

# Linear Regression

*Hoàng-Nguyên Vũ*

Các bạn có thể dùng internet để tìm kiếm nhanh thông tin như cách dùng một hàm nào đó. Các bạn cũng có thể dùng Python editor (như Colab) để thử nhanh hay xác nhận cách dùng một hàm nào đó. Tất cả có 60 câu và thời gian mong muốn hoàn thành trong 90 phút.

Chúc các bạn làm bài tốt!

---

**Câu hỏi 1:** Hàm len trả về kết quả gì khi áp dụng lên chuỗi "Python"?

```
1 print(len("Python"))
```

- A. 5
- B. 6
- C. 7
- D. Error

**Câu hỏi 2:** Kiểu dữ liệu của biến c là gì sau khi thực hiện phép tính?

```
1 a = 10
2 b = 2.5
3 c = a + b
4 print(type(c))
```

- A. Integer
- B. String
- C. Float
- D. Boolean

**Câu hỏi 3:** Kết quả in ra của đoạn code trên là gì?

```
1 numbers = [4, 2, 2, 5, 1, 3, 4, 9]
2 print(numbers.count(4))
```

- A. 0
- B. 8
- C. 2
- D. Error

**Câu hỏi 4:** Kết quả của biến `info` là gì?

```
1 name = "Alice"  
2 age = 25  
3 info = f"{name} is {age} years old"  
4 print(info)
```

- A. Alice is 25 years old
- B. Alice 25
- C. Alice is years old
- D. Error

**Câu hỏi 5:** Giá trị của biến `grade` là gì sau khi thực thi đoạn code trên?

```
1 score = 85  
2 if score >= 90:  
3     grade = "A"  
4 elif score >= 80:  
5     grade = "B"  
6 else:  
7     grade = "C"  
8 print(grade)
```

- A. A
- B. B
- C. C
- D. D

**Câu hỏi 6:** Kết quả của biến `result` là bao nhiêu?

```
1 def a_function(n):  
2     if n == 0:  
3         return 1  
4     else:  
5         return n * a_function(n-1)  
6  
7 result = a_function(4)  
8 print(result)
```

- A. 4
- B. 24
- C. 120
- D. Error

**Câu hỏi 7:** Kết quả in ra của đoạn code trên là gì?

```
1 animals = ['cat', 'dog', 'rabbit', 'horse']  
2 print(animals.index('rabbit'))
```

- A. 0
- B. 1
- C. 2
- D. 3

**Câu hỏi 8:** Giá trị của biến `data` là gì sau khi thực thi đoạn code trên?

```
1 data = {"name": "Bob", "age": 30}  
2 data["age"] += 5  
3 print(data)
```

- A. "name": "Bob", "age": 35
- B. "name": "Bob", "age": 30
- C. "name": "Bob", "age": 40
- D. Error

**Câu hỏi 9:** Kết quả của biến `result` là bao nhiêu?

```
1 def a_function(n):  
2     if n <= 0:  
3         return 0  
4     elif n == 1:  
5         return 1  
6     else:  
7         return a_function(n-1) + a_function(n-2)  
8  
9 result = a_function(6)  
10 print(result)
```

- A. 5
- B. 8
- C. 13
- D. Error

**Câu hỏi 10:** Kết quả của đoạn code trên là gì?

```
1 for i in range(1, 6):  
2     print(i)  
3     break
```

A. 1 2 3 4 5 6

B. 0

C. 1

D. 1 2 3 4 5

**Câu hỏi 11:** Kết quả của đoạn code trên là gì?

```
1 i = 0
2 while i < 3:
3     print(i)
4     i += 1
```

A. 1 2 3

B. 0 1 2

C. 0 1 2 3

D. 1 2

**Câu hỏi 12:** Đoạn code trên làm gì?

```
1 file = open("example.txt", "w")
2 file.write("Hello, World!")
3 file.close()
```

A. Đọc nội dung từ file "example.txt"

B. Ghi nội dung vào file "example.txt"

C. Error

D. Đổi tên file "example.txt"

**Câu hỏi 13:** Kết quả của đoạn code trên là gì?

```
1 for i in range(5):
2     if i == 3:
3         break
4     print(i)
```

A. 0 1 2 3

B. 0 1 2

C. infinite loop

D. 1 2 3

**Câu hỏi 14:** Kết quả của đoạn code trên là gì?

```
1 for i in range(5):  
2     if i % 2 == 0:  
3         continue  
4     print(i)
```

- A. 0 2 4
- B. 1 3 5
- C. infinite loop
- D. 1 3

**Câu hỏi 15:** Kết quả của đoạn code trên là gì nếu nội dung của file "example.txt" là "Python Programming"?

```
1 with open("example.txt", "r") as file:  
2     content = file.read()  
3 print(content)
```

- A. In ra: AttributeError
- B. In ra: example.txt
- C. In ra: Python Programming
- D. Không in ra gì

**Câu hỏi 16:** Kết quả của đoạn code trên là gì?

```
1 x = 10  
2 while x > 0:  
3     x -= 3  
4     if x % 2 == 0:  
5         print(x)  
6         break  
7     print(x, end=" ")
```

- A. 7 4 1
- B. 7 4
- C. 4 1
- D. 7 1

**Câu hỏi 17:** Kết quả của đoạn code trên là gì?

```
1 result = []  
2 for i in range(5, -6, -2):  
3     result.append(i)  
4 print(result)
```

- A. [5, 3, 1, -1, -3, -5, -7]
- B. [5, 4, 3, 2, 1, 0, -1, -2, -3, -4, -5]
- C. [5, 3, 1, -1, -3]
- D. [5, 3, 1, -1, -3, -5]

**Câu hỏi 18:** Kết quả của đoạn code trên là gì?

```
1 result = 0
2 for i in range(1, 10, 2):
3     if i % 3 == 0:
4         break
5     result += i
6 print(result)
```

- A. 1
- B. 1 3 5 7 9
- C. 4
- D. 16

**Câu hỏi 19:** Đoạn code trên có mục đích dùng để làm gì?

```
1 with open("data.txt", "r") as file:
2     lines = file.readlines()
3
4 count = 0
5 for line in lines:
6     if line.strip() == "Python":
7         count += 1
8
9 print(count)
```

- A. Tách riêng các từ "Python" từ file data.txt
- B. Đếm số lần xuất hiện của "Python" trong file data.txt
- C. Thay thế "Python" trong file data.txt bằng "1"
- D. Đếm số dòng chỉ chứa chuỗi "Python" trong file data.txt

**Câu hỏi 20:** Kết quả của đoạn code trên là gì?

```
1 i = 0
2 while i < 5:
3     i += 1
4     if i == 3:
5         continue
6     print(i, end=" ")
```

- A. Infinite loop
- B. 0 1 2 4
- C. 1 2 4 5
- D. 1 2 4

**Câu hỏi 21:** Kết quả của đoạn code trên là gì?

```
1 x = 0
2 while x <= 4:
3     if x == 3:
4         continue
5     print(x)
6     x += 1
```

- A. 0 1 2 4
- B. 0 1 2 3
- C. 0 1 3 4
- D. infinite loop

**Câu hỏi 22:** Kết quả của đoạn mã sau là gì?

```
1 text = "Python is fun"
2 result = text.replace("fun", "awesome")
3 print(result[5:])
```

- A. is awesome
- B. Python is awesome
- C. Python is fun
- D. n is awesome

**Câu hỏi 23:** Đoạn mã sau thực hiện thao tác gì trên ma trận?

```
1 matrix = [[1, 2, 3],
2           [4, 5, 6],
3           [7, 8, 9]]
4 result = [row[1] for row in matrix]
5 print(result)
```

- A. Lấy tất cả các phần tử của cột đầu tiên
- B. Lấy tất cả các phần tử của cột thứ hai
- C. Lấy tất cả các phần tử của hàng đầu tiên

D. Lấy tất cả các phần tử của hàng thứ hai

**Câu hỏi 24:** Kết quả trả về của đoạn mã sau?

```
1 matrix = [[1, 2, 3],  
2           [4, 5, 6],  
3           [7, 8, 9]]  
4 print(matrix[0:2][0:1])
```

- A. `[[1] [4]]`
- B. `[[1, 2, 3] [4, 5, 6]]`
- C. `[[1, 2] [4, 5]]`
- D. `[[1, 2, 3]]`

**Câu hỏi 25:** Đoạn mã sau thực hiện thao tác gì trên chuỗi?

```
1 text = "a,b,c,d,e"  
2 result = text.split(",")[:-1]  
3 print(result)
```

- A. Chia chuỗi thành các phần tử và đảo ngược danh sách các phần tử
- B. Chia chuỗi thành các phần tử và loại bỏ dấu phẩy
- C. Chia chuỗi thành các phần tử và đảo ngược từng phần tử
- D. Chia chuỗi thành các phần tử và giữ nguyên thứ tự

**Câu hỏi 26:** Kết quả của đoạn mã sau là gì?

```
1 text = "Data Science"  
2 result = text[5:].upper() + text[:4].lower()  
3 print(result)
```

- A. SCIENCEdata
- B. SCIENCEDATA
- C. DATAscience
- D. scienceDATA

**Câu hỏi 27:** Kết quả cuối cùng của `filtered_data` là gì?

```
1 data = ['Python', 'Data Science', 'Machine Learning', 'AI']  
2 filtered_data = [word for word in data if 'a' in word.lower()]  
3 print(filtered_data)
```

- A. `['Data Science', 'Machine Learning']`



- B. ['Python', 'Data Science', 'AI']
- C. ['Data Science', 'Machine Learning', 'AI']
- D. ['Python', 'Machine Learning', 'AI']

**Câu hỏi 28:** Kết quả của đoạn mã sau là gì?

```
1 text = "1234567890"  
2 result = text[::2].replace('2', 'X') + text[1::2].replace('5', 'Y')  
3 print(result)
```

- A. 1X34Y67890
- B. 1357924680
- C. 1X35794Y680
- D. 1X357924680

**Câu hỏi 29:** Kết quả của đoạn mã sau là gì?

```
1 data = [['Python', '3.8'], ['Data Science', 'ML'], ['AI', 'Deep Learning']  
          ]]  
2 cleaned_data = [[item.lower() for item in row] for row in data]  
3 print(cleaned_data)
```

- A. [['Python', '3.8'], ['Data Science', 'ML'], ['AI', 'Deep Learning']]
- B. [['python', '3.8'], ['data science', 'ml'], ['ai', 'deep learning']]
- C. [['python', '38'], ['datascience', 'ml'], ['ai', 'deeplearning']]
- D. [['PYTHON', '3.8'], ['DATA SCIENCE', 'ML'], ['AI', 'DEEP LEARNING']]

**Câu hỏi 30:** Kết quả đoạn mã dưới đây?

```
1 my_set = {1, 2, 3, 4, 5}  
2 my_set.add(6)  
3 print(my_set[5])
```

- A. 6
- B. 5
- C. Error
- D. None

**Câu hỏi 31:** Kết quả đoạn mã dưới đây?

```
1 my_set = {1, 2, 3}
2 your_set = {3, 4, 5}
3 union_set = my_set.union(your_set)
4 print(union_set)
```

- A. {1, 2, 3, 4, 5}
- B. {1, 2, 4, 5}
- C. {1, 2}
- D. {3}

**Câu hỏi 32:** Kết quả đoạn mã dưới đây?

```
1 my_set = {1, 2, 3}
2 your_set = {4, 5, 6}
3 sdset = my_set ^ your_set
4 print(sdset)
```

- A. {1, 2, 4, 5, 6}
- B. {1, 2, 4, 5}
- C. {1, 2, 3, 4, 5, 6}
- D. None

**Câu hỏi 33:** Kết quả đoạn mã dưới đây?

```
1 my_tuple = (1, 2, 3)
2 my_set = set(my_tuple)
3 my_dict = {x: x**2 for x in my_set}
4 print(len(my_dict))
```

- A. 2
- B. 3
- C. 4
- D. 5

**Câu hỏi 34:** Đoạn mã sau thực hiện thao tác gì trên các set?

```
1 my_set = {1, 2, 3}
2 your_set = {3, 4, 5}
3 inset = my_set & your_set
4 print(inset)
```

- A. Trả về tất cả các phần tử trong my\_set

- B. Trả về tất cả các phần tử trong `your_set`
- C. Trả về các phần tử chung của `my_set` và `your_set`
- D. Trả về các phần tử chỉ có trong `my_set`

**Câu hỏi 35:** Kết quả của đoạn mã sau là gì?

```
1 my_set = {1, 2, 3, 4}
2 your_set = {3, 4, 5, 6}
3 dset = my_set - your_set
4 print(dset)
```

- A. {1, 2}
- B. Error
- C. {1, 2, 5, 6}
- D. {3, 4}

**Câu hỏi 36:** Kết quả của đoạn mã sau là gì?

```
1 my_tuple = (1, 2, 3, 4, 5)
2 new_tuple = my_tuple[::-1]
3 print(new_tuple)
```

- A. (5, 4, 3, 2, 1)
- B. (1, 2, 3, 4, 5)
- C. (5, 4, 3)
- D. Error: Tuple can't be changed

**Câu hỏi 37:** Output của đoạn code dưới đây là?

```
1 def process_tuple(t):
2     return t[1::2]
3
4 my_tuple = (1, 2, 3, 4, 5, 6, 7, 8)
5 result = process_tuple(my_tuple)
6 print(result)
```

- A. (1, 3, 5, 7)
- B. (2, 4, 6, 8)
- C. (1, 2, 3, 4)
- D. (5, 6, 7, 8)

**Câu hỏi 38:** Kết quả của đoạn mã sau là gì?

```
1 my_dict = {'a': 1, 'b': 2, 'c': 3}
2 my_set = set(my_dict.values())
3 my_tuple = tuple(my_set)
4 print(my_tuple[1])
```

- A. 1
- B. 2
- C. 3
- D. Error

**Câu hỏi 39:** Kết quả của đoạn mã sau là gì?

```
1 my_set = {1, 2, 3, 4, 5}
2 another_set = {1, 2, 3, 4, 5, 6, 7}
3 result = my_set.issubset(another_set) and not my_set.issuperset(
    another_set)
4 print(result)
```

- A. True
- B. False
- C. None
- D. Error

**Câu hỏi 40:** Đoạn code này đang cố thực hiện điều gì?

```
1 my_dict = {'a': 1, 'b': 2, 'c': 3}
2 new_dict = {k: v for k, v in my_dict.items() if v > 1}
3 result = tuple(new_dict.keys())
4 print(result)
```

- A. Loại bỏ tất cả các phần tử có values lớn hơn hoặc bằng 1
- B. Loại bỏ tất cả các phần tử có keys lớn hơn hoặc bằng 1
- C. Loại bỏ tất cả phần tử có values bé hơn hoặc bằng 1
- D. Đảo ngược dictionary theo values

**Câu hỏi 41:** Giá trị cuối cùng của my\_tuple là?

```
1 my_dict = {'x': 10, 'y': 20}
2 my_set = {key for key in my_dict if my_dict[key] > 15}
3 my_tuple = tuple(my_set)
4 print(my_tuple)
```

- A. ('x', 'y')
- B. ('y',)
- C. ('x',)
- D. ('10', '20')

**Câu hỏi 42:** Kết quả cuối cùng in ra màn hình?

```
1 def setOperation(set1, set2):
2     result = (set1 | set2) - (set1 & set2)
3     final = {x for x in result if x % 2 == 0}
4     return final
5
6 set_a = {1, 2, 3, 4, 5}
7 set_b = {4, 5, 6, 7, 8}
8 result = setOperation(set_a, set_b)
9 print(result)
```

- A. 8, 2, 6
- B. 6, 7, 8
- C. 2, 6, 8, 1, 3, 7
- D. 6, 8

**Câu hỏi 43:** Kết quả in ra màn hình của đoạn code dưới đây là gì?

```
1 my_set = {1, 2, 3}
2 my_dict = {x: x**3 for x in my_set}
3 my_tuple = tuple(my_dict.items())
4 filtered_tuple = tuple((k, v) for k, v in my_tuple if v % 2 != 0)
5 print(filtered_tuple)
```

- A. ((1, 1), (2, 8), (3, 27))
- B. ((1, 1), (3, 27))
- C. ((2, 8),)
- D. ((1, 1), (3, 27), (4, 64))

**Câu hỏi 44:** Kết quả của đoạn code sau?

```
1 my_dict = {'a': (1, 2), 'b': (3, 4), 'c': (5, 6)}
2 result = {k: sum(v) for k, v in my_dict.items()}
3 sorted_result = sorted(result.items(), key=lambda item: item[1], reverse=
    True)
4 print(sorted_result)
```

- A. [('c', 11), ('b', 7), ('a', 3)]

- B. [('a', 3), ('b', 7), ('c', 11)]
- C. [('b', 7), ('c', 11), ('a', 3)]
- D. [('c', 11), ('a', 3), ('b', 7)]

**Câu hỏi 45:** Câu nào sau đây đúng về class và object trong Python?

- A. Class là một đối tượng cụ thể của object
- B. Object là một đối tượng cụ thể của class
- C. Class và object là giống nhau
- D. Object là một thuộc tính của class

**Câu hỏi 46:** Phương thức nào sau đây được gọi là constructor trong Python?

- A. `__init__`
- B. `__del__`
- C. `__str__`
- D. `__repr__`

**Câu hỏi 47:** Kết quả của đoạn code sau là gì?

```
1 class MyClass:
2     def __init__(self, value):
3         self.value = value
4
5 obj = MyClass(self, 10)
6 print(obj.value)
```

- A. 10
- B. 0
- C. None
- D. Error

**Câu hỏi 48:** Đoạn mã sau thực hiện thao tác gì?

```
1 class MyClass:
2     def __init__(self, value):
3         self.value = value
4
5 obj = MyClass(20)
6 obj.value = 30
7 print(obj.value)
```

- A. Tạo một class với thuộc tính value
- B. Tạo một class với phương thức value
- C. Tạo một object với thuộc tính value ban đầu là 20, sau đó thay đổi thành 30
- D. Tạo một object với thuộc tính value ban đầu là 30, sau đó thay đổi thành 20

**Câu hỏi 49:** Kết quả của đoạn code sau là gì?

```
1 class Person:
2     def __init__(self, name, age):
3         self.name = name
4         self.age = age
5
6     def birthday(self):
7         self.age += 1
8
9 person = Person("Alice", 30)
10 person.birthday()
11 print(person.age)
```

- A. 29
- B. 30
- C. 31
- D. 32

**Câu hỏi 50:** Đây là kết quả của đoạn code sau?

```
1 class Counter:
2     count = 0
3
4     def __init__(self):
5         Counter.count += 1
6
7 c1 = Counter()
8 c2 = Counter()
9 c3 = Counter()
10 print(Counter.count)
```

- A. Error
- B. 3
- C. None
- D.

**Câu hỏi 51:** Kết quả của đoạn mã sau là gì?

```
1 class Animal:
2     def __init__(self, name):
3         self.name = name
4
5     def speak(self):
6         return f"{self.name} makes a noise"
7
8 class Dog(Animal):
9     def speak(self):
10        return f"{self.name} barks"
11
12 dog = Dog("Buddy")
13 print(dog.speak())
```

- A. Buddy makes a noise
- B. Buddy barks
- C. Dog makes a noise
- D. Dog barks

**Câu hỏi 52:** Đây là kết quả của đoạn code dưới đây?

```
1 class A:
2     def __init__(self):
3         self.__x = 1
4
5     def get_x(self):
6         return self.__x
7
8 class B(A):
9     def __init__(self):
10        super().__init__()
11        self.__x = 2
12
13    def get_x(self):
14        return self.__x
15
16 b = B()
17 print(b.get_x())
```

- A. 1
- B. 2
- C. Error
- D. None

**Câu hỏi 53:** Kết quả của đoạn code sau là gì?



```
1 class MyClass:
2     __hidden_variable = 10
3
4     def add(self, increment):
5         self.__hidden_variable += increment
6         print(self.__hidden_variable)
7
8 obj = MyClass()
9 obj.add(2)
10 obj.add(5)
11 print(obj.__hidden_variable)
```

- A. 12, 17, 10
- B. 12, 17, Error
- C. 10, 12, 17
- D. Error

**Câu hỏi 54:** Output cuối cùng được in ra màn hình của đoạn code sau là gì?

```
1 class A:
2     def __init__(self):
3         self.value = 0
4
5     def increment(self):
6         self.value += 1
7
8 class B(A):
9     def __init__(self):
10         super().__init__()
11         self.value = 5
12
13     def increment(self):
14         self.value += 2
15
16 class C(B):
17     def __init__(self):
18         super().__init__()
19         self.value = 10
20
21     def increment(self):
22         super().increment()
23         self.value += 3
24
25 c = C()
26 c.increment()
27 print(c.value)
```

- A. 11
- B. 12

C. 13

D. 15

**Câu hỏi 55:** Kết quả của đoạn mã sau là gì?

```
1 class Base:
2     def __init__(self):
3         self.a = "Base"
4
5     def display(self):
6         return self.a
7
8 class Derived(Base):
9     def __init__(self):
10        super().__init__()
11        self.a = "Derived"
12
13    def display(self):
14        return self.a + " " + super().display()
15
16 d = Derived()
17 print(d.display())
```

A. "Derived Derived"

B. "Derived"

C. "Derived Base"

D. "Base Derived"

**Câu hỏi 56:** Kết quả của đoạn code dưới đây là gì?

```
1 class Parent:
2     def __init__(self, value):
3         self.value = value
4
5     def show(self):
6         return self.value
7
8 class Child(Parent):
9     def __init__(self, value, extra):
10        super().__init__(value)
11        self.extra = extra
12
13    def show(self):
14        return super().show() + self.extra
15
16 c = Child(10, 5)
17 print(c.show())
```

A. 10

- B. 15
- C. 10, 5
- D. Error

**Câu hỏi 57:** Kết quả của đoạn code trên là gì?

```
1 class Vehicle:
2     def __init__(self, brand):
3         self.brand = brand
4
5 class Car(Vehicle):
6     def __init__(self, brand, model):
7         super().__init__(brand)
8         self.model = model
9
10 car = Car("Toyota", "Corolla")
11 print(car.brand, car.model)
```

- A. Error
- B. Toyota
- C. Corolla
- D. Toyota Corolla

**Câu hỏi 58:** Kết quả của đoạn code trên là gì?

```
1 class Parent:
2     def greet(self):
3         print("Hello from Parent")
4
5 class Child(Parent):
6     def greet(self):
7         pass
8
9 child = Child()
10 child.greet()
```

- A. Hello from Parent
- B. Hello from Child
- C. Error
- D. None

**Câu hỏi 59:** Kết quả của đoạn code sau là gì?

```
1 class Base:
2     def __init__(self):
3         self.__name = "John"
4         self._gender = "Male"
5
6     def get_name(self):
7         return self.__name
8
9 class Derived(Base):
10    def __init__(self):
11        super().__init__()
12
13    def get_gender(self):
14        return self._gender
15
16 obj = Derived()
17 print(obj.get_name())
18 print(obj.get_gender())
```

A. John, Male

B. Error, Male

C. John, Error

D. Error, Error

**Câu hỏi 60:** Kết quả của đoạn code dưới đây là gì?

```
1 class X:
2     def method(self):
3         return "X"
4
5 class Y(X):
6     def method(self):
7         return "Y" + super().method()
8
9 class Z(Y):
10    def method(self):
11        return "Z" + super().method()
12
13 z = Z()
14 print(z.method())
```

A. ZYX

B. XYZ

C. ZY

D. YX

**Câu hỏi 61:** Kết quả của đoạn code dưới đây là gì?

```
1 class A:
2     def __init__(self):
3         self.value = "A"
4
5     def get_value(self):
6         return self.value
7
8 class B(A):
9     def __init__(self):
10        super().__init__()
11        self.value += "B"
12
13    def get_value(self):
14        return self.value
15
16 class C(B):
17     def __init__(self):
18         self.value = "C"
19         super().__init__()
20
21 c = C()
22 print(c.get_value())
```

- A. AB
- B. C
- C. CAB
- D. CB

**Câu hỏi 62:** Kết quả của đoạn code trên là gì?

```
1 class A:
2     def __init__(self, x=1):
3         self.x = x
4
5     def method(self):
6         return "A"
7
8 class B(A):
9     def __init__(self, x=2, y=3):
10        super().__init__(x)
11        self.y = y
12
13    def method(self):
14        return "B" + super().method()
15
16 class C(B):
17     def __init__(self, x=4, y=5, z=6):
18         super().__init__(x, y)
19         self.z = z
20
21     def method(self):
```

```
22         return "C" + super().method()  
23  
24 c = C()  
25 print(c.method())  
26 print(c.__dict__)
```

- A. CBA và {'x': 4, 'y': 5, 'z': 6}
- B. CBA và {'x': 2, 'y': 3, 'z': 6}
- C. ABC và {'x': 2, 'y': 3, 'z': 6}
- D. ABC và {'x': 4, 'y': 5, 'z': 6}