

Never Create Interdependent Test Methods

When unit tests are interdependent, it is hard to identify which unit is really broken. Indeed, if test_Y needs test_X, then when test_X fails, test_Y will also fail, even if the unit tested by test_Y might still work. Also, if you do this way you can't run a single test at once and you will thus lose so much time either to run all tests or to find which tests must be run before this one.

There is many way of avoid such practice, see below good example with some shared code that will create needed objects.

Bad Example:

```
def test_X(self):
    foo = self.getFooModule().newContent(id='foo')
    # Here start some assertions for functionality A

def test_Y(self):
    foo = self.getFooModule().getObject('foo')
    # Here start some assertions for functionality B
```

Good Example:

```
def createFoo(self):
    return self.getFooModule().newContent()

def test_X(self):
    foo = self.createFoo()
    # Here start some assertions for functionality A

def test_Y(self):
    foo = self.createFoo()
    # Here start some assertions for functionality B
```