

TSD 1~3 심화

Use Explicit Relative Imports

Code	Import Type	Usage
<code>from core.views import FoodMixin</code>	absolute import	Use when importing from outside the current app
<code>from .models import WaffleCone</code>	explicit relative	Use when importing from another module in the current app

BAD

```
from eight.constants import UTM_KEYS
from eight.constants import choices_values
from eight.constants.choices import CreditType
from eight.constants.choices import DealRepaymentState
from eight.constants.choices import DealState
from eight.constants.choices import DealType
from eight.constants.choices import PaperType
from eight.constants.choices import PaymentType
from eight.constants.choices import SectionType
from eight.constants.choices import UseOption
from eight.models import CreditRating
from eight.models import Deal
from eight.models import DealApplication
from eight.models import Investment
from eight.models import KcbKScore
from eight.models import Payback
from eight.models import SpecialDetail
from eight.tasks import send_email
from eight.tasks import update_user_irr
from eight.transaction import do_cancel_investment
from eight.transaction import do_invest
from eight.utils.decorators import email_confirm_required
from eight.utils.decorators import email_required
from eight.utils.decorators import remember_last_query_params
from eight.utils.financial import get_payback_info_by_amount
from eight.utils.financial import get_tax
from eight.utils.sessionutils import get_utmlog_values
from eight.utils.sms import send_lms
```

Good

```
from django.contrib import admin

from eight.constants.choices import Authority

from .models import Bank
from .models import BankAccount
```

Single quotes

Use single quotes for strings, or a double quote if the the string contains a single quote. Don't waste time doing unrelated refactoring of existing code to conform to this style.

BAD

```
def test_referer_domain(self):
    domain = "8percent.kr"
    utmlog = UtmLogFactory(referer="http://{0}".format(domain))
    self.assertEqual(utmlog.referer_domain, domain)
```

Docsting

<https://www.python.org/dev/peps/pep-0257/>

One-line Docstrings




One-liners are for really obvious cases. They should really fit on one line. For example:








```
def kos_root():
    """Return the pathname of the KOS root directory."""
    global _kos_root
    if _kos_root: return _kos_root
    ...
```

```
def complex(real=0.0, imag=0.0):  
    """Form a complex number.  
  
    Keyword arguments:  
    real -- the real part (default 0.0)  
    imag -- the imaginary part (default 0.0)  
    """  
    if imag == 0.0 and real == 0.0:  
        return complex_zero  
    ...
```

```
>>> import math  
>>> math.__doc__  
'This module is always available. It provides access to the\nmathematical functions defined by the C standard.'
```

Pycharm 에서는 F1을 눌러 보세요.

Documentation assertE...




 < Pyth...

```
class TestCase(object)

def assertEqual
(self, first, second, msg=None)
Inferred type: (self: TestCase, first: Any, second: Any, msg: Any) -> None
```

Fail if the two objects are unequal as determined by the '==' operator.

External documentation:
<http://docs.python.org/3.5/library/unittest.html#unittest.TestCase.assertEqual>

 8: Hierarchy
 Database
 Documentation

BAD

```
def loan_apply_error(request):
    """
    대출 신청 중 에러나는 경우 보여주는 페이지. messages를 통해서 에러 메시지를 받아서 화면에 보여준다
    """
    msg_list = [str(msg) for msg in messages.get_messages(request)]

    # 기본 메시지
    if not msg_list:
        msg_list.append('일시적 오류가 발생하였습니다.')

    return render(request, 'pages/loan_error_mobile.html', dict(message="\n".join(msg_list)))
```

한줄로 표시하자. (물론 주석이 없는것 보다 좋지)

assertRaisesMessage

In tests, use [assertRaisesMessage\(\)](#) instead of [assertRaises\(\)](#) so you can check the exception message. Use [assertRaisesRegex\(\)](#) only if you need regular expression matching.

Bad

```
class CreditRatingRate만기일시상환TestCase(CreditRatingVersion2TestCase):
    def test_rm_period(self):
        with self.assertRaises(Exception):
            self.assertRaises(12, self.cr.period)
```

Good

```
def test_invest_point_not_sufficient(self):
    with self.assertRaisesMessage(ValidationError, '포인트 잔고가 부족합니다.'):
        do_invest(self.user, 10000, self.deal, point_amount=20000)
```

SQLite -> Postgresql

- Type에서 오는 문제
- 다음 코드를 어떻게 테스트 할까?

```
@advisory_lock("invest_lock")
@transaction.atomic
def do_invest(user, amount, deal, auto_invest=False, point_amount=0):
    from eight.tasks import make_userpaybacks
```

PIP

패키지 관리에 대한 간략한 소개

- <https://github.com/pypa/sampleproject/blob/master/setup.py>
- <https://pypi.python.org/pypi>