Microsoft Windows [Version 10.0.18362.356]

(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Hoang Ta>D:

D:\>cd D:\A\_learning\abc\codingdojo\_week3\_python\python\_stack\django\django\_intro\book\_authors\_proj

D:\A\_learning\abc\codingdojo\_week3\_python\python\_stack\django\django\_intro\book\_authors\_proj>python manage.py makemigrations

Traceback (most recent call last):

File "manage.py", line 8, in <module>

from django.core.management import execute\_from\_command\_line

ModuleNotFoundError: No module named 'django'

The above exception was the direct cause of the following exception:

Traceback (most recent call last):

File "manage.py", line 14, in <module>

) from exc

ImportError: Couldn't import Django. Are you sure it's installed and available on your PYTHONPATH environment variable? Did you forget to activate a virtual environment?

D:\A\_learning\abc\codingdojo\_week3\_python\python\_stack\django\django\_intro\book\_authors\_proj>cd cd D:\A\_learning\abc\codingdojo\_week3\_python\python\_stack\my\_environments\django

The filename, directory name, or volume label syntax is incorrect.

D:\A\_learning\abc\codingdojo\_week3\_python\python\_stack\django\django\_intro\book\_authors\_proj>cd D:\A\_learning\abc\codingdojo\_week3\_python\python\_stack\my\_environments\django

D:\A\_learning\abc\codingdojo\_week3\_python\python\_stack\my\_environments\django>Scripts\activate

(django) D:\A\_learning\abc\codingdojo\_week3\_python\python\_stack\my\_environments\django>cd D:\A\_learning\abc\codingdojo\_week3\_python\python\_stack\django\django\_intro\book\_authors\_proj

(django) D:\A\_learning\abc\codingdojo\_week3\_python\python\_stack\django\django\_intro\book\_authors\_proj>python manage.py makemigrations

Migrations for 'books\_authors\_app':

apps\books\_authors\_app\migrations\0001\_initial.py

- Create model Author

- Create model Book

- Add field books to author

(django) D:\A\_learning\abc\codingdojo\_week3\_python\python\_stack\django\django\_intro\book\_authors\_proj>python manage.py migrate

Operations to perform:

Apply all migrations: admin, auth, books\_authors\_app, contenttypes, sessions

Running migrations:

Applying contenttypes.0001\_initial... OK

Applying auth.0001\_initial... OK

Applying admin.0001\_initial... OK

Applying admin.0002\_logentry\_remove\_auto\_add... OK

Applying admin.0003\_logentry\_add\_action\_flag\_choices... OK

Applying contenttypes.0002\_remove\_content\_type\_name... OK

Applying auth.0002\_alter\_permission\_name\_max\_length... OK

Applying auth.0003\_alter\_user\_email\_max\_length... OK

Applying auth.0004\_alter\_user\_username\_opts... OK

Applying auth.0005\_alter\_user\_last\_login\_null... OK

Applying auth.0006\_require\_contenttypes\_0002... OK

Applying auth.0007\_alter\_validators\_add\_error\_messages... OK

Applying auth.0008\_alter\_user\_username\_max\_length... OK

Applying auth.0009\_alter\_user\_last\_name\_max\_length... OK

Applying books\_authors\_app.0001\_initial... OK

Applying sessions.0001\_initial... OK

(django) D:\A\_learning\abc\codingdojo\_week3\_python\python\_stack\django\django\_intro\book\_authors\_proj>python manage.py shell

Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:54:40) [MSC v.1900 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license" for more information.

(InteractiveConsole)

>>> from apps.books\_authors\_app.models import \*

>>> cs = Book.objects.create(title='C Sharp', desc='learn C#')

>>> java = Book.objects.create(title='Java', desc ='learn Java')

>>> python = Book.objects.create(title='Python', desc='learn Python')

>>> php = Book.objects.create(title='PHP', desc='learn PHP')

>>> ruby = Book.objects.create(title=

... exit

... exit

File "<console>", line 3

exit

^

SyntaxError: invalid syntax

>>> ruby = Book.objects.create(title='Ruby',desc='learn Ruby')

>>> au1 = Author.objects.create(first\_name='Jane',last\_name='Austen')

>>> au2 = Author.objects.create(first\_name='Emily',last\_name='Dickinson')

>>> au3 = Author.objects.create(first\_name='Fyodor',last\_name='Dostoevksy')

>>> au4 = Author.objects.create(first\_name='william',last\_name='Shakespeare')

>>> au5 = Author.objects.create(first\_name='Lau',last\_name='Tzu')

>>>

>>> exit()

(django) D:\A\_learning\abc\codingdojo\_week3\_python\python\_stack\django\django\_intro\book\_authors\_proj>python manage.py makemigrations

You are trying to add a non-nullable field 'notes' to book without a default; we can't do that (the database needs something to populate existing rows).

Please select a fix:

1) Provide a one-off default now (will be set on all existing rows with a null value for this column)

2) Quit, and let me add a default in models.py

Select an option: 1

Please enter the default value now, as valid Python

The datetime and django.utils.timezone modules are available, so you can do e.g. timezone.now

Type 'exit' to exit this prompt

>>> 'dont read this book'

Migrations for 'books\_authors\_app':

apps\books\_authors\_app\migrations\0002\_book\_notes.py

- Add field notes to book

(django) D:\A\_learning\abc\codingdojo\_week3\_python\python\_stack\django\django\_intro\book\_authors\_proj>python manage.py makemigrations

No changes detected

(django) D:\A\_learning\abc\codingdojo\_week3\_python\python\_stack\django\django\_intro\book\_authors\_proj>python manage.py migrate

Operations to perform:

Apply all migrations: admin, auth, books\_authors\_app, contenttypes, sessions

Running migrations:

Applying books\_authors\_app.0002\_book\_notes... OK

(django) D:\A\_learning\abc\codingdojo\_week3\_python\python\_stack\django\django\_intro\book\_authors\_proj>python manage.py shell

Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:54:40) [MSC v.1900 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license" for more information.

(InteractiveConsole)

>>> cs

Traceback (most recent call last):

File "<console>", line 1, in <module>

NameError: name 'cs' is not defined

>>> cs = Book.objects.get(id=1)

Traceback (most recent call last):

File "<console>", line 1, in <module>

NameError: name 'Book' is not defined

>>> from apps.book\_authors\_app import \*

Traceback (most recent call last):

File "<console>", line 1, in <module>

ModuleNotFoundError: No module named 'apps.book\_authors\_app'

>>> from apps.book\_authors\_app.models import \*

Traceback (most recent call last):

File "<console>", line 1, in <module>

ModuleNotFoundError: No module named 'apps.book\_authors\_app'

>>> from apps.books\_authors\_app import \*

>>> cs

Traceback (most recent call last):

File "<console>", line 1, in <module>

NameError: name 'cs' is not defined

>>> cs = Book.objects.get(id=1)

Traceback (most recent call last):

File "<console>", line 1, in <module>

NameError: name 'Book' is not defined

>>> cs = Book.objects.get(id=1)

Traceback (most recent call last):

File "<console>", line 1, in <module>

NameError: name 'Book' is not defined

>>> Book

Traceback (most recent call last):

File "<console>", line 1, in <module>

NameError: name 'Book' is not defined

>>> Author

Traceback (most recent call last):

File "<console>", line 1, in <module>

NameError: name 'Author' is not defined

>>> from apps.books\_authors\_app.models import \*

>>> cs = Book.objects.get(id=1)

>>> cs

<Dojo object: C Sharp (1)>

>>> cs.title = 'C#'

>>> cs.title

'C#'

>>> au4 = Author.objects.get(id=4)

>>> au4.first\_name

'william'

>>> au4.first\_name='Bill'

>>> au4.first\_name

'Bill'

>>> au1 = Author.objects.first()

>>> au1

<Ninja object: Jane (1)>

>>> book1 = Book.objects.first()

>>> book2 = Book.objects.get(id=2)

>>> au1.books.add(book1)

>>> au1.books.add(book2)

>>> au1.books.all()

<QuerySet [<Dojo object: C Sharp (1)>, <Dojo object: Java (2)>]>

>>> au2 = Author.objects.get(id=2)

>>> book3 = Book.objects.get(id=3)

>>> au3.books.add(book1,book2,book3)

Traceback (most recent call last):

File "<console>", line 1, in <module>

NameError: name 'au3' is not defined

>>> au2.books.add(book1,book2,book3)

>>> au2.books.all()

<QuerySet [<Dojo object: C Sharp (1)>, <Dojo object: Java (2)>, <Dojo object: Python (3)>]>

>>> au1.book.all()

Traceback (most recent call last):

File "<console>", line 1, in <module>

AttributeError: 'Author' object has no attribute 'book'

>>> au3 = Author.objects.get(id=3)

>>> book4=Book.objects.get(id=4)

>>> au3.books.add(book1,book2,book3,book4)

>>> au3.books.all()

<QuerySet [<Dojo object: C Sharp (1)>, <Dojo object: Java (2)>, <Dojo object: Python (3)>, <Dojo object: PHP (4)>]>

>>> book5=Book.objects.get(id=5)

>>> au4=Author.objects.last()

>>> au4

<Ninja object: Lau (5)>

>>> au4.books.add(book1,book2,book3,book4,book5)

>>> au4.books.all()

<QuerySet [<Dojo object: C Sharp (1)>, <Dojo object: Java (2)>, <Dojo object: Python (3)>, <Dojo object: PHP (4)>, <Dojo object: Ruby (5)>]>

>>> au4\_books = au4.books.all()

>>> au4\_books

<QuerySet [<Dojo object: C Sharp (1)>, <Dojo object: Java (2)>, <Dojo object: Python (3)>, <Dojo object: PHP (4)>, <Dojo object: Ruby (5)>]>

>>> book3.authors.all()

<QuerySet [<Ninja object: Emily (2)>, <Ninja object: Fyodor (3)>, <Ninja object: Lau (5)>]>

>>> book3.authors.remove(au1)

>>> book3.authors.all()

<QuerySet [<Ninja object: Emily (2)>, <Ninja object: Fyodor (3)>, <Ninja object: Lau (5)>]>

>>> first\_third = book3.authors.first()

>>> first\_third

<Ninja object: Emily (2)>

>>> book3.authors.remove(book3.authors.first())

>>> book3.authors.all()

<QuerySet [<Ninja object: Fyodor (3)>, <Ninja object: Lau (5)>]>

>>> book2.authors.all()

<QuerySet [<Ninja object: Jane (1)>, <Ninja object: Emily (2)>, <Ninja object: Fyodor (3)>, <Ninja object: Lau (5)>]>

>>> book2.authors.add(au5=Author.objects.get(id=5))

Traceback (most recent call last):

File "<console>", line 1, in <module>

TypeError: add() got an unexpected keyword argument 'au5'

>>> au5 = Author.objects.get(id=5)

>>> au5

<Ninja object: Lau (5)>

>>> book2.authors.add(au5)

>>> book2.authors.all()

<QuerySet [<Ninja object: Jane (1)>, <Ninja object: Emily (2)>, <Ninja object: Fyodor (3)>, <Ninja object: Lau (5)>]>

>>> au5.books.all()

<QuerySet [<Dojo object: C Sharp (1)>, <Dojo object: Java (2)>, <Dojo object: Python (3)>, <Dojo object: PHP (4)>, <Dojo object: Ruby (5)>]>

>>> au4.books.all()

<QuerySet [<Dojo object: C Sharp (1)>, <Dojo object: Java (2)>, <Dojo object: Python (3)>, <Dojo object: PHP (4)>, <Dojo object: Ruby (5)>]>

>>> au3.books.all()

<QuerySet [<Dojo object: C Sharp (1)>, <Dojo object: Java (2)>, <Dojo object: Python (3)>, <Dojo object: PHP (4)>]>

>>> au3.books.all()

<QuerySet [<Dojo object: C Sharp (1)>, <Dojo object: Java (2)>, <Dojo object: Python (3)>, <Dojo object: PHP (4)>]>

>>> book5.authors.all()

<QuerySet [<Ninja object: Lau (5)>]>

>>>