- d) Error

✓

✓ 44. Output: *

1/1

```
def f():  
    return True  
  
if f():  
    print("Yes")  
  
else:  
    print("No")
```

- a) Yes
- b) No
- c) None
- d) Error



✓ 45. Which is valid? *

1/1

- a) return lambda x: x+1
- b) return def g(): return 1



✓ 46. Output: *

1/1

```
def f():
```

```
    return 1,2,3
```

```
x,y,z = f()
```

```
print(z)
```

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



✓ 47. Output: *

1/1

```
def f():
```

```
    return "A","B"
```

```
print(len(f()))
```

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



✓ 48. Output: *

1/1

```
def f():
```

```
    return
```

```
    print(type(f()))
```

a) NoneType



b) str

c) int

d) tuple

✓ 49. Output: *

1/1

```
def f():
```

```
    return (1)
```

```
    print(type(f()))
```

a) int



b) tuple

c) str

✓ 50. Output: *

1/1

```
def f():
```

```
    return (1,)
```

```
print(type(f()))
```

- a) int
- b) tuple
- c) list



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

Google Forms



