

Click

A Pleasure To Write
A Pleasure To Use

Sebastian Vetter
@elbaschid

Slides: <http://bit.ly/pycon-2016-click-slides>

Seb

- [@elbaschid](#)
- Living in Vancouver 🇨🇦
- Backend Engineer at [Mobify](#)
- Co-Organizer of VanPy & DjangoVan

Why This Talk?

Terminology

https://www.flickr.com/photos/chris_schultz/11306328493

Parameter

- Argument
- Option

Argument

- **Mandatory** parameter
- Only values required

What It Looks Like:

```
$ pgcli postgresql://....
```

Option

- **Optional** parameter
- Name and value required

What It Looks Like:

```
$ heroku logs --app my-heroku-app
```

```
$ heroku --help
```

(Sub-)Command

- Nested commands allowed
- Groups sub-commands
- Has options & arguments

What It Looks Like:

```
$ calypso aws --region us-east-1 instances
```


A photograph of a bicycle frame and its various components laid out on a cardboard surface. The frame is black with white and black stripes on the down tube and seat tube. The handlebars are blue. The seat is brown. The pedals are silver. The chainrings are yellow. The chain is silver. The rear wheel hub is silver. The front wheel hub is silver. The text "In Plain Python" is overlaid in the center of the image.

In Plain Python

<https://www.flickr.com/photos/tutam/5008073062/>

Most Basic CLI

```
import sys

if len(sys.argv) <= 1:
    print('You need to give me an argument')
    sys.exit(1)

args = sys.argv[1:]

print('Your argument are: {}'.format(args))
```

The Limitations

- Manual parsing
- No input validation
- No help text formatting

A row of beer taps at a bar. The taps are mounted on a metal rail. The background is a blurred bar scene with people and signs. The text "Choose A Library" is overlaid in the center in a large, white, sans-serif font.

Choose A Library

<https://www.flickr.com/photos/14601516@N00/3778738056>

Your Choices

- optparse
- argparse
- docopt
- and there are more...

optparse

- Standard Library
- Default in Python 2.x
- Deprecated since 3.2

optparse **Example**

```
parser = OptionParser()
parser.add_option("-f", "--file",
                  dest="filename",
                  help="write report to FILE")

parser.add_option("-q", "--quiet",
                  action="store_false",
                  dest="verbose",
                  default=True,
                  help="don't print status messages to stdout")

(options, args) = parser.parse_args()
```

argparse

- Standard Library since 3.2
- Replacement for optparse

argparse Example

```
parser = argparse.ArgumentParser()
parser.add_argument('integers', type=int, nargs='+',
                    help='an integer for the accumulator')
parser.add_argument('--sum', dest='accumulate', action='store_const',
                    const=sum, default=max,
                    help='sum the integers (default: find the max)')

args = parser.parse_args()
```

docopt

- A "documentation first" approach
- Declaring the CLI in docstrings
- Available on PyPI

docopt **Example**

```
def main():  
    """  
    Naval Fate.  
  
    Usage:  
        naval_fate ship new <name>...  
        naval_fate -h | --help  
        naval_fate --version  
  
    Options:  
        -h --help          Show this screen.  
        --version          Show version.  
    """
```

A person is captured in mid-air, jumping joyfully in a field of dry, brown grass. The person is wearing a dark jacket and pants, and their arms are outstretched. The background consists of bare trees and a hazy sky. The text 'Introducing Click' is overlaid in a large, white, bold font across the center of the image.

Introducing Click

<https://www.flickr.com/photos/27587002@N07/7155464950>

click

- Author: Armin Ronacher
- Version 6.x
- <http://click.pocoo.org/6/>

click

- Intuitive
- Nestable and composable
- Better handling of input and output
- Why Click?



Let's Build A CLI

<https://www.flickr.com/photos/31381897@N03/3967634792>

Our Example



<https://www.flickr.com/photos/judy-van-der-velden/14687818499>

A man with glasses and a beard, wearing a dark t-shirt and jeans, is kneeling in a server room. He is using a screwdriver to work on a network switch or patch panel. The room is filled with server racks and a dense network of colorful cables (red, blue, green, yellow) running vertically and horizontally. The lighting is somewhat dim, and the overall scene is a technical environment.

Setting Up A CLI

<https://www.flickr.com/photos/sveinhal/2416609728>

```
pip install click
```

On Github

- Click Template: <http://bit.ly/click-cookiecutter>
- Example Code: bit.ly/pycon-2016-click-example

Run it!

```
$ # Run it
```

```
$ ad_notifier
```

```
I am the ad_notifier CLI
```

Decorator-based approach

```
# ad_notifier/cli.py
import click

@click.command()
def main():
    print('I am the ad_notifier CLI')
```

Specify a URL

Run it!

```
$ ad_notifier "http://www.pinkbike.com/buysell/..."  
Processing URL: http://www.pinkbike.com/buysell/list/...  
Found 20 ads!
```



Arguments

<https://www.flickr.com/photos/alper/2781645509>

Add an argument

```
# ad_notifier/cli.py

@click.command()
@click.argument('url')
def main(url):
    print('Processing URL:', url)

    ads = find_ads(url)

    print('Found {} ads!'.format(len(ads)))
```

Send an Notification Email

Run it!

```
$ ad_notifier "http://..." --email my@email.com
```

```
Processing URL: http://...
```

```
Found 20 ads!
```

```
Sending email to my@email.com
```

A grid of many small baskets containing various patterned items, likely fabric or paper, with price tags. The baskets are arranged in rows and columns, and the items inside are colorful and diverse in pattern. The word "Options" is overlaid in the center in a large, white, sans-serif font.

Options

<https://www.flickr.com/photos/photographingtravis/15427838493>

Add Email Option

```
@click.command()
@click.argument('url')
@click.option('--email', envvar='EMAIL_ADDRESS')
def main(url, email):
    ...

    if email and new_ads:
        send_email(email, ads)
```

Handle New Ads Only

Run it!

```
$ ad_notifier "http://..."
```

```
...
```

```
Found 0 ads!
```

```
$ ad_notifier "http://..." --reset-cache
```

```
...
```

```
Found 20 ads!
```

A large field of various national flags flying on tall poles against a blue sky. The flags are arranged in rows and are waving in the wind. The colors are vibrant and varied, including red, blue, green, yellow, and white. The word "Flags" is written in large, white, bold letters across the center of the image.

Flags

<https://www.flickr.com/photos/pennstatelive/16216912657>

Reset the cache

```
@click.command()
...
@click.option('--reset-cache', default=False, is_flag=True)
def main(url, email, reset_cache):
    print('Processing URL:', url)

    new_ads = find_new_ads(url, reset_cache)

    print('Found {} ads!'.format(len(new_ads)))
    ...
```

Make it a Periodic Task

Run it!

```
$ # With scheduler
```

```
$ ad_notifier <URL> run_periodically \  
    --run-every 5 --email my@email.com
```

```
$ # Once
```

```
$ ad_notifier <URL> run_once \  
    --email my@email.com
```

A soldier in camouflage gear is shown in a field, aiming a rifle. The soldier is wearing a helmet with a night vision device and has a focused expression. The background shows a blurred landscape with hills and a body of water. The text "(Sub-)Commands" is overlaid in large white letters across the center of the image.

(Sub-)Commands

(<https://www.flickr.com/photos/soldiersmediacenter/5754070333>)

Add a run & schedule command

```
@click.group()  
def main():  
    pass
```

Add the run subcommands

```
@main.command()  
...  
def run_once(url, email, reset_cache):  
    process_url(url, email, reset_cache)
```

Add schedule subcommand

```
@main.command()
...
@click.option('--run-every', default=5, type=int)
def run_periodically(..., run_every):
    schedule.every(run_every).minutes.do(...)

    while True:
        schedule.run_pending()
        time.sleep(1)
```

Global Settings



Context

<https://www.flickr.com/photos/chanceprojects/16286089602>

Using the Context obj

```
@click.group()
@click.argument('url')
@click.option('--email')
@click.option('--reset-cache', ...)
@click.pass_context
def main(context, url, email, reset_cache):
    context.obj = {
        'url': url,
        'email': email,
        'reset_cache': reset_cache}
```

Updating Sub-Commands

```
@main.command()  
@click.pass_context  
def run_once(context):  
    process_url(**context.obj)
```

Add schedule subcommand

```
@main.command()
@click.pass_context
def run_periodically(context):
    schedule.every(run_every).minutes.do(
        process_url, **context.obj)

    while True:
        schedule.run_pending()
        time.sleep(1)
```

Improve Documentation

Run it!

```
$ ad_notifier
```

```
Usage: ad_notifier [OPTIONS] URL COMMAND [ARGS]...
```

Check pinkbike ads **for** URL and (optionally) send email notification.

Options:

```
--email TEXT      email to send notifications to
--reset-cache     reset the internal ads cache
--help           Show this message and exit.
```

Commands:

```
run_once          Run check for new ads once.
run_periodically  Run scheduler to check for new ads...
```

A man with a beard and sunglasses, wearing a vibrant, multi-colored striped shirt, is shown in profile, focused on operating a vintage movie camera. The camera is mounted on a tripod and has a white protective cap over the lens. The scene is set outdoors at dusk or dawn, with a dark, cloudy sky and a sandy beach in the background. The overall mood is artistic and nostalgic.

Documentation

<https://www.flickr.com/photos/nicoyogui/3898808599>

Adding help text

```
@click.option('--reset-cache', default=False, is_flag=True,
              help='reset the internal ads cache')
@click.pass_context
def main(context, url, email, reset_cache):
    """
    Check pinkbike ads for URL and (optionally) send email notification.
    """
```


Wasn't That A Pleasure?

- Intuitive usage of the decorators
- Nesting/grouping of commands
- Easy validation of input

Also Check Out

- [Environment Variables](#)
- [Parameter Types](#)
- [Testing](#)
- [Bash Autocomplete](#)



Questions?

<https://www.flickr.com/photos/thomashawk/9058066607>

Click

A Pleasure To Write

A Pleasure To Use

Sebastian Vetter

[@elbaschid](#)

Slides: <http://bit.ly/pycon-2016-click-slides>