**from** django.test **import** TestCase

**from** django.utils **import** timezone

**from** .models **import** Question

**class** **QuestionModelTests**(TestCase):

**def** **test\_was\_published\_recently\_with\_future\_question**(self):

"""

was\_published\_recently() returns False for questions whose pub\_date

is in the future.

"""

time **=** timezone**.**now() **+** datetime**.**timedelta(days**=**30)

future\_question **=** Question(pub\_date**=**time)

self**.**assertIs(future\_question**.**was\_published\_recently(), **False**)

>>> **from** django.test **import** Client

>>> *# create an instance of the client for our use*

>>> client **=** Client()

>>> *# get a response from '/'*

>>> response **=** client**.**get('/')

Not Found: /

>>> *# we should expect a 404 from that address; if you instead see an*

>>> *# "Invalid HTTP\_HOST header" error and a 400 response, you probably*

>>> *# omitted the setup\_test\_environment() call described earlier.*

>>> response**.**status\_code

404

>>> *# on the other hand we should expect to find something at '/polls/'*

>>> *# we'll use 'reverse()' rather than a hardcoded URL*

>>> **from** django.urls **import** reverse

>>> response **=** client**.**get(reverse('polls:index'))

>>> response**.**status\_code

200

>>> response**.**content

b'\n <ul>\n \n <li><a href="/polls/1/">What&#x27;s up?</a></li>\n \n </ul>\n\n'

>>> response**.**context['latest\_question\_list']

<QuerySet [<Question: What's up?>]>

**class** **MyTestCase**(TestCase):

@classmethod

**def** **setUpClass**(cls):

super()**.**setUpClass()

**...**

@classmethod

**def** **tearDownClass**(cls):

**...**

super()**.**tearDownClass()

**class** **AnimalTestCase**(TestCase):

fixtures **=** ['mammals.json', 'birds']

**from** django.test **import** TestCase

**class** **LoginTestCase**(TestCase):

**def** **test\_login**(self):

*# First check for the default behavior*

response **=** self**.**client**.**get('/sekrit/')

self**.**assertRedirects(response, '/accounts/login/?next=/sekrit/')

*# Then override the LOGIN\_URL setting*

**with** self**.**settings(LOGIN\_URL**=**'/other/login/'):

response **=** self**.**client**.**get('/sekrit/')

self**.**assertRedirects(response, '/other/login/?next=/sekrit/')

or

@override\_settings(LOGIN\_URL**=**'/other/login/')

**from** django.core **import** mail

**from** django.test **import** TestCase

**class** **EmailTest**(TestCase):

**def** **test\_send\_email**(self):

*# Send message.*

mail**.**send\_mail(

'Subject here', 'Here is the message.',

'from@example.com', ['to@example.com'],

fail\_silently**=False**,

)

*# Test that one message has been sent.*

self**.**assertEqual(len(mail**.**outbox), 1)

*# Verify that the subject of the first message is correct.*

self**.**assertEqual(mail**.**outbox[0]**.**subject, 'Subject here')

**class** **MyTests**(TestCase):

@skipUnlessDBFeature('supports\_transactions')

**def** **test\_transaction\_behavior**(self):

*# ... conditional test code*

**pass**