

Class #06: Functions in Python

Your Full Name: **Minhazul Kabir**

নিচে Class #06: Functions in Python বিষয়ক ৫০টি MCQ বাংলায় ফরম্যাট আকারে দেওয়া হলো:

ফাংশন ডিফাইন করা

1. Python-এ একটি ফাংশন ডিফাইন করার সঠিক সিনট্যাক্স কী?

- a) def function_name[]:
- b) function function_name():
- c) def function_name():
- d) def function_name[]:

উত্তর: c) def function_name():

2. Python-এ একটি ফাংশন ডিফাইন করার জন্য কোন কীওয়ার্ড ব্যবহার করা হয়?

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

উত্তর: b) def

3. নিম্নলিখিত কোডের আউটপুট কী হবে?

- 4. def greet():
- 5. return "Hello"
- 6. print(greet())

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

উত্তর: a) Hello

7. my_function নামে একটি ফাংশন কল করার সঠিক উপায় কী?

- a) call my_function()
- b) my_function()
- c) invoke my_function()
- d) execute my_function()

উত্তর: b) my_function()

8. যদি একটি ফাংশনে return স্টেটমেন্ট না থাকে, তবে সেই ফাংশনটি কী রিটার্ন করবে?

- a) None

- b) 0
- c) False
- d) Empty string

উত্তর: a) None

2. ফাংশন আর্গুমেন্ট ব্যবহার করা

6. Python-এ একটি ফাংশনে আর্গুমেন্ট পাস করার সঠিক উপায় কী?

- a) function_name(arg)
- b) function_name{arg}
- c) function_name[arg]
- d) function_name<arg>

উত্তর: a) function_name(arg)

7. নিম্নলিখিত কোনটি Python-এ একটি ডিফল্ট আর্গুমেন্টের উদাহরণ?

8. def my_function(a, b=2):

- a) আর্গুমেন্ট b ঐচ্ছিক
- b) আর্গুমেন্ট a এর একটি ডিফল্ট মান রয়েছে
- c) ফাংশনটি দুটি আর্গুমেন্ট নেবেই
- d) ফাংশনটি সঠিকভাবে কাজ করবে না

উত্তর: a) আর্গুমেন্ট b ঐচ্ছিক

9. কীভাবে আপনি একটি ফাংশনে একটি ভেরিয়েবল সংখ্যা আর্গুমেন্ট পাস করতে পারেন?

- a) *args সিনট্যাক্স ব্যবহার করে
- b) **kwargs সিনট্যাক্স ব্যবহার করে
- c) *args এবং **kwargs উভয়ই ব্যবহার করে
- d) উপরের কোনটিই নয়

উত্তর: a) *args সিনট্যাক্স ব্যবহার করে

10. নিম্নলিখিত কোনটি আর্গুমেন্ট নাম দিয়ে পাস করার উদাহরণ?

- a) *args
- b) **kwargs
- c) args
- d) kwargs

উত্তর: b) **kwargs

11. নিম্নলিখিত কোডের আউটপুট কী হবে?

12. def add(a, b=5):

13. return a + b

14. print(add(10))

- a) 15
- b) 10

c) None

d) 5

উত্তর: a) 15

3. রিকার্সন (Recursion) বোঝা

11. Python-এ রিকার্সন কী?

- a) একটি ফাংশন যা নিজেকে কল করে
- b) একটি লুপ যা ফাংশনের মধ্যে থাকে
- c) একটি মেথড যা একাধিক মান রিটার্ন করে
- d) একটি ফাংশন যার প্যারামিটার থাকে না

উত্তর: a) একটি ফাংশন যা নিজেকে কল করে

12. রিকার্সনে বেস কেস কী?

- a) যে শর্তে রিকার্সন থেমে যাবে
- b) রিকার্সনের শুরুর পয়েন্ট
- c) রিকার্সন ফাংশনের রিটার্ন মান
- d) রিকার্সন ফাংশনের প্যারামিটার

উত্তর: a) যে শর্তে রিকার্সন থেমে যাবে

13. নিম্নলিখিত কোডের আউটপুট কী হবে?

14. `def factorial(n):`

15. `if n == 1:`

16. `return 1`

17. `else:`

18. `return n * factorial(n - 1)`

19. `print(factorial(3))`

a) 3

b) 6

c) 9

d) None

উত্তর: b) 6

20. কেন Python-এ গভীর রিকার্সনের জন্য সাধারণত রিকার্সন ব্যবহার করা হয় না?

- a) এটি খুব বেশি মেমরি ব্যবহার করে
- b) এটি স্ট্যাক ওভারফ্লো ঘটাতে পারে
- c) এটি ইটারেশনের তুলনায় ধীর গতিতে কাজ করে
- d) উপরের সব

উত্তর: b) এটি স্ট্যাক ওভারফ্লো ঘটাতে পারে

21. নিম্নলিখিত কোনটি একটি রিকার্সন ফাংশনের উদাহরণ?

- a) `def sum(n): return n + sum(n-1) if n > 0 else 0`

- b) `def sum(n): return n + sum(n-2)`
- c) `def sum(n): return sum(n-1)`
- d) `def sum(n): return n`

উত্তর: a) `def sum(n): return n + sum(n-1) if n > 0 else 0`

4. বিল্ট-ইন ফাংশনসমূহ

16. Python-এ একটি লিস্টের দৈর্ঘ্য বের করার জন্য কোন বিল্ট-ইন ফাংশন ব্যবহার করা হয়?

- a) `len()`
- b) `length()`
- c) `size()`
- d) `count()`

উত্তর: a) `len()`

17. Python-এ একটি স্ট্রিংকে পূর্ণসংখ্যায় রূপান্তর করতে কোন ফাংশন ব্যবহার করা হয়?

- a) `int()`
- b) `convert()`
- c) `str()`
- d) `float()`

উত্তর: a) `int()`

18. `max()` ফাংশনটি কোনটি রিটার্ন করবে যখন এটি একটি সংখ্যার লিস্টে প্রয়োগ করা হবে?

- a) সর্বনিম্ন সংখ্যা
- b) সর্বোচ্চ সংখ্যা
- c) সংখ্যাগুলোর যোগফল
- d) লিস্টের দৈর্ঘ্য

উত্তর: b) সর্বোচ্চ সংখ্যা

19. Python-এ একটি সংখ্যার আবসলুট মান বের করার জন্য কোন ফাংশন ব্যবহার করা হয়?

- a) `abs()`
- b) `round()`
- c) `max()`
- d) `min()`

উত্তর: a) `abs()`

20. Python-এ একটি অবজেক্টের টাইপ জানার জন্য কোন ফাংশন ব্যবহার করা হয়?

- a) `type()`
- b) `get_type()`
- c) `object_type()`
- d) `check_type()`

উত্তর: a) `type()`

5. অ্যানোনিমাস ফাংশন (Lambda Expressions)

21. Python-এ একটি অ্যানোনিমাস ফাংশন তৈরি করতে কোন কীওয়ার্ড ব্যবহার করা হয়?

- a) function
- b) lambda
- c) def
- d) anonymous

উত্তর: b) lambda

22. নিম্নলিখিত কোডের আউটপুট কী হবে?

23. `result = lambda x, y: x * y`

24. `print(result(3, 4))`

- a) 7
- b) 12
- c) 34
- d) None

উত্তর: b) 12

25. Python-এ lambda ফাংশন ব্যবহারের সুবিধা কী?

- a) এগুলি সাধারণ ফাংশনের চেয়ে দ্রুত কাজ করে
- b) এগুলি আরও পড়তে সহজ
- c) এগুলি ছোট, ইনলাইন ফাংশন ডিফাইন করতে সহায়তা করে
- d) এগুলি একাধিক এক্সপ্রেশন ব্যবহার করতে দেয়

উত্তর: c) এগুলি ছোট, ইনলাইন ফাংশন ডিফাইন করতে সহায়তা করে

26. নিম্নলিখিত কোনটি একটি বৈধ lambda ফাংশন?

- a) `lambda x, y: return x + y`
- b) `lambda x, y: x + y`
- c) `lambda: return`
- d) `lambda(x, y) { return x + y }`

উত্তর: b) `lambda x, y: x + y`

27. lambda ফাংশন কীভাবে `map()` ফাংশনের সাথে ব্যবহার করা যায়?

- a) `map(lambda x: x + 1, [1, 2, 3])`
- b) `map(lambda x, y: x + y, [1, 2], [3, 4])`
- c) উভয়ই
- d) উপরের কোনটিই নয়

উত্তর: c) উভয়ই

বিভিন্ন প্রশ্ন

26. Python-এ একটি ফাংশন সম্পর্কে সঠিক কী?

- a) একটি ফাংশন একাধিক মান রিটার্ন করতে পারে

- b) একটি ফাংশন কোনও মান রিটার্ন করতে পারে না
- c) একটি ফাংশনের অবশ্যই প্যারামিটার থাকতে হবে
- d) একটি ফাংশন নিজেই কল করা যাবে না

উত্তর: a) একটি ফাংশন একাধিক মান রিটার্ন করতে পারে

27. Python-এ *args ব্যবহার করে একটি লিস্টের আর্গুমেন্ট ফাংশনে পাস করার সঠিক উপায় কী?

- a) function_name(list)
- b) function_name(*args)
- c) function_name(**args)
- d) function_name([args])

উত্তর: b) function_name(*args)

28. নিম্নলিখিত কোডের আউটপুট কী হবে?

29. def my_func(x):

30. return x * x

31. print(my_func(4))

- a) 16
- b) 8
- c) None
- d) 4

উত্তর: a) 16

32. Python-এ একটি সিকোয়েন্সের উপরে ইটারেট করতে কোন বিল্ট-ইন ফাংশন ব্যবহার করা হয়?

- a) map()
- b) filter()
- c) reduce()
- d) all()

উত্তর: a) map()

33. Python-এ lambda ফাংশনের উদ্দেশ্য কী?

- a) নামকরা ফাংশন তৈরি করা
- b) ছোট, অ্যানোনিমাস ফাংশন ডিফাইন করা
- c) রিকার্সন ফাংশন ডিফাইন করা
- d) একাধিক রিটার্ন মান তৈরি করা

উত্তর: b) ছোট, অ্যানোনিমাস ফাংশন ডিফাইন করা

রিকার্সন এবং ফাংশন আর্গুমেন্ট

31. নিম্নলিখিত কোডের আউটপুট কী হবে?

32. def recurse(x):

33. if x == 0:

34. return 0

35. else:

36. return x + recurse(x - 1)

37. print(recurse(4))

a) 10

b) 4

c) 0

d) None

উত্তর: a) 10

38. Python-এ একটি ফাংশন আর্গুমেন্ট কী?

a) ফাংশনের ভিতরে ডিফাইন করা একটি ভেরিয়েবল

b) একটি মান যা ফাংশনে পাস করা হয়

c) একটি ফাংশনের প্রকার

d) উপরের কোনটি নয়

উত্তর: b) একটি মান যা ফাংশনে পাস করা হয়

39. নিম্নলিখিত কোনটি Python ফাংশনের বৈধ আর্গুমেন্টের ধরন নয়?

a) পজিশনাল আর্গুমেন্ট

b) কীওয়ার্ড আর্গুমেন্ট

c) ডিফল্ট আর্গুমেন্ট

d) ভ্যারিয়েবল-লেংথ আর্গুমেন্ট

উত্তর: d) ভ্যারিয়েবল-লেংথ আর্গুমেন্ট

40. Python-এ একটি ভেরিয়েবলের উপর কলেবল (ফাংশন হিসেবে কল করা যায়) কীভাবে যাচাই করা হয়?

a) is_function()

b) callable()

c) is_callable()

d) function()

উত্তর: b) callable()

41. নিম্নলিখিত কোনটি একটি রিকার্সন ফাংশনের উদাহরণ?

○ a) একটি ফাংশন যা অন্য একটি ফাংশন কল করে

○ b) একটি ফাংশন যা নিজেকে কল করে

○ c) একটি ফাংশন যা একাধিক মান রিটার্ন করে

○ d) একটি ফাংশন যার কোনও রিটার্ন মান নেই

উত্তর: b) একটি ফাংশন যা নিজেকে কল করে

এই প্রশ্নগুলি Python এর ফাংশন, আর্গুমেন্ট, রিকার্সন, বিল্ট-ইন ফাংশন এবং lambda ফাংশন বিষয়ে বিস্তারিতভাবে কভার করে।