

- Trắc nghiệm Python 1
  - Biến
  - Câu điều kiện
  - Vòng lặp
  - Danh sách (List)

# Trắc nghiệm Python 1

---

Mr. BằngCa

## Biến

---

1. Cách đúng để tạo một biến với giá trị 5 trong Python là gì?

- a) `x == 5`
- b) `x = 5`
- c) `int x = 5`
- d) `x := 5`

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

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

3. Kết quả của đoạn mã sau là gì?

```
x = 10
y = 5
x = x + y
print(x)
```

- a) 5
- b) 10
- c) 15
- d) None

4. Kiểu của biến `x` trong đoạn mã sau là gì?

```
x = "Hello, World!"
```

- a) `int`
- b) `str`
- c) `float`
- d) `bool`

5. Làm thế nào để tạo một biến tham chiếu đến cùng một đối tượng như một biến khác?

- a) `x == y`
- b) `x = y`
- c) `x := y`
- d) `x => y`

## Câu điều kiện

6. Kết quả của đoạn mã sau là gì?

```
x = 10
if x > 5:
    print("Lớn hơn 5")
else:
    print("Bằng hoặc nhỏ hơn 5")
```

- a) `Lớn hơn 5`
- b) `Bằng hoặc nhỏ hơn 5`
- c) `None`
- d) `Error`

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

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

8. Kết quả của đoạn mã sau là gì?

```
x = 10
y = 20
if x > y:
    print("x lớn hơn")
elif x < y:
    print("y lớn hơn")
else:
    print("x và y bằng nhau")
```

- a) x lớn hơn
- b) y lớn hơn
- c) x và y bằng nhau
- d) Error

9. Cú pháp đúng cho câu lệnh if lồng nhau trong Python là gì?

- a)

```
if x > 10
    if y > 10:
        print("x và y lớn hơn 10")
```

- b)

```
if x > 10:
    if y > 10:
        print("x và y lớn hơn 10")
```

- c)

```
if x > 10:
    if (y > 10):
        print("x và y lớn hơn 10")
```

- d)

```
if (x > 10):
    if (y > 10):
```

```
print("x và y lớn hơn 10")
```

10. Kết quả của đoạn mã sau là gì?

```
x = 5
if x > 10:
    print("Lớn hơn 10")
elif x > 5:
    print("Lớn hơn 5")
else:
    print("Bằng hoặc nhỏ hơn 5")
```

- a) Lớn hơn 10
- b) Lớn hơn 5
- c) Bằng hoặc nhỏ hơn 5
- d) Error

## Vòng lặp

11. Kết quả của đoạn mã sau là gì?

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

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

12. Làm thế nào để tạo một vòng lặp while trong Python?

- a)

```
while x > y:
    # do something
```

- b)

```
while (x > y):  
    # do something
```

o c)

```
while x > y then:  
    # do something
```

o d)

```
while x > y do:  
    # do something
```

13. Kết quả của đoạn mã sau là gì?

```
count = 0  
while count < 5:  
    print(count)  
    count += 1
```

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

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

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

15. Mục đích của câu lệnh **continue** trong vòng lặp là gì?

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

# Danh sách (List)

---

16. Làm thế nào để tạo một danh sách trong Python?

- a) `my_list = []`
- b) `my_list = {}`
- c) `my_list = ()`
- d) `my_list = <>`

17. Làm thế nào để thêm một phần tử vào danh sách trong Python?

- a) `my_list.add(10)`
- b) `my_list.append(10)`
- c) `my_list.insert(10)`
- d) `my_list.include(10)`

18. 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]`

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

- a) `my_list[0]`
- b) `my_list[1]`
- c) `my_list.first()`
- d) `my_list.start()`

20. Kết quả của đoạn mã sau là gì?

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

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

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

- a) `my_list.delete(10)`
- b) `my_list.remove(10)`
- c) `my_list.pop(10)`
- d) `my_list.discard(10)`

22. Kết quả của đoạn mã sau là gì?

```
my_list = [1, 2, 3, 4, 5]
my_list.remove(3)
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 độ dài của một danh sách trong Python?

- a) `len(my_list)`
- b) `my_list.length()`
- c) `length(my_list)`
- d) `my_list.size()`

24. Kết quả của đoạn mã sau là gì?

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

- a) `4

– b) 5 – c) 6 – d) None`

25. Làm thế nào để nối hai danh sách trong Python?

- a) `list1 + list2`
- b) `list1.concat(list2)`
- c) `list1.append(list2)`
- d) `list1.extend(list2)`

26. Kết quả của đoạn mã sau là gì?

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

- 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 0 đến 9 trong Python?

- a) `list(range(10))`
- b) `list(range(9))`
- c) `range(10)`
- d) `range(9)`

28. Kết quả của đoạn mã sau là gì?

```
my_list = list(range(5))
print(my_list)
```

- a) `[0, 1, 2, 3, 4]`
- b) `[1, 2, 3, 4, 5]`
- c) `[0, 1, 2, 3, 4, 5]`
- d) `range(5)`

29. Làm thế nào để đảo ngược một danh sách trong Python?

- a) `my_list.reverse()`
- b) `my_list[::-1]`
- c) `reversed(my_list)`
- d) Cả ba cách trên



30. Kết quả của đoạn mã sau là gì?

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

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