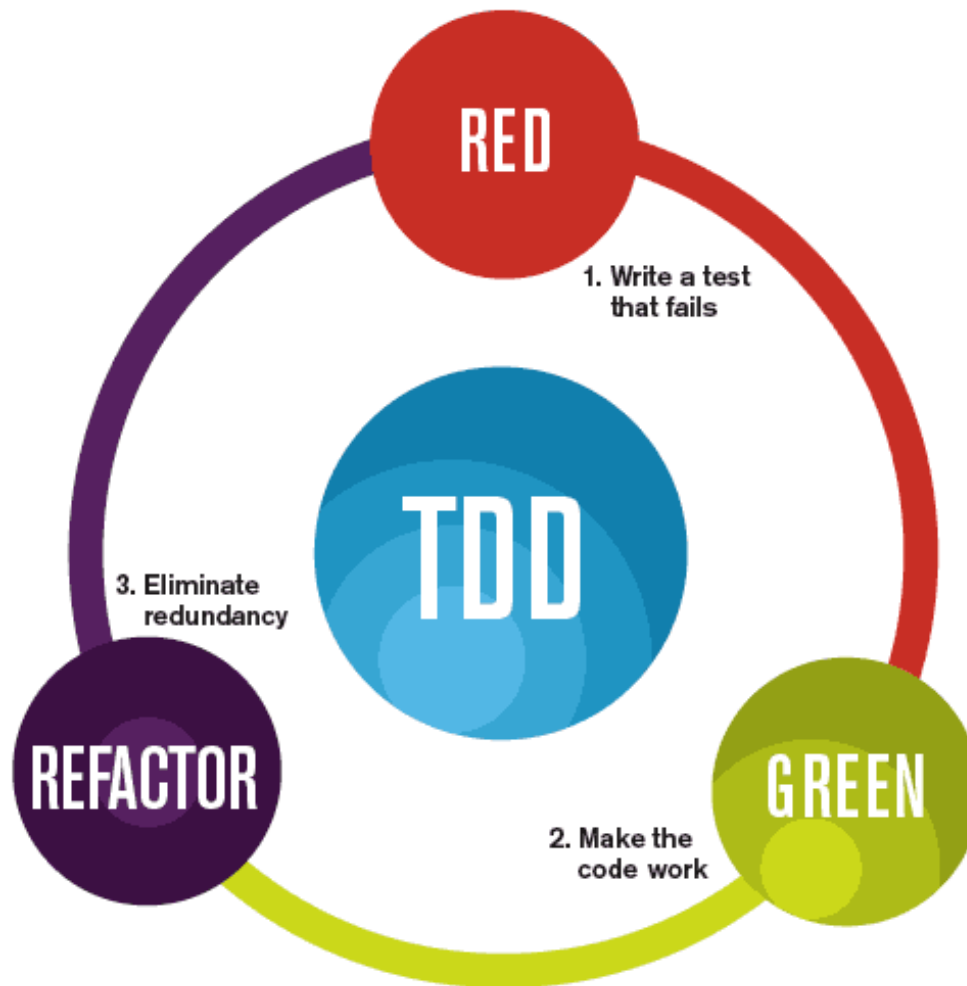


# SciLifeLab



Enabler for Life Sciences





The mantra of Test-Driven Development (TDD) is "red, green, refactor."



<https://python-guide.readthedocs.org/en/latest/writing/tests/>

# putron™

**Beautiful** is better than ugly.  
**Explicit** is better than implicit. **Simple**  
is better than complex. **Complex** is better  
than complicated. **Flat** is better than  
nested. **Sparse** is better than dense.  
**Readability** counts. *Special cases* aren't  
special enough to  
break the rules.

Although **practicality** beats purity. *Errors* should never  
pass silently. Unless **explicitly** silenced. In the face of  
*ambiguity*, **refuse** the temptation to guess. There should be **one**  
— and preferably only one — obvious way to do it. Although that  
way may not be obvious at first *unless you're Dutch*. **Now** is  
better than never. Although never is **often** better than *right*  
now. If the implementation is *hard* to explain, it's a **bad**  
idea. If the implementation  
is *easy* to explain, it  
*may* be a **good** idea.  
**Namespaces** are  
one *honking great*  
idea — let's do  
more of those!

**Beautiful** is better than ugly.  
**Explicit** is better than implicit. **Simple**  
is better than complex. **Complex** is better  
than complicated. **Flat** is better than  
nested. **Sparse** is better than dense.  
**Readability** counts. *Special cases* aren't  
special enough to  
break the rules.  
Although **practicality** beats purity. *Errors* should never  
pass silently. Unless **explicitly** silenced. In the face of  
*ambiguity*, **refuse** the temptation to guess. There should be **one**  
— and preferably only one — obvious way to do it. Although that  
way may not be obvious at first *unless you're Dutch*. **Now** is  
better than never. Although never is **often** better than *right*  
now. If the implementation is *hard* to explain, it's a **bad**  
idea. If the implementation  
is *easy* to explain, it  
*may* be a **good** idea.  
**Namespaces** are  
one *honking great*  
idea — let's do  
more of those!

idea. If the implementation  
is *easy* to explain, it  
*may* be a **good** idea.  
**Namespaces** are  
one *honking great*  
idea — let's do  
more of those!





[https://github.com/brainstorm/python\\_koans](https://github.com/brainstorm/python_koans)

```
Terminal — bash — 90x26
test_reduce_will_blow_your_mind has expanded your awareness.
test_use_pass_for_iterations_with_no_body has expanded your awareness.

Thinking AboutGenerators
test_coroutines_can_take_arguments has expanded your awareness.
test_generating_values_on_the_fly has expanded your awareness.
test_generator_expressions_are_a_one_shot_deal has expanded your awareness.
test_generator_keeps_track_of_local_variables has expanded your awareness.
test_generator_method_will_yield_values_during_iteration has expanded your awareness.
test_generators_are_different_to_list_comprehensions has expanded your awareness.
test_generators_can_see_if_they_have_been_called_with_a_value has expanded your awareness.
s.
test_generators_can_take_coroutines has expanded your awareness.
test_generator_method_with_parameter has damaged your karma.

You have not yet reached enlightenment ...
AssertionError: '-=> FILL ME IN! <==-' != [4, 9, 16]

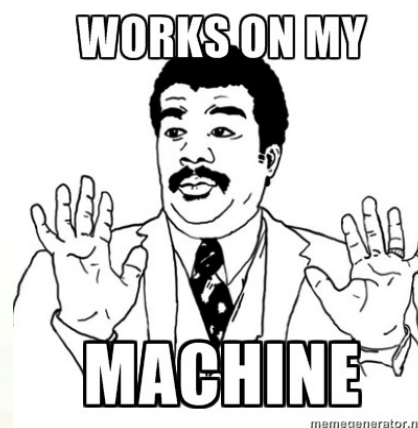
Please meditate on the following code:
File "/Users/Greg/hg/python_koans/python 2/koans/about_generators.py", line 75, in test_
generator_method_with_parameter
    self.assertEqual(__, list(result))


Flat is better than nested.
d60-65-195-206:python 2 Greg$
```

# T Travis



# Jenkins



SciLifeLab + 

=



<https://github.com/SciLifeLab/bcbio-nextgen-deploy>





```
import pdb; pdb.set_trace()
```

<http://docs.python.org/2/library/pdb.html>

<https://raw.githubusercontent.com/nblock/pdb-cheatsheet/master/pdb-cheatsheet.png>

...when finished, run "pip install ipdb", read the docs and see the difference



From: `about_asserts.py`

To: `about_control_statements.py`