

PYTHON TEST - 3.5 (PASSING PARAMETERS)

Total points 50/50 ?

Passing Parameters in Python

STUDENT NAME *

VIVA

✓ 1. What are parameters in Python functions? *

1/1

- a) Values returned by a function
- b) Variables defined inside a function
- c) Variables listed in the function definition
- d) Constants used in a program

✓



✓ 2. What are arguments in Python? *

1/1

- a) Values passed into a function call
- b) Variables declared globally
- c) Data types of variables
- d) Operators used in expressions

✓

✓ 3. Which keyword is used to define a function in Python? *

1/1

- a) def
- b) func
- c) function
- d) define

✓

✓ 4. In Python, parameters inside a function are: *

1/1

- a) Local variables
- b) Global variables
- c) Both local and global
- d) Constants

✓

✓ 5. Which of the following correctly defines a function with parameters? * 1/1

- a) `def add(a, b):`
- b) `func add(a, b):`
- c) `define add(a, b):`
- d) `function add(a, b):`



✓ 6. How many types of function parameters are there in Python? * 1/1

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



✓ 7. Which of these is NOT a type of parameter in Python? * 1/1

- a) Positional parameter
- b) Keyword parameter
- c) Default parameter
- d) Global parameter



✓ 8. Positional parameters mean: *

1/1

- a) Parameters matched based on order
- b) Parameters matched by name
- c) Parameters with default values
- d) Parameters stored in a dictionary

✓

✓ 9. Keyword arguments are passed: *

1/1

- a) Based on order
- b) Using variable scope
- c) By specifying parameter names
- d) Only for integers

✓

✓ 10. Default parameters are used when: *

1/1

- a) Values are passed explicitly
- b) Values are not provided during the function call
- c) Values are global
- d) Function has no return value

✓

✓ 11. What will `def greet(name): print("Hello", name)` output if called as `*greet("John")`? 1/1

- a) Hello
- b) John
- c) Hello John
- d) Error



✓ 12. What happens if the number of arguments passed is less than the parameters defined? *1/1

- a) Error occurs
- b) Default values are assigned
- c) Function executes normally
- d) Arguments are ignored



✓ 13. In `def add(x, y): return x+y`, calling `add(3)` will result in: * 1/1

- a) 3
- b) Error
- c) 0
- d) None



✓ 14. In positional arguments, order of arguments: *

1/1

- a) Doesn't matter
- b) Must match the parameters
- c) Is randomly chosen by Python
- d) Is optional



✓ 15. Which statement is correct for `def f(a,b): print(a,b)` and call `f(2,3)`? *

1/1

- a) a=2, b=3
- b) a=3, b=2
- c) a=2, b=None
- d) Error



✓ 16. In `def show(name, age): print(name, age)`, calling `show(age=20, name="John")`

*1/1

- a) John 20
- b) 20 John
- c) Error
- d) None



✓ 17. Keyword arguments allow: *

1/1

- a) Reordering of arguments
- b) Omitting arguments
- c) Ignoring function definitions
- d) Declaring constants

✓

✓ 18. What will `def sum(a,b): return a+b; print(sum(b=4,a=5))` return? *

1/1

- a) 9
- b) 20
- c) Error
- d) None

✓

✓ 19. If both positional and keyword arguments are passed, positional ones *1/1 must appear:

- a) After keyword arguments
- b) Before keyword arguments
- c) Anywhere
- d) Only if global

✓

✓ 20. Which call is invalid for `def test(x,y): pass?` *

1/1

- a) `test(3,4)`
- b) `test(x=3,y=4)`
- c) `test(3,y=4)`
- d) `test(x=3,4)`



✓ 21. Default arguments are defined in: *

1/1

- a) Function call
- b) Function definition
- c) Return statement
- d) Global scope



✓ 22. Which is valid? *

1/1

- a) `def f(a=10, b):`
- b) `def f(a, b=10):`
- c) `def f(a=10, b=20, c):`
- d) All of the above



✓ 23. Default arguments should always be placed: *

1/1

- a) Before non-default arguments
- b) After non-default arguments
- c) Anywhere
- d) At global level



✓ 24. What will `def greet(name="Guest"): print("Hello", name); greet()` print? * 1/1

- a) Hello
- b) Guest
- c) Hello Guest
- d) Error



✓ 25. `def f(x=5,y=10): return x+y; print(f())` outputs: *

1/1

- a) 15
- b) 10
- c) 5
- d) Error



✓ 26. Which symbol is used for variable-length positional arguments? * 1/1

- a) *
- b) **
- c) &
- d) #



✓ 27. Which symbol is used for variable-length keyword arguments? * 1/1

- a) *
- b) **
- c) &
- d) \$



✓ 28. In `def f(*args): print(args)`, calling `f(1,2,3)` prints: * 1/1

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



✓ 29. In `def f(**kwargs): print(kwargs)`, calling `f(a=1,b=2)` prints: * 1/1

a) {a:1,b:2}

b) {'a':1,'b':2}

c) (a=1,b=2)

d) Error



✓ 30. What data type does `*args` collect values into? * 1/1

a) List

b) Tuple

c) Dictionary

d) Set



✓ 31. Can positional, keyword, default, and variable-length arguments coexist? *1/1

a) Yes

b) No

c) Only positional and keyword

d) Only default and variable-length



✓ 32. Which order of parameters is correct? *

1/1

- a) Positional → Default → *args → **kwargs ✓
- b) Default → Positional → **kwargs → *args
- c) *args → Positional → Default → **kwargs
- d) None

✓ 33. What is the output of: *

1/1

```
def f(a,b=2,*args,**kwargs):
```

```
    print(a,b,args,kwargs)
```

```
f(1,3,4,5,x=10,y=20)
```

- a) 1 3 (4,5) {'x':10,'y':20} ✓
- b) 1 2 (3,4,5) {'x':10,'y':20}
- c) 1 3 (4,5) {}
- d) Error

✓ 34. In def f(a,*b,**c);, calling f(1,2,3,x=4) assigns: *

1/1

- a) a=1, b=(2,3), c={'x':4} ✓
- b) a=1, b=[2,3], c=x=4
- c) a=1, b=(2), c=(3,4)
- d) Error

✓ 35. Which is invalid? *

1/1

- a) `def f(a,b=5,*args): pass`
- b) `def f(*args,a=5): pass`
- c) `def f(**kwargs,*args): pass`
- d) `def f(a,*args,**kwargs): pass`

✓

✓ 36. In Python, arguments are passed by: *

1/1

- a) Value
- b) Reference
- c) Object reference
- d) Copy

✓

✓ 37. If a mutable object is passed and modified inside a function: *

1/1

- a) Original object changes
- b) Original object remains same
- c) Error occurs
- d) Copy is modified

✓

✓ 38. If an immutable object is passed and modified inside a function: * 1/1

- a) Original object changes
- b) New object is created
- c) Function fails
- d) Error



✓ 39. What will this print? * 1/1

```
def f(x): x.append(5)  
a=[1,2]; f(a); print(a)
```

- a) [1,2]
- b) [1,2,5]
- c) Error
- d) None



✓ 40. What will this print? * 1/1

```
What will this print?  
def f(x): x+=1  
a=10; f(a); print(a)
```

- a) 10
- b) 11
- c) Error
- d) None



✓ 41. What will `def f(a,b,/): return a+b; print(f(2,3))` output? *

1/1

- a) 5
- b) Error
- c) None
- d) (2,3)

✓

✓ 42. The / in function parameters means: *

1/1

- a) Only positional arguments allowed before it
- b) Only keyword arguments allowed before it
- c) Default arguments required
- d) Nothing

✓

✓ 43. The * in function parameters means: *

1/1

- a) Only positional arguments allowed after it
- b) Only keyword arguments allowed after it
- c) All parameters become optional
- d) Function ends

✓

✓ 44. Which of these allows keyword-only arguments? *

1/1

- a) def f(a,*b): pass
- b) def f(*a,b): pass
- c) Both a and b
- d) None



✓ 45. What is the output of: *

1/1

```
def f(a,*b):  
    print(a,b)  
f(2,b=3)
```

- a) 2 3
- b) Error
- c) (2,3)
- d) None



✓ 46. What happens if the same keyword is passed twice in a function call? * 1/1

- a) Last one is used
- b) First one is used
- c) Error
- d) Both are ignored



✓ 47. In `def f(a,b): return a+b`, calling `f(2,"3")` results in: *

1/1

- a) 5
- b) "23"
- c) Error
- d) None

✓

✓ 48. Can functions return values based on parameters? *

1/1

- a) Yes
- b) No
- c) Only integers
- d) Only strings

✓

✓ 49. What will `def f(x,y=10): return x*y; print(f(2))` output? *

1/1

- a) 2
- b) 20
- c) Error
- d) None

✓

✓ 50. Which statement about parameters in Python is correct? *

1/1

- a) Python supports both fixed and variable-length parameters
- b) Python doesn't allow keyword arguments
- c) Default arguments must come first
- d) Python passes arguments by value



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

Google Forms



