



# Python

# Les Trois Mousquetaires

Tox, Travis and Coveralls.

@lqez / SMARTSTUDY



# 파이썬 삼총사

Tox, Travis 그리고 Coveralls.

@lqez / SMARTSTUDY

알림:

이 발표에는 코드가 나오지 않습니다.

Disclaimer:

This session does not have any code.

알림:

실습은 3일차 튜토리얼에서 같이 해요.

Disclaimer:

Practice together at 3rd day's tutorial.

이 발표는 무엇을 위함인가  
The purpose of this session

# Tox

vision: standardize testing in **Python**

<https://testrun.org/tox/>

솔직히 얘기해서  
travis랑 coveralls가  
파이썬하고 무슨 상관이 있습니까.

Frankly speaking,  
there are no direct correlation  
between travis/coveralls and Python.

우리는 도대체 왜 파이썬을 쓰죠?

Then why do we use Python?



이 세상에는 왜 다양한  
프로그래밍 언어가 있는가

Why do we have plenty of  
programming languages?

**기계 : 사람**

**Machine : Human**

**효율**

**Efficiency**

문화

Culture

공동체

Community

**나누고 돌려주기**

**Share and give back**

우리는 도대체 왜 파이썬을 좋아하죠?

Then why do we **like** Python?

그래서 이 발표는 무엇을 위함인가

So, the purpose of this session



파이썬 코드로  
공동체에 기여하기 위한 방법 알기

To find out the way  
how **contribute to community**  
via Python codes

이야기는 이렇게 시작됩니다.

Story begins from,

한 소녀가 있었다.

There was a girl.

작지만 소중한...

Small, but important...

파이썬 코드 조각

Python code snippet

소녀는 물었다.

**“이 코드가 쓸만한 지 한 번 봐주십시오.”**

She asked,

*“Please have a look how valuable  
this code is.”*

하지만 지나가는 이 중  
코드를 유심히 보는 이는 없었다.

But there is no one  
even just looking over it.

절망한 소녀는 코드를 안고 울었다.

How desperate she is.  
Just cry and crying over.



울다 지친 소녀는  
꿈 속에서 우연히 파이썬 아이돌을 만났다.

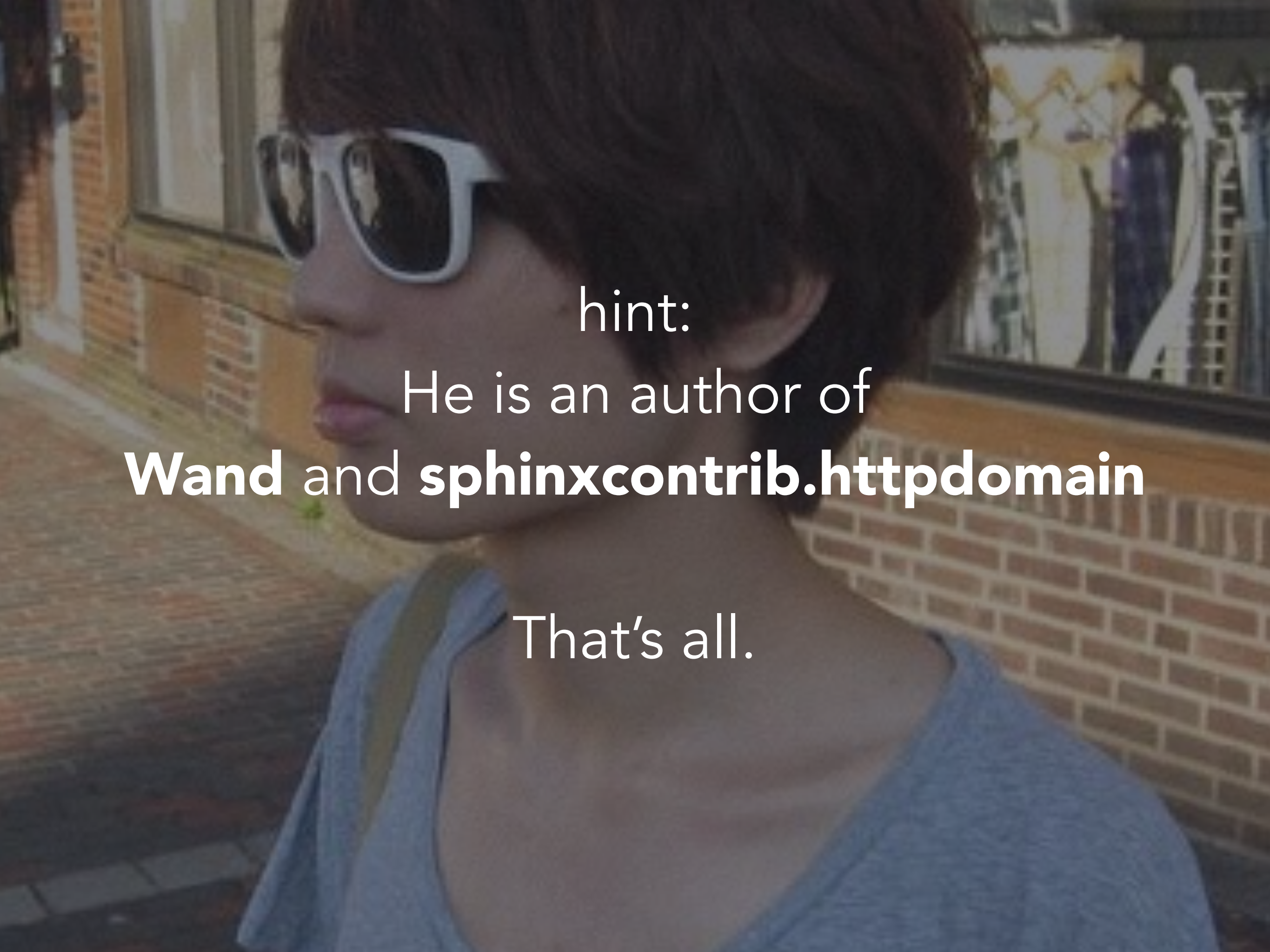
She met a Python idol in her dream.



(딱히 누구라고 하지는 않겠습니다.)

(Actually, I barely know about him.)





hint:

He is an author of  
**Wand** and **sphinxcontrib.httpdomain**

That's all.

어쨌든

Anyway

아이돌이 말했다.

The idol says,

“바보야! 문제는 테스트야!”

*“Idiot! The real problem is **Testing!**”*

**테스트!**

**Testing!**

파이썬은 2.1 부터  
**unittest** 프레임워크가 내장되어 있습니다.

Python supports  
**unittest** framework since version 2.1.



그냥 쓰면 됩니다.

Just use it.

테스트 코드를 보여드리고 싶지만...  
코드 안 쓸꺼라고 했으니까...

I really want to show you  
an unit testing code but I already  
swore not embedding one...

어쨌든

Anyway

그럼 그냥 쓰면 되는데  
**Tox** 얘기는 왜 나왔나?

Then why do we mentioned  
**Tox** for it?

# tox

vision: standardize testing in **Python**

이번 PyCon APAC 2016 등록하고  
받은 팔찌가 2개 있을겁니다.

You may have two wrist bands  
of PyCon APAC 2016.



하나는 노란색이고, 다른 하나는 하늘색.

One is yellow and other is skyblue.





하나는 Python 2, 다른 하나는 Python 3.

One means Python 2,  
and other means Python 3.



이번 행사 슬로건인  
**Respect, Diversity**에 따라 만들었습니다.

I made these bands according to  
the slogan of this year  
- **Respect, Diversity.**

저는 다양성을 존중합니다만,  
존중하기 위해서는 노력이 필요합니다.

I really respect diversities,  
but need more effort to preserve it.

이 슬라이드를 두 가지 언어로  
작성하기 시작한 것을 후회하는 것 처럼요.

I already regret to make this slides  
in bilingual format.

노오오오오오오력이 필요합니다.

We need Effooooooooooooooooort.

심지어 저는 이번에 제가 말할 대부분의  
문장을 글로 적고 있습니다.

I also even wrote text into this slide  
what i'm speaking now.

이것 또한 누군가에게  
작은 도움이 되기를 바라면서요.

I wish my little effort also could be  
helpful for someone.

어쨌든

Anyway

파이썬에는 2.7, 3.5 외에도  
다양한 런타임이 존재합니다.

Python has many runtimes,  
not only 2.7 and 3.5.



다양한 런타임.

**Diversity** of runtimes.

역시 노오오오오오오력이 필요합니다.

Again, we need Effooooooooooooort.

pyenv 등의 환경을 쓴다고 해도  
여러 파이썬 환경을 테스트하기는 귀찮습니다.

It's a such boring stuff to test  
many kind of python runtimes  
even with help of pyenv.

그래서 **Tox** 가 필요합니다.

That's why we need **Tox**.

삼총사 중 첫번째를 소개하는데  
벌써 50페이지를 소비해버렸습니다.

I already spent 50 pages  
just introducing why we need  
the first one of Les Trois Mousquetaires.

죄송하지만, 코드가 없으니까  
어떻게든 되겠죠...

Sorry but, we don't have any code  
I can manage this one...

코드 한 줄 쓰지 않고  
2.7, 3.4, 3.5, 3.6 를 테스트 성공!  
(했다고 칩시다)

Without one line of code  
Succeed to test 2.7, 3.4, 3.5 and 3.6.  
(Yeah, just assuming)

하지만, 장고가 출동하면 어떨까?

But, how about Django's diversity?



# import module change for django 1.6 #11

**Merged** lqez merged 1 commit into `summernote:master` from `unknown repository` on Nov 12, 2013

Conversation 1 Commits 1 Files changed 1



perhapsspy commented on Nov 12, 2013



ImportError: No module named defaults

django.conf.urls.defaults is deprecated use django.conf.urls instead

import module change for django 1.6 ...

5b88fda



lqez merged commit `4ab4d2b` into `summernote:master` on Nov 12,

1 check passed

Revert

1.6

# Issue with django 1.9 #108

 **Closed**

**peterchibunna** opened this issue on Dec 10, 2015 · 2 comments



**peterchibunna** commented on Dec 10, 2015



This plugin does not run on django 1.9. The error is on this line:  
`from django.utils.importlib import import_module`



**Robinfr** commented on Dec 11, 2015



`import_module` was deprecated in 1.8 and removed in 1.9.



**lqez** commented on Dec 16, 2015

Summ 

We already patched this issue by [#84](#)  
And checkout <https://travis-ci.org/summernote/django-summernote/builds/108111111>  
1.9 tests. :)

1.9

# Removed deprecated patterns syntax #107

**Merged** lqez merged 1 commit into `summernote:master` from `claudep:removed_patterns` on Feb 15

Conversation 7    Commits 1    Files changed 2



claudep commented on Dec 9, 2015



Django 1.10 won't support the deprecated `django.conf.urls.patterns`.

Removed deprecated patterns syntax ...

✓ 4b842ba



lqez commented on Dec 9, 2015

Summe



Thanks! I'll add a test for it (If 1.10 is available for testing) and merge it after

master

장고에도 다양한 버전이 존재하고  
**LTS**(Long term support)가 있는지 아시나요?

Did you know;  
Django has multiple active versions and  
also comes with **LTS**(Long term support)?

## **Python versions**

2.7   3.2   3.3   3.4   3.5

x

1.7   1.8   1.9   1.10   dev

## **Django versions**

# Python versions

2.7 3.2 3.3 3.4 3.5

문제:  
테스트 조합의 가짓수를 구하시오.

X

Question:

1.7 1.8 1.9 1.10 dev

How many combinations to be tested?

# Django versions

Python versions

2.7 3.2 3.3 3.4 3.5  
1.7 1.8 1.9 1.10 dev

Django versions

# Python versions

2.7 3.2 3.3 3.4 3.5

틀렸습니다.

Wrong answer.

1.7 1.8 1.9 1.10 dev

# Django versions



# What Python version can I use with Django?

Django version	Python versions
1.8	2.7, 3.2 (until the end of 2016), 3.3, 3.4, 3.5
<b>1.9, 1.10</b>	<b>2.7, 3.4, 3.5</b>

For each version of Python, only the latest micro release (A.B.C) is officially supported. You can find the latest micro version for each series on the [Python download page](#).

Typically, we will support a Python version up to and including the first Django LTS release whose security support ends after security support for that version of Python ends. For example, Python 3.3 security support ends September 2017 and Django 1.8 LTS security support ends April 2018. Therefore Django 1.8 is the last version to support Python 3.3.

**Tox**에서는 이런 조합을  
`envlist`로 지정할 수 있습니다.

With **Tox**, you can make your own  
combinations with `envlist`.

설정을 마치고 tox 를 실행하면 끝!

After setting then just type tox,  
and get **PROFIT!**

---

```
py27-dj17: commands succeeded
py34-dj17: commands succeeded
py27-dj18: commands succeeded
py27-dj19: commands succeeded
py27-djmaster: commands succeeded
py34-dj18: commands succeeded
py34-dj19: commands succeeded
py34-djmaster: commands succeeded
py35-dj18: commands succeeded
py35-dj19: commands succeeded
py35-djmaster: commands succeeded
congratulations :)
```

```
lqezs-mba:django-summernote lqez$ █
```

...이야기로 돌아와서,  
소녀는 좋은 꿈을 꾸었습니다.

... Back to the story,  
She dreamed such a beautiful dream.

기억은 잘 나지 않지만,  
소녀는 기억을 더듬어 테스트를 작성했습니다.

Hard to remember the dream,  
but she tried to write testing code on it.

그리고 소녀는 다시 물었다.

**“이 코드가 쓸만한 지 부디 한 번 봐주십시오.”**

And then she asked again,  
*“Please, please have a look  
how valuable this code is.”*

하지만 지나가는 이 중  
코드를 유심히 보는 이는 없었다.

But there is no one  
even just looking over it.



좌절하려고 돌아가려는 그 때,  
지나가던 사람이 뱉은 한 마디...

Getting back to home with empty hands,  
One stranger says silently...

“... 뱃지도 없으면서”

“... *without badge*”

build

passing

게으름은 인간의 본성.  
나쁜다는 의미가 아닙니다.

**Laziness** is a human nature.  
I didn't mean it's bad.

사람은 게으르기 위해서 **최적화**를 합니다.


Human always do **optimize** to being lazy.

**Travis**가 여러분을 게으르게 만들어 드립니다.

**Travis** drives you to be more lazy person.


# summernote / django-summernote build passing


[Current](#) [Branches](#) [Build History](#) [Pull Requests](#)



More options 


✓ **master** Merge pull request #152 from philliparente/master


using the another comment style to avoid corruption in runtime inline...


 Commit 827d3db


 Compare 98dc8bf..827d3db

 Hyunwoo Park authored  Hyunwoo Park committed













🔗 #222 passed 

 Elapsed time 1 min 52 sec

 Total time 4 min 54 sec

 2 days ago

## Build Jobs

✓ #222.1	  Python: 2.7	 no environment variables set	 1 min 34 sec
✓ #222.2	  Python: 3.4	 no environment variables set	 1 min 28 sec
✓ #222.3	  Python: 3.5	 no environment variables set	 1 min 52 sec

가장 중요한 것은,  
내가 만든 코드의 테스트가 아니라  
다른 사람이 만든 코드의 테스트입니다.

The most important is  
not testing my own code,  
testing codes **from others**.





✓ **PR #152** using the another comment style to avoid corru

phillip

#221 passed

2442af3

4 min 5 sec

11 days ago

✓ **PR #151** Add Codemirror support

Randy Lien

#219 passed

1cc9555

4 min 32 sec

20 days ago

✓ **PR #142** Update widgets.py

Dorian Amouroux

#217 passed

8be7d39

4 min 34 sec

2 months ago

✓ **PR #134** Update README.md

liondancer

#211 passed

6df5270

5 min

4 months ago

✓ **PR #133** Added translated strings in views

Łukasz Kopeć

#209 passed

c107d4e

4 min 37 sec

4 months ago

✓ PR #152 using the another comment style to avoid corru

phillip

#221 passed

4 min 5 sec

11 days ago

이 표에서 보실 수 있듯,

✓ PR #151 Add Codemirror support

Randy Lien

#219 passed

4 min 32 sec

20 days ago

저는 4개월간 PR을 만들지 않았습니다.

✓ PR #142 Update widgets.py

Dorian Amouroux

#217 passed

4 min 34 sec

2 months ago

As you can see here,

✓ PR #134 Update README.md

liondancer

#211 passed

5 min

4 months ago

I didn't make any PR during 4 months.

✓ PR #133 Added translated strings in views

Łukasz Kopeć

#209 passed

4 min 37 sec

4 months ago

c107d4e

이 얘기는, 빌드 자동화가 이루어졌을 때야  
비로서 다른 사람의 코드를 받아들일  
준비가 되었다는 것입니다.

This means, only the project uses  
**automatic build system** have a chance to  
get contributions from outside and others.



✓ PR #152 using the another comment style to avoid corru

phillip

#221 passed

4 min 5 sec

11 days ago

아... 물론 **Travis**가 저를

✓ PR #151 and Collection support

Kandy Lien

#225 passed

4 min 32 sec

20 days ago

게으른 사람으로 만드는데 성공했다는 의미도...

✓ PR #142 Update widgets.py

Dorian Amouroux

#217 passed

4 min 34 sec

2 months ago

Ahhh... Also means **Travis** succeed to

✓ PR #134 Update README.md

liondancer

#211 passed

5 min

4 months ago

make me more lazy...

✓ PR #133 Added translated strings in views

Łukasz Kopeć

#209 passed

4 min 37 sec

4 months ago

테스트의 작성은 기본이고,  
테스트를 자동화해야 합니다.

Writing testing code is pretty basic,  
And we should make it runs  
automatically.

중요하니까 두 번 말씀드립니다.

**Travis**가 여러분을 게으르게 만들어 드립니다.

I usually speak twice for important things.

**Travis** can give you awesome laziness.



## All checks have passed

2 successful checks

[Hide all checks](#)



**continuous-integration/travis-ci/pr** — The Travis CI build passed

[Details](#)



**coverage/coveralls** — Coverage remained the same at 96.875%

[Details](#)



## This branch has no conflicts with the base branch

Merging can be performed automatically.



**Merge pull request**

You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

**Travis**가 없다면, 우리는 모든 커밋에 대해  
직접 테스트하고 코드를 병합해야 합니다.

Without **Travis**, we have to test  
every commit in local, merge manually.



하지만 저는 게을러서 그렇게  
프로젝트를 운영할 수는 없습니다.

But I'm an extremely lazy person,  
I can't manage my project in a such way.

**Travis**가 파이썬과 관련이 없다고 얘기했지만,  
그래도 지원하는 부분이 조금 있습니다.

I told you there is no correlation between  
**Travis** and **Python**, but there are little.

# Building a Python Project

## What This Guide Covers #

This guide covers build environment and configuration topics specific to Python projects. Please make sure to read our [Getting Started](#) and [general build configuration](#) guides first.

Python builds are not available on the OSX environment.

- CHOOSING PYTHON VERSIONS TO TEST AGAINST
- DEFAULT PYTHON VERSION
- SPECIFYING TEST SCRIPT
- DEPENDENCY MANAGEMENT
- NIGHTLY BUILD SUPPORT
- ON-DEMAND INSTALLATIONS
- BUILD MATRIX
- EXAMPLES

<https://docs.travis-ci.com/user/languages/python>

**Travis**에서 테스트 할 파이썬 버전을  
여러 개 선택할 수 있습니다.

In **Travis**, you can choose multiple  
Python runtimes.

**Tox** 에서도 여러 파이썬 버전  
**Travis** 에서도 여러 파이썬 버전

**Tox** supports multiple Pythons  
**Travis** supports multiple Pythons

어?

**What?**

제가 좋은 물건 하나 가지고 왔는데요,

I'm bringing some good for you...

# **tox-travis**

Seamless integration of Tox into Travis CI

<https://github.com/ryanhiebert/tox-travis>









**Tox**에서 여러 설정을 마치고, 그 내용을  
**Travis**에서 그대로 실행할 수 있습니다.

**Tox** handles configurations,  
**Travis** runs it.

```
-----  
py27-dj17: commands succeeded  
py34-dj17: commands succeeded  
py27-dj18: commands succeeded  
py27-dj19: commands succeeded  
py27-djmaster: commands succeeded  
py34-dj18: commands succeeded  
py34-dj19: commands succeeded  
py34-djmaster: commands succeeded  
py35-dj18: commands succeeded  
py35-dj19: commands succeeded  
py35-djmaster: commands succeeded  
congratulations :)  
lqezs-mba:django-summernote lqez$ █
```

로컬 환경에서 tox 를 실행  
Run tox in local environment

## Build Jobs

✓ #222.1	 </> Python: 2.7	 no environment variables set	🕒 1 min 34 sec
✓ #222.2	 </> Python: 3.4	 no environment variables set	🕒 1 min 28 sec
✓ #222.3	 </> Python: 3.5	 no environment variables set	🕒 1 min 52 sec

### summary

```
py27-dj17: commands succeeded
py27-dj18: commands succeeded
py27-dj19: commands succeeded
py27-djmaster: commands succeeded
congratulations :)
```

### summary

```
py34-dj17: commands succeeded
py34-dj18: commands succeeded
py34-dj19: commands succeeded
py34-djmaster: commands succeeded
congratulations :)
```

### summary

```
py35-dj18: commands succeeded
py35-dj19: commands succeeded
py35-djmaster: commands succeeded
congratulations :)
```

tox-travis 를 통해 실행  
Run testing via tox-travis

어쨌든

Anyway

소녀는 의기양양해졌습니다.

She became triumphant.

이제 작지만 번쩍번쩍한 뱃지도 가졌으니까요.

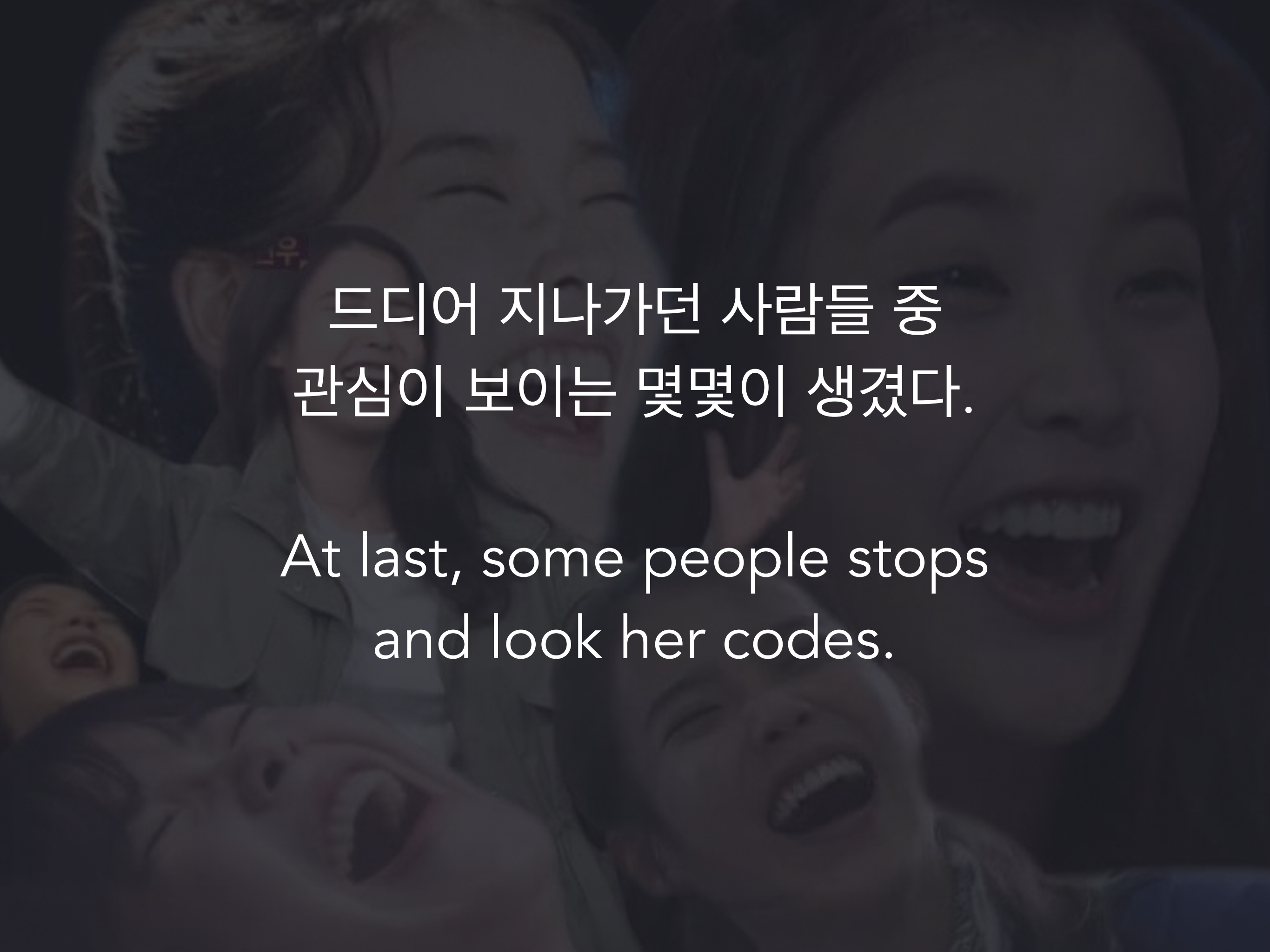
build passing

Now she got a small but brilliant badge.

그래서 소녀는 자신있게  
“이 코드를 써보는데 어때?”

And she says proudly,  
*“Hey, use my code and get profit.”*



A dark, semi-transparent background image showing a crowd of people, likely at a concert or event, with their mouths open as if cheering or singing. The image is used as a backdrop for the text.

드디어 지나가던 사람들 중  
관심이 보이는 몇몇이 생겼다.

At last, some people stops  
and look her codes.



하지만, 이내 절레절레 고개를 흔들고  
지나가 버렸습니다.

But after that,  
They all gone shaking there heads.

소녀는 더 이상 빈손으로 집에 돌아가기 싫어  
따라가서 물어봤습니다.

She doesn't want back home with empty,  
she chased them and asked.

“왜 내 코드를 쓰지 않는거야?”  
대답은 간단했다.

*“Hey, why don’t you use my code once?”*  
And got simple answer.

“커버리지가 없잖아.”

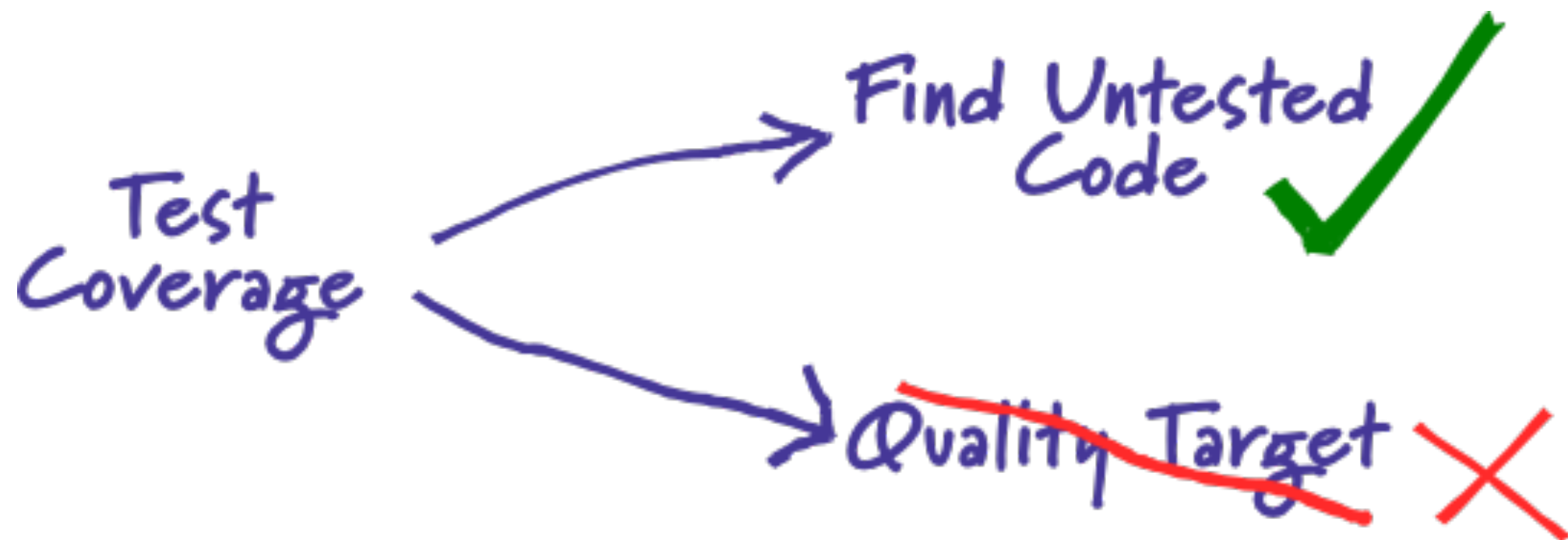
*“No coverage.”*

도대체 커버리지가 뭐길래.

What the hell coverage is.

일찍이 **마틴 파울러**가 말했습니다.

In early, **Martin Fowler** says.



<http://martinfowler.com/bliki/TestCoverage.html>



//

Test coverage is a useful tool for **finding untested parts** of a codebase.

Test coverage is of little use as a numeric statement of **how good your tests** are.

//



코드 커버리지는 목표가 아니라 **상태**입니다.

Code coverage is not purpose,  
it means **status**.

당신의 코드가 얼마나  
건강한지 알려주는 지표입니다.

Code coverage could be  
an **indicator** how healthy code.

별도 많고 커버리지가 높은 코드라면 아무래도  
믿고 쓸 확률이 높겠죠.

It probably has higher chance to be used  
if it has many stars and high coverage.

어디까지나 확률적이지만요.

Yes, only probably.

정말 쓸모 있는 코드 두 개가 있는데,  
한 쪽은 문서화도 테스트도 커버리지도 높고  
다른 쪽은 아니라면 무엇을 선택하시겠습니까?

If there are two codes,  
one has docs, tests with coverage  
and another one has nothing,  
then what will you choose?

그래서 필요합니다.

That's the reason why we do these.

coverage

96%

**Coveralls** 는 바로 이 때 필요한 도구입니다.

It's time to use **Coveralls**.



ALL 11

CHANGED 0

SOURCE CHANGED 0

COVERAGE CHANGED 0

SEARCH:

▲ COVERAGE	◆	◆ FILE	◆ LINES	◆ RELEVANT	◆ COVERED	◆ MISSED	◆ HITS/LINE
<div><div>—</div><div>91.89</div></div>		<a href="#">django_summernote/models.py</a>	68	37	34	3	1.0
<div><div>—</div><div>96.3</div></div>		<a href="#">django_summernote/widgets.py</a>	139	54	52	2	1.0
<div><div>—</div><div>96.43</div></div>		<a href="#">django_summernote/settings.py</a>	139	28	27	1	1.0
<div><div>—</div><div>100.0</div></div>		<a href="#">django_summernote/__init__.py</a>	8	5	5	0	1.0
<div><div>—</div><div>100.0</div></div>		<a href="#">django_summernote/views.py</a>	70	30	30	0	1.0
<div><div>—</div><div>100.0</div></div>		<a href="#">django_summernote/urls.py</a>	9	3	3	0	1.0
<div><div>—</div><div>100.0</div></div>		<a href="#">django_summernote/migrations/__init__.py</a>	0	0	0	0	0.0
<div><div>—</div><div>100.0</div></div>		<a href="#">django_summernote/migrations/0001_initial.py</a>	27	6	6	0	1.0

파일 단위 커버리지  
Coverage report per file

BUILDS	BRANCH	COVERAGE	COMMIT	COMMITTER	TYPE	TIME	VIA
#212	master	96.89	Merge pull request #134 from liondancer/patch-2 Update README.md	lqez	push	26 Apr 2016	travis-ci
#211	master	96.89	Update README.md Missing comma	liondancer	PULL #134	25 Apr 2016	travis-ci
#210	master	96.89	Merge pull request #133 from ktokolwiek/master Added translated strings in views	lqez	push	25 Apr 2016	travis-ci
#209	master	96.89	Added translated strings in views	ktokolwiek	PULL #133	24 Apr 2016	travis-ci
#208	master	96.88	Add note about non-image uploads in README	clauddep	PULL #132	22 Apr 2016	travis-ci
#207	master	96.88	Update README.md	lqez	push	10 Apr 2016	travis-ci
#206	0.8.5	96.88	Redefine settings for css/jss files	lqez	push	10 Apr 2016	travis-ci
#205	master	96.88	Redefine settings for css/jss files	lqez	push	10 Apr 2016	travis-ci
#204	0.8.4	96.88	Support distinct js/css settings for each widget	lqez	push	10 Apr 2016	travis-ci
#203	master	96.88	Support distinct js/css settings for each widget	lqez	push	10 Apr 2016	travis-ci
#202	master	96.88	Merge pull request #126 from webdev4u/master Ability to enable/disable upload images	lqez	push	31 Mar 2016	travis-ci
#201	master	96.88	Added ability to disable image uploading	webdev4u	PULL #126	28 Mar 2016	travis-ci
#200	0.8.3	96.88	Bump to 0.8.3	lqez	push	16 Mar 2016	travis-ci
#199	master	96.88	Bump to 0.8.3	lqez	push	16 Mar 2016	travis-ci
#198	master	96.88	Update README for settings of media files	lqez	push	16 Mar 2016	travis-ci

커밋 단위 커버리지  
Coverage report per commit

**Coveralls**도 게으른 사람들을 위한 도구죠.

**Coveralls** is also a tool for lazy one.

얼마나 간단한지  
설정 파일을 보여드리고 싶지만,

I deathly want to show  
how simple code is,

어쨌든 정말 몇 줄 안됩니다.

Anyway, it's a piece of code.

**Coveralls** 와 **Travis** 를 같이 쓰면  
가장 게으른 프로젝트 관리자가 될 수 있습니다.

With **Coveralls** and **Travis** together,  
you can be the most laziness project  
manager in the world.



**coveralls** commented on Apr 22



**coverage** 70%

Coverage decreased (-0.04%) to 70.32% when pulling [06a3d38](#) on **bebeeteam:patch/backspace-remove-image** into [a120ac7](#) on **summernote:develop**.



### Some checks were not successful

[Hide all checks](#)

1 failing and 1 successful checks



**coverage/coveralls** — Coverage decreased (-0.04%) to 70.32%

[Details](#)



**continuous-integration/travis-ci/pr** — The Travis CI build passed

[Details](#)



### This branch has no conflicts with the base branch

Merging can be performed automatically.



**Merge pull request**

You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

테스트가 통과해도 커버리지를 낮추는 PR은  
**Coveralls**가 알려줍니다.

**Coveralls** will notify you when a commit  
comes with lower coverage even it  
passed the test.



이 이야기는 어떻게 끝날까요?

BTW, how this story ends?

소녀는 깃헙 스타가 되어 파이콘에 올까요?

She would be a Github star and  
come to join PyCon?

사실, 그건 그렇게 중요하지 않습니다.

Actually, it's not the core of this story.

가장 중요한건 이 일을 얼마나  
**꾸준히, 지속적으로** 하는가 입니다.

The most important thing is  
how **steady** and **continuously**.

귀찮고 어려운 일은 계속해서 하기 어렵습니다.

Hard and boring works  
cannot be done continually.

여러분은 왜 아침 일찍 일어나  
여기까지 힘들게 오셨습니까?

What is the real reason  
makes you get up and being here?

싫어하는 일은 오래하기 어렵습니다.

We cannot do hate things  
for a long time.

좋아하는 일은 비교적 오래 할 수 있습니다.

We can keep doing what we like  
during some long time.



하지만 충분하지 않습니다.

But It's not enough.

공고한 오픈소스 커뮤니티를 만들고  
오픈소스 프로젝트를 하기 위해서는  
충분하지 않습니다.

Not enough to  
do OpenSource project and  
make solid OpenSource community.

하지만

But,

지켜야 하는 것이 있으면  
오래 할 수 있습니다.

You can be stronger when you have  
something to **keep and protect**.

무엇을 지킬 지는 사람마다 다릅니다.

It's vary personally.

저는 오픈소스의 **문화**를 지키고 싶습니다.  
특별히 파이썬 공동체와 파이콘에서요.

I want to keep the **culture** of  
OpenSource community,  
especially in Python and PyCon.

그러기 위해서는 견뎌내야 합니다.

To achieve it, we should endure.

하지만, 진보된 도구들의 힘을 빌리면  
더 쉽게 목표를 달성 할 수 있습니다.

Now we have enhanced and modern  
tools to achieve easier and better.

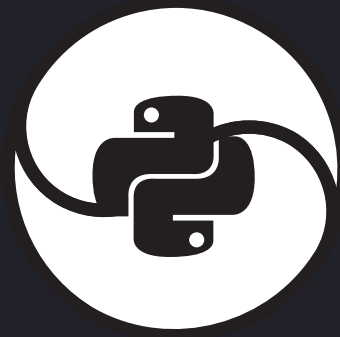


일단 작게 시작하세요.

Begin small.

하지만 계속하세요.

But steady.



# Respect, Diversity

## PyCon APAC 2016

감사합니다.

Thank you.