

# Confessions of Joe Developer

Daniel Roy Greenfeld

# Confessions of Joe Developer

Daniel Roy Greenfeld

Way back in

20004

I had long hair



# Got interview at NASA





**No way I was going to work for NASA**

**No computer science degree  
Only 7 years professional experience**

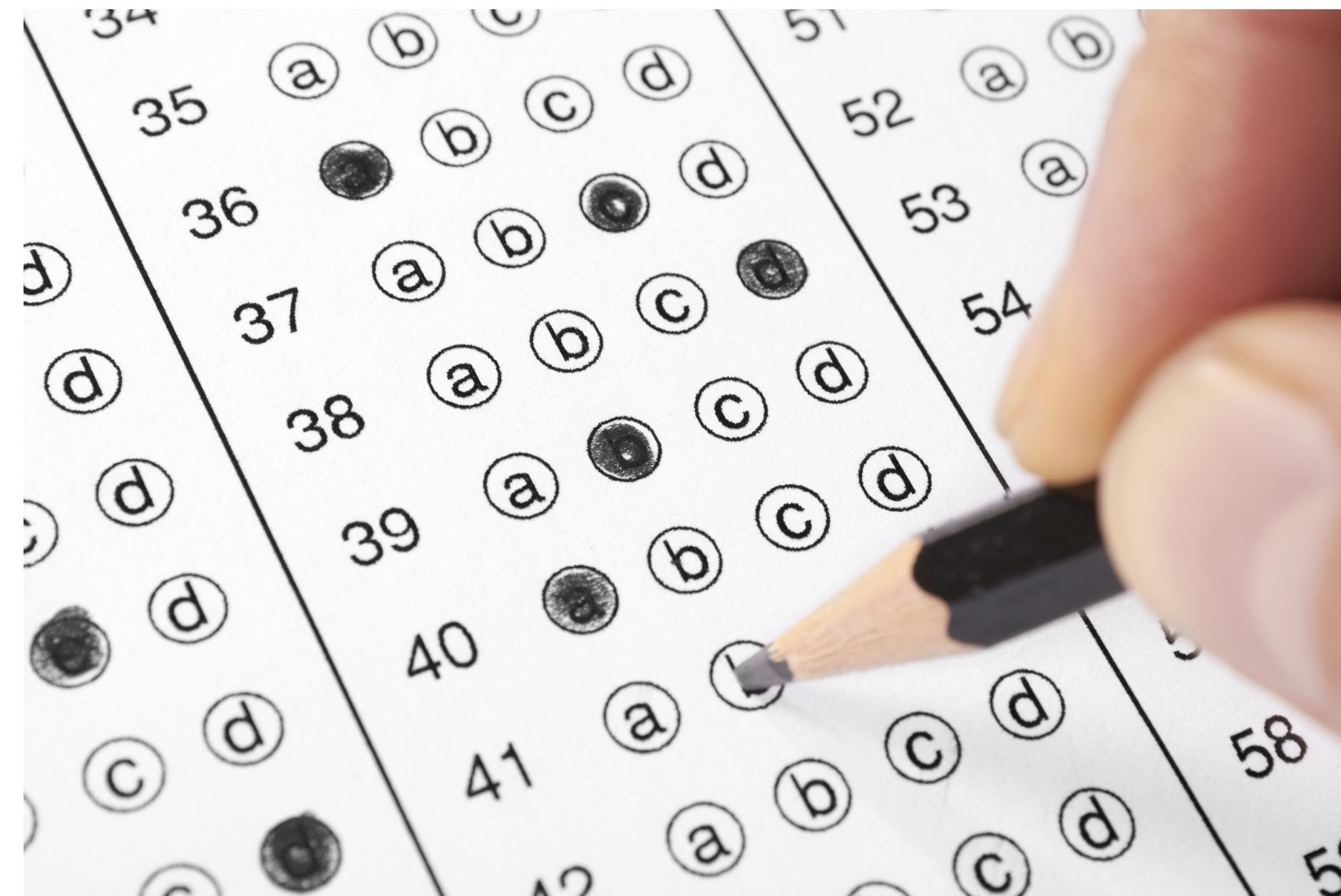
Pia

# Treat interview as a practice session



Still from Jackie Chan's 1984 movie "Meals of Wheels"

# Test new approaches to typical questions



**Have fun!**

How it went

**"What's your biggest weakness?"**

- NASA Interviewer

**"My biggest weakness is that I'm emotionally tied to my work so if I'm not doing well I take it personally. Then I work longer and harder to improve."**

- My normal response

This time:

"I'm stupid and lazy"

# NASA Interviewer

WTF?!

# NASA Interviewer

"You going to explain that?"

I'm stupid

# I'm stupid

- Can't figure things out
- Can't remember things
- Too stupid not to ask stupid questions

Let's go over each item

I'm stupid

Can't figure things out

I'm stupid

Can't figure things out

Always look for libraries first

# Can't figure things out

# OMG argparse

```
import argparse

# sub-command functions
def foo(args):
    print(args.x * args.y)

def bar(args):
    print('(%s)' % args.z)

# create the top-level parser
parser = argparse.ArgumentParser()
subparsers = parser.add_subparsers()

# create the parser for the "foo" command
parser_foo = subparsers.add_parser('foo')
parser_foo.add_argument('-x', type=int, default=1)
parser_foo.add_argument('y', type=float)
parser_foo.set_defaults(func=foo)

# create the parser for the "bar" command
parser_bar = subparsers.add_parser('bar')
parser_bar.add_argument('z')
parser_bar.set_defaults(func=bar)

parser.parse_args()
```

# Can't figure things out

## Yay Typer!

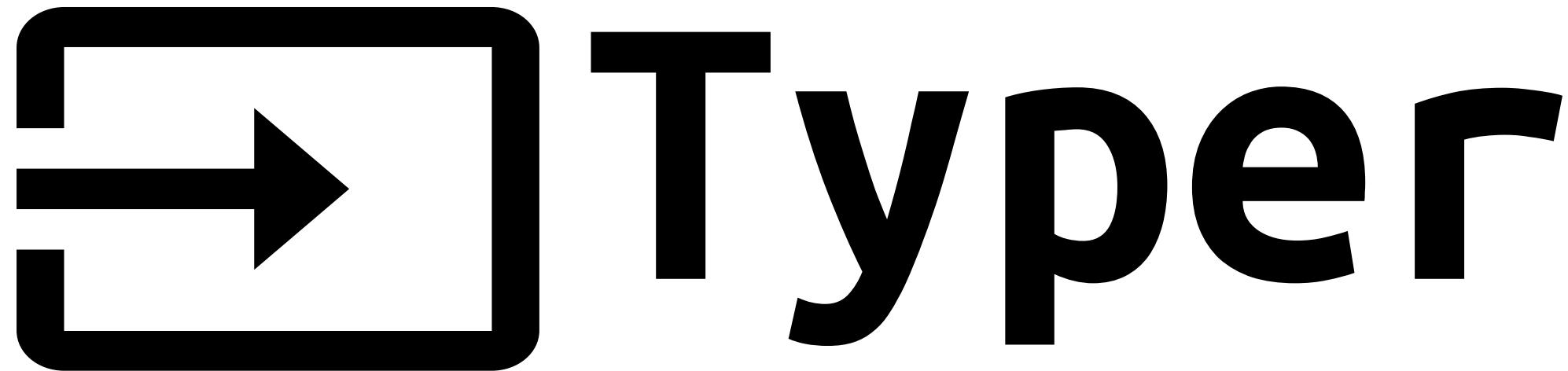
```
import typer

app = typer.Typer()

@app.command()
def foo(y: float, x: int = 1):
    print(x * y)

@app.command()
def bar(z: str):
    print(f'(({z}))')

if __name__ == '__main__':
    app()
```



[typer.tiangolo.com](http://typer.tiangolo.com)

I'm stupid

Too stupid not to ask stupid questions

*hint:*

There are no stupid questions

*hint:*

The only bad question is the one  
you do not ask

*important professional advice:*

Don't try to impress people by not  
asking questions

Ashamed to ask questions?

No one cares

No one remembers

Too stupid not to ask stupid questions

Obey the Thirty Minute Rule

[daniel.feldroy.com/30](http://daniel.feldroy.com/30)



# Obey the Thirty Minute Rule

Don't waste more than 30 minutes on a problem without asking questions

Obey the Thirty Minute Rule

Change to 60 minutes if you like

Important: Don't be a jerk and not let other people ask questions



I'm stupid

Can't remember things

Can't remember things

Docstrings are awesome

Can't remember things

Typehints are awesome too, use  
them to reduce docstrings

# Remember this?

Typehints means we don't have to write so many docs

```
import typer

app = typer.Typer()

@app.command()
def foo(y: float, x: int = 1):
    print(x * y)

@app.command()
def bar(z: str):
    print(f'(({z}))')

if __name__ == '__main__':
    app()
```

Can't remember things

Use markdown because  
of ease and portability

Use markdown because of ease/portability

Even when I wrote daily in RST I had  
to look stuff up

Use markdown because of ease/portability

Markdown is easy to learn

Use markdown because of ease/portability

Reduce friction in your docs!

Markdown is so awesome it has a logo!



Can't remember things

When following a talk or tutorial,

Write down even the slide bullets!

For some people writing down  
notes enhances learning

I'm stupid

Can't remember things

Me back in the day:

- [pydanny-event-notes.readthedocs.io/](http://pydanny-event-notes.readthedocs.io/)
- [daniel.feldroy.com/posts/2011-12-story-of-live-noting](http://daniel.feldroy.com/posts/2011-12-story-of-live-noting)

I'm stupid

Can't remember things

Documentation makes

me look good!

Documentation makes me look good!

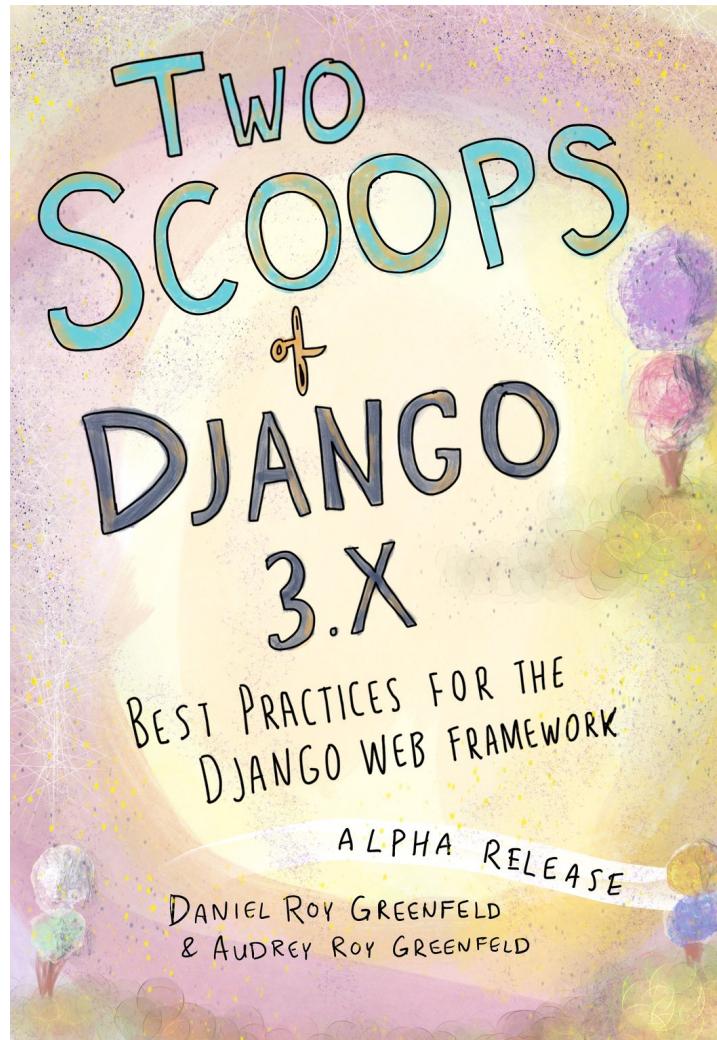
I can pull out old tech details that  
others don't have

Documentation makes me look good!

Can be used later on other projects

# Documentation makes me look good!

## Great for book content



Documentation can you look good too:

[docs.python-requests.org](https://docs.python-requests.org)

[fastapi.tiangolo.com](https://fastapi.tiangolo.com)



**Requests**  
http for humans

 **FastAPI**

!mazy

# I'm lazy

- Don't wanna do anything twice
- Don't wanna debug code when I had it working before
- Don't wanna look hard for docs

Here we go on laziness!

I'm lazy

Don't wanna do anything twice

If I write the same code twice I stick  
in a function

I'm lazy

Don't wanna do anything twice

I use typehints so I write less docs

I'm lazy

Don't wanna do anything twice

I stick the function into a utils  
module

I'm lazy

Don't wanna do anything twice

I put the function on GitHub so I  
don't lose it

I'm lazy

Don't wanna do anything twice

Isn't this the foundation of open  
source?

I'm lazy

Don't wanna debug code that worked before

- Manually testing code by watching it run is hard
- ...and boring
- ...and hence is error prone
- ...meaning you have to do more work

I'm lazy

Don't wanna debug code when I had it working before

Write down how to test lest you forget

## Running the test suite

To run all of the Django Packages tests:

```
docker-compose -f dev.yml run django python manage.py test
```

To run tests for a particular Django Packages app, for example the feeds app:

```
docker-compose -f dev.yml run django python manage.py test feeds
```

I'm lazy

Don't wanna look hard for docs

Every project should have a README that covers:

- Installation
- Running the code
- Testing the code

I'm lazy

Don't wanna look hard for docs

Longer form docs should use:

- [readthedocs.io](#)
- [GitHub pages](#)
- Whatever

Did I get the job at NASA?



**YES**

In hindsight I was lucky

Thank you  
Sharon Campbell

One of my favorite managers ever

**Sharon Campbell**

If you're watching this video, please contact  
me, my company could use your talents

# My history at NASA

- Started in 2005, ended in 2010
- Coded a lot
- Learning Python and Django
- Started what became django-crispy-forms
- Met my wife
- Learned about climate change

# My history at NASA

- Started in 2005, ended in 2010
- **Coded a lot**
- Learning Python and Django
- Started what became django-crispy-forms
- Met my wife
- Learned about climate change

Coded a Lot

Code, Code, Code

[daniel.feldroy.com/posts/code-code-code](https://daniel.feldroy.com/posts/code-code-code)

If you want to get good at anything

**Practice**

Practice what you learn from this event

Use it

Always be typing

Pairing is great

But if you're trying to improve?

Always be the person typing

Taking a class?

Always be typing

No exercises in the class?

Then you are watching  
a spectator sport

You will learn as a spectator  
just not as fast  
and...

Mastery can't be  
found as a spectator

**"You can't become a tech CEO by  
watching the social network"**

You can't win at  
sports by watching TV

To master anything...

To master coding...

... you have to practice

Practice makes perfect

Always be typing

Code, Code, Code

[daniel.feldroy.com/posts/code-code-code](https://daniel.feldroy.com/posts/code-code-code)

# About Me

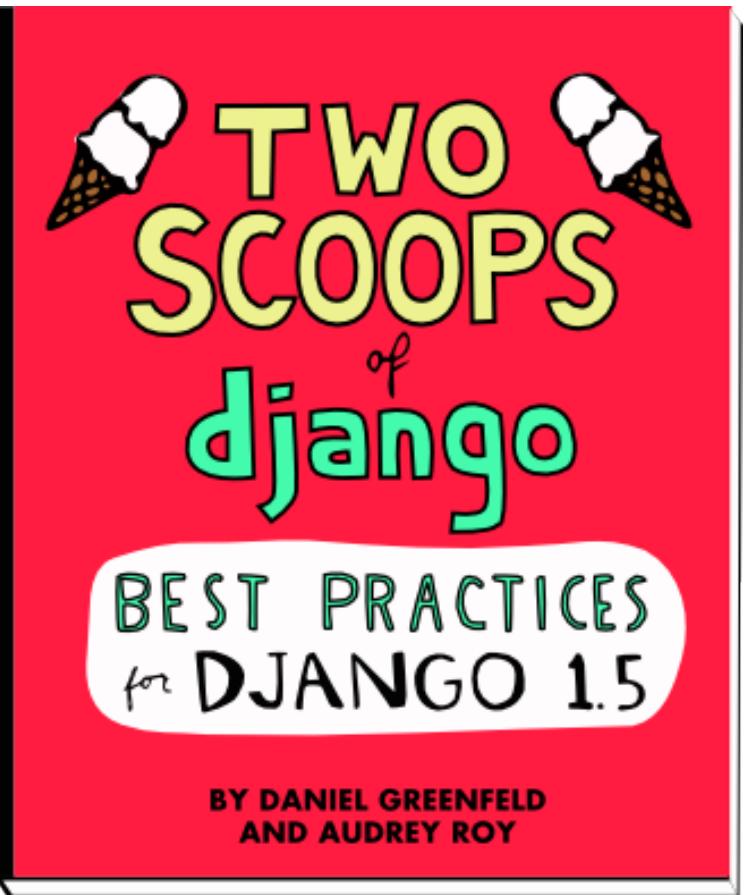
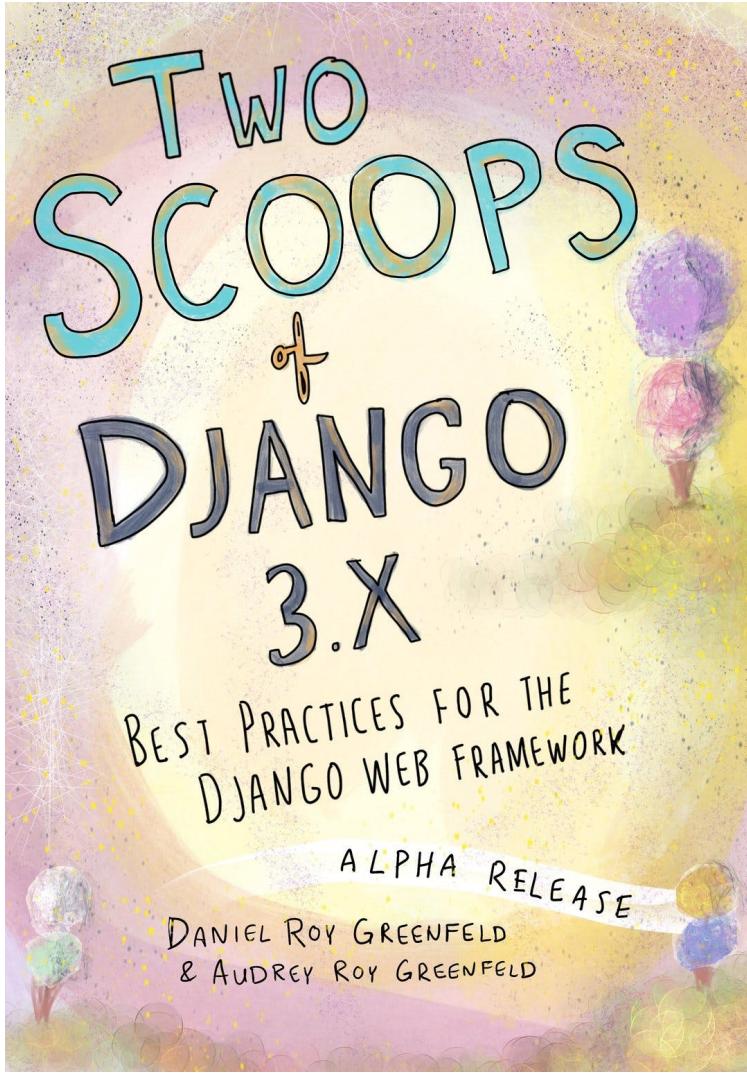


Daniel Roy Greenfeld

AKA "pydanny"

Coder  
Leader  
Speaker

# Author, Husband



# Father



Working to give her the best planet possible

Want to use your skills to fight climate change?



Want to be a planet-saving superhero?

Join me at Octopus Energy. We're hiring!

[octopusenergy.com/careers](https://octopusenergy.com/careers)



# Questions

