U29163YQH : but normally they shouldn't need it as they can easily retrieve it for you
U48F1FMEJ : this is a pretty basic questions - but when an exception is thrown the rest of the code after where the exception occurred is not executed unless it was handled in a `try` `except`?
U48F1FMEJ : I'm deciding whether I should test code with a context manager of an object that throws an exception, although I don't think I am correct in the what happens
U29163YQH : yes. you can add a `finally` if you want a piece of code to always be executed
U29163YQH : it depends on the what but in some case it can overlapp yes
U48F1FMEJ : so let's say that's the code and we're expecting a new_object to be created based on the paramteters, but an exception can occur like a URLError or something
U48F1FMEJ : what I'm trying to do is test the correct exception is thrown/handled
U48F1FMEJ : so I was thinking...
U5NMSURAQ : `as new_object` looks strange to me
U29163YQH : If you are testing code I believe both pytest and unitest have special context manager to test errors
U5NMSURAQ : just do```
with self.assertRaises(URLError):
    new_object = NewObject(param_to_cause_exception)
```

U48F1FMEJ : yah that is unittest
U5NMSURAQ : also, don't test anything else in this function
U5NMSURAQ : you're testing only the fact that wrong URL causes an exception
U29163YQH : in that case if the assert is correct the test will continue as expected
U48F1FMEJ : with the `as new_object` I can pass it through the rest of the code as an exception object which I found here<https://docs.python.org/2/library/unittest.html#unittest.TestCase.assertRaises>
U29163YQH : but <@U5NMSURAQ> make a good point
U29163YQH : oh ok. You want to test what is in the exception ?
U48F1FMEJ : thanks <@U5NMSURAQ> that does make sense
U48F1FMEJ : right
U48F1FMEJ : and how the rest of the code responds to that <@U29163YQH> if that makes sense
U29163YQH : both object have the same name that's a little confusing
U29163YQH : you can interact with the exception after the `with` like any other object. For example to check what's inside
U48F1FMEJ : right, check error codes and such
U48F1FMEJ : I think I was incorrectly thinking of how the object would be handled if an exception was thrown (not under test)
U48F1FMEJ : in my test I am asserting the correct exception is thrown given a bad parameter
U48F1FMEJ : I think I've got it from here thanks guys <@U5NMSURAQ> :taco: <@U29163YQH> :taco:
U48F1FMEJ : I think I just gave you guys 2 tacos each but you guys were great so keep 'em
U29163YQH : thanks good luck :slightly_smiling_face:
U28MDQRL2 : Best way to order a list based on another list?
U29163YQH : there is something like `sorted()`
U29163YQH : and I guess you can pass like a lambda to do custom sort
U5NMSURAQ : ```&gt;&gt;&gt; order = ['b', 'c', 'a']
&gt;&gt;&gt; sorted(['a', 'b', 'c'], key=order.index)
['b', 'c', 'a']
&gt;&gt;&gt;
```

U28MDQRL2 : yeah <@U0L8Y8ZEW> helped me on the django channel. btw have a :taco:
U0L8Y8ZEW : <@U5NMSURAQ> :taco: I missed that you could just provide order.index instead of a full on lambda. This will be much better
U5NMSURAQ : :heart:
U29163YQH : Do you have a specific question about dns ?
U5NMSURAQ : <@U29163YQH> I'm using namecheap
U5NMSURAQ : Does this record look alright?
U5NMSURAQ : Or should it be the whole domain name?
U29163YQH : this will point `www.my_awesome_website` to the same IP as `c.storage....`
U29163YQH : no you don't need to put the whole name