

- [Trắc Nghiệm Python 2](#)

# Trắc Nghiệm Python 2

---

Mr. BằngCa

---

1. Cách đúng để tạo một biến với giá trị **None** trong Python là gì?

- a) `x == None`
- b) `x = None`
- c) `x := None`
- d) `x === None`

2. Tên biến nào sau đây là không hợp lệ trong Python?

- a) `variable1`
- b) `_variable`
- c) `variable-1`
- d) `variable_1`

3. Kết quả của đoạn mã sau là gì?

```
a = 5
b = 2
a *= b
print(a)
```

- a) **10**
- b) **7**
- c) **3**
- d) **5**

4. Kiểu của biến **y** trong đoạn mã sau là gì?

```
y = [1, 2, 3]
```

- a) **int**
- b) **str**

- c) `list`
- d) `bool`

5. Làm thế nào để chuyển đổi giá trị của biến `x` từ số nguyên `10` thành chuỗi `"10"`?

- a) `x = str(10)`
- b) `x = int(10)`
- c) `x = int("10")`
- d) `x = str(10)`

6. Kết quả của đoạn mã sau là gì?

```
x = 4
if x % 2 == 0:
    print("Chẵn")
else:
    print("Lẻ")
```

- a) `Chẵn`
- b) `Lẻ`
- c) `None`
- d) `Error`

7. Làm thế nào để viết một câu lệnh if-else trong Python?

- a) `if x < y then else:`
- b) `if x < y: else:`
- c) `if (x < y): else:`
- d) `if x < y: else:`

8. Kết quả của đoạn mã sau là gì?

```
a = 15
b = 20
if a != b:
    print("a không bằng b")
elif a > b:
    print("a lớn hơn b")
else:
    print("a nhỏ hơn b")
```

- a) `a không bằng b`

- b) a lớn hơn b
- c) a nhỏ hơn b
- d) Error

9. Cú pháp đúng cho câu lệnh if-elif-else trong Python là gì?

- a)

```
if x == 10
    print("Bằng 10")
elif x > 10
    print("Lớn hơn 10")
else:
    print("Nhỏ hơn 10")
```

- b)

```
if x == 10:
    print("Bằng 10")
elif x > 10:
    print("Lớn hơn 10")
else:
    print("Nhỏ hơn 10")
```

- c)

```
if x == 10:
    print("Bằng 10")
else if x > 10:
    print("Lớn hơn 10")
else:
    print("Nhỏ hơn 10")
```

- d)

```
if (x == 10):
    print("Bằng 10")
else:
    print("Nhỏ hơn 10")
```

10. Kết quả của đoạn mã sau là gì?

```
x = 8
if x > 10:
    print("Lớn hơn 10")
elif x == 8:
    print("Bằng 8")
else:
    print("Nhỏ hơn 10")
```

- a) Lớn hơn 10
- b) Bằng 8
- c) Nhỏ hơn 10
- d) Error

11. Kết quả của đoạn mã sau là gì?

```
for i in range(6):
    print(i, end=' ')
```

- a) 0 1 2 3 4 5
- b) 1 2 3 4 5 6
- c) 0 1 2 3 4 5 6
- d) 1 2 3 4 5

12. Làm thế nào để tạo một vòng lặp for đơn giản trong Python?

- a)

```
for x in y:
    # do something
```

- b)

```
for (x in y):
    # do something
```

- c)

```
for x in y then:
```

```
# do something
```

- d)

```
for x in y do:  
    # do something
```

13. Kết quả của đoạn mã sau là gì?

```
count = 5  
while count > 0:  
    print(count)  
    count -= 1
```

- a) 5 4 3 2 1 0
- b) 5 4 3 2 1
- c) 4 3 2 1 0
- d) 5 4 3 2

14. Mục đích của câu lệnh **break** trong vòng lặp là gì?

- a) Để tạm dừng vòng lặp
- b) Để thoát khỏi vòng lặp
- c) Để khởi động lại vòng lặp
- d) Để bỏ qua vòng lặp hiện tại

15. Kết quả của đoạn mã sau là gì?

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

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

16. Làm thế nào để tạo một danh sách với các số từ 1 đến 5 trong Python?

- a) `my_list = list(range(5))`
- b) `my_list = list(range(1, 6))`
- c) `my_list = [1, 2, 3, 4, 5]`
- d) `my_list = [1, 2, 3, 4, 5, 6]`

17. Kết quả của đoạn mã sau là gì?

```
my_list = [1, 2, 3]
my_list.append(4)
print(my_list)
```

- a) `[1, 2, 3]`
- b) `[1, 2, 3, 4]`
- c) `[4, 1, 2, 3]`
- d) `[1, 2, 4, 3]`

18. Kết quả của đoạn mã sau là gì?

```
my_list = [1, 2, 3, 4, 5]
print(my_list[2])
```

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

19. Làm thế nào để truy cập phần tử cuối cùng của danh sách trong Python?

- a) `my_list[-1]`
- b) `my_list.last()`
- c) `my_list[len(my_list)-1]`
- d) `my_list.end()`

20. Kết quả của đoạn mã sau là gì?

```
my_list = [1, 2, 3, 4, 5]
print(my_list[:3])
```

- a) `[1, 2, 3]`

- b) [1, 2, 3, 4]
- c) [2, 3, 4]
- d) [1, 2, 3, 4, 5]

21. Làm thế nào để xóa phần tử thứ hai khỏi danh sách trong Python?

- a) `my_list.delete(1)`
- b) `my_list.remove(1)`
- c) `my_list.pop(1)`
- d) `my_list.discard(1)`

22. Kết quả của đoạn mã sau là gì?

```
my_list = [1, 2, 3, 4, 5]
my_list.pop(2)
print(my_list)
```

- a) [1, 2, 3, 4, 5]
- b)

) [1, 2, 4, 5] - c) [2, 3, 4, 5] - d) [1, 2, 3, 5]

23. Làm thế nào để tìm số lần xuất hiện của một phần tử trong danh sách?

- a) `my_list.count(10)`
- b) `my_list.frequency(10)`
- c) `my_list.occure(10)`
- d) `my_list.times(10)`

24. Kết quả của đoạn mã sau là gì?

```
my_list = [1, 2, 2, 3, 4, 2]
print(my_list.count(2))
```

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

25. Làm thế nào để lặp qua các phần tử của một danh sách trong Python?

- a) `for i in range(my_list)`
- b) `for i in my_list:`
- c) `foreach (i in my_list):`
- d) `for i through my_list:`

26. Kết quả của đoạn mã sau là gì?

```
list1 = [1, 2, 3]
list2 = [4, 5, 6]
list1.extend(list2)
print(list1)
```

- a) `[1, 2, 3, [4, 5, 6]]`
- b) `[1, 2, 3, 4, 5, 6]`
- c) `[1, 2, 3] + [4, 5, 6]`
- d) `None`

27. Làm thế nào để tạo một danh sách với các số từ 1 đến 10 trong Python?

- a) `list(range(1, 11))`
- b) `list(range(10))`
- c) `range(1, 11)`
- d) `range(10)`

28. Kết quả của đoạn mã sau là gì?

```
my_list = list(range(3, 8))
print(my_list)
```

- a) `[0, 1, 2, 3, 4, 5, 6, 7, 8]`
- b) `[3, 4, 5, 6, 7, 8]`
- c) `[3, 4, 5, 6, 7]`
- d) `[3, 4, 5, 6, 7, 8, 9]`

29. Làm thế nào để xóa tất cả các phần tử khỏi danh sách trong Python?

- a) `my_list.clear()`
- b) `my_list.remove_all()`
- c) `my_list.delete_all()`
- d) `my_list.empty()`



30. Kết quả của đoạn mã sau là gì?

```
my_list = [1, 2, 3, 4, 5]  
my_list.clear()  
print(my_list)
```

- a) [1, 2, 3, 4, 5]
- b) []
- c) None
- d) [0, 0, 0, 0, 0]