



Empowering Snake Charmers-in-Training

Stefan van der Merwe

PyConZA 2022



MEET THE SNAKE CHARMER

The ego trip



Meet the snake charmer



- Software Engineer by title
- Electronic Engineer by training
- Passionate about empowering people



Stefan van der Merwe

stefanvdm-em

Follow

 Ex Mente Technologies

 ex-mente.co.za



YOU'RE OUT OF YOUR MIND!

Ex Mente Technologies



- Pyrometallurgical consultants
 - Thermochemistry
 - Multiphysics modelling
- Advancement of society
- Advancing through insight





THERE IS A SNAKE IN MY FURNACE

The role of Python in Ex Mente projects

The role of Python in Ex Mente



-
- **Mixing Metals with Snakes**
(Johan Zietsman, PyConZA 2017)
 - **ChemApp for Python**
(GTT Technologies)
 - **EMThermo**
(Ex Mente Technologies)



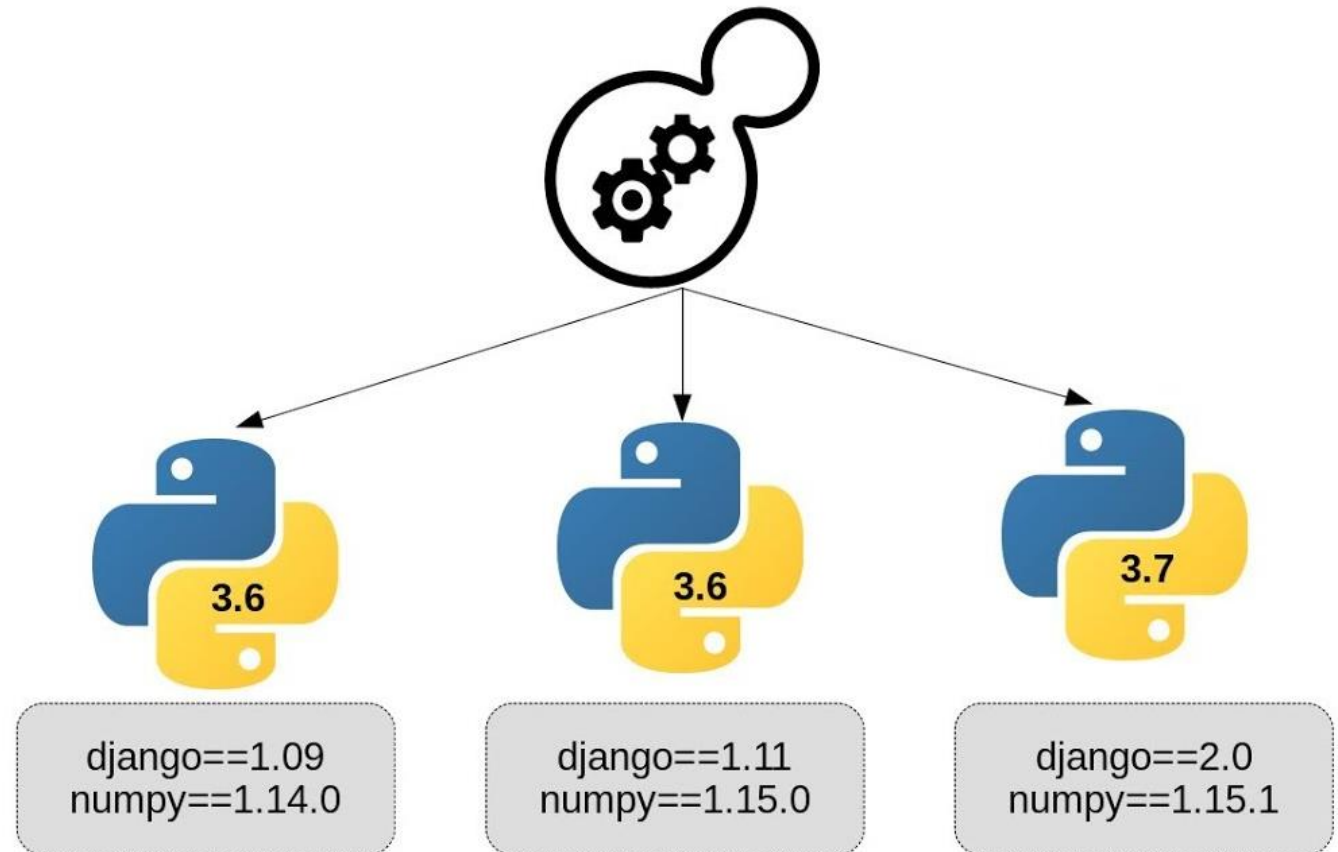
KEEP THAT ANACONDA IN A CAGE!

The role of contained environments in Ex Mente projects

The role of contained environments in Ex Mente



- Environmental control
- Separation
- Reproducible
- Transferrable

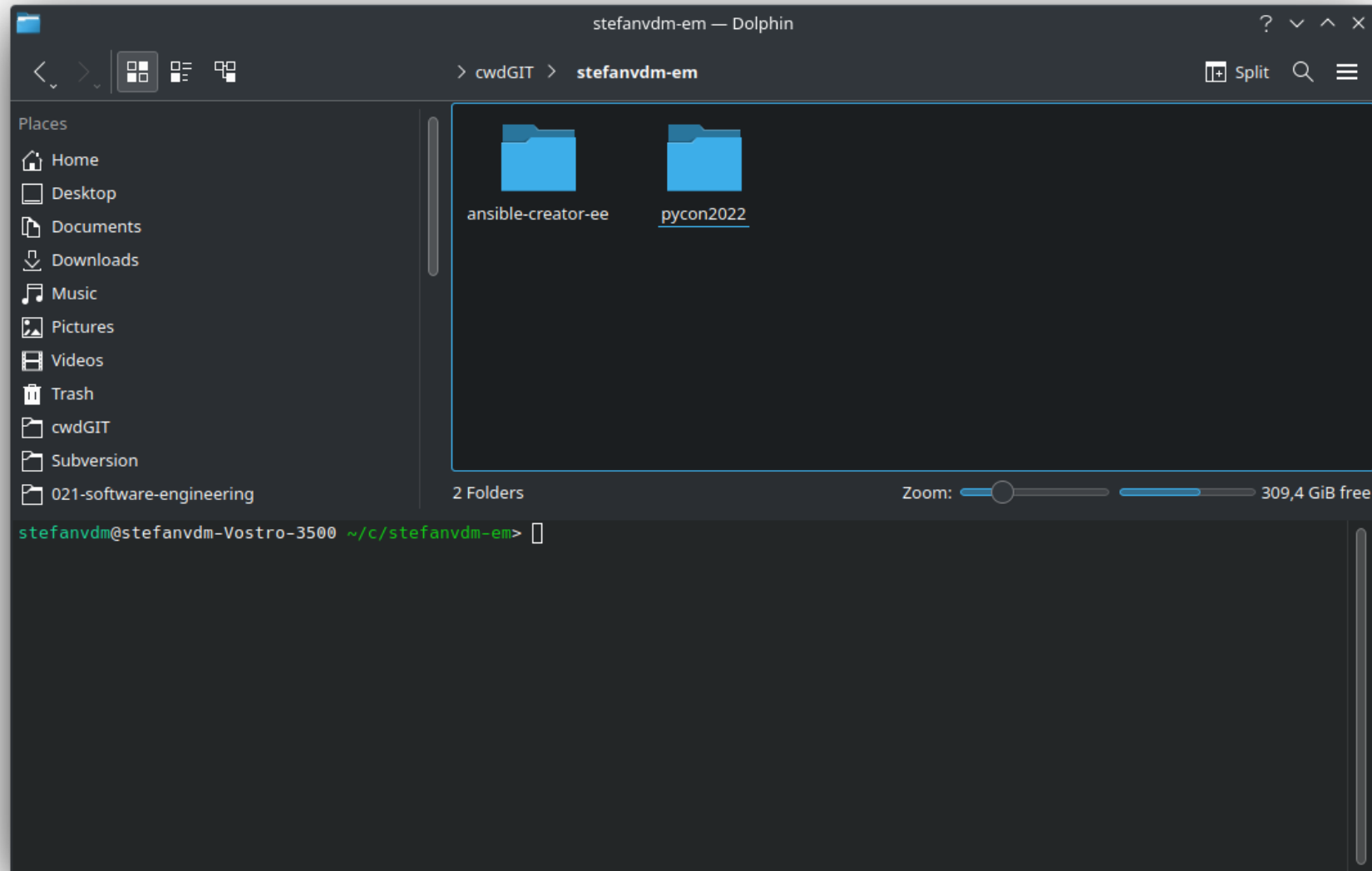




CHARMING THAT SNAKE

The role of CLI apps in Ex Mente projects

Dolphins are smart...



The role of CLI apps in Ex Mente

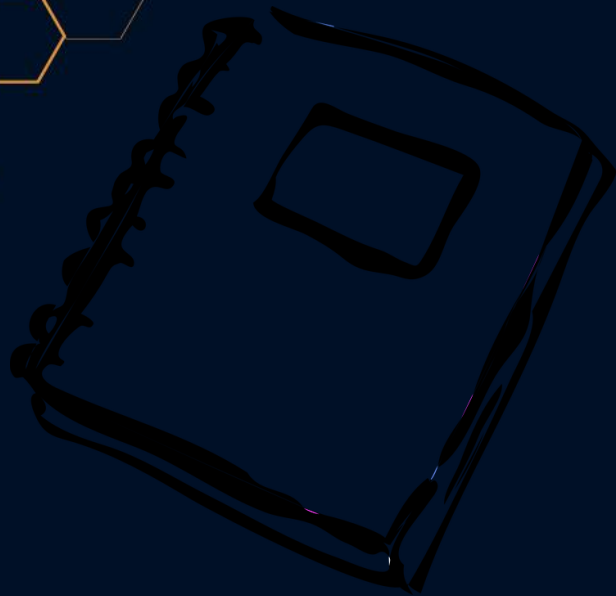


- The general problem: repetitive tasks
- The example problem:
 - Create new report from template
 - Requirements:
 - CLI app
 - Two arguments:
 - Template (required)
 - Destination directory (optional)
 - Accessible language



DIARY OF A SNAKE CHARMER

A developers journey with CLI apps



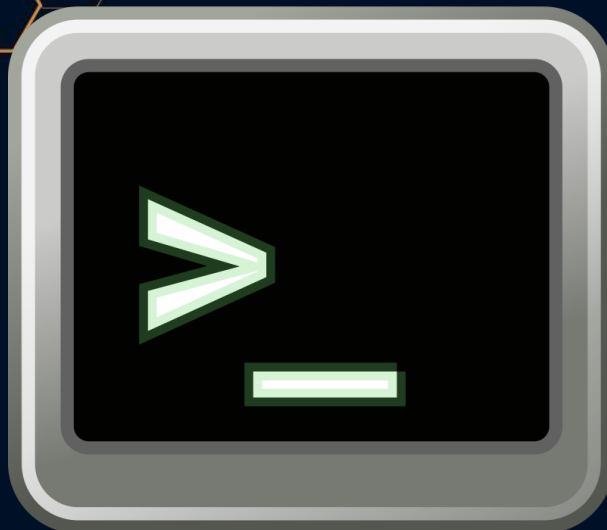
There be personal opinions here...





THE TERMINAL LIFE

Terminal commands



Terminal



```
~ : lxc — Konsole
File Edit View Bookmarks Plugins Settings Help
stefanvdm@pyconza:~/ex-mente/client/somecompany/projects/investigations/103-thermcalc$ pwd
/home/stefanvdm/ex-mente/client/somecompany/projects/investigations/103-thermcalc
stefanvdm@pyconza:~/ex-mente/client/somecompany/projects/investigations/103-thermcalc$ cp ~/somepath/where/i/can/always/find/my/templates/
20220627-kick-off-internal.mm          personal-development-plan/
_tex-shared/                          personal-development-plan.zip
abstract-new-template/                presentation-beamer/
analytical-service-request/           presentation-template/
common-content/                       presentation-template-new-branding/
computational-model-manual/            presentation-template-powerpoint/
computational-model-manual-mindmap/    procedure/
curriculum-vitae/                     project-review-report/
investigation-report/                  project-status-report/
investigation-report-new-branding/     project-status-report-new-branding/
kick-off-meeting/                      quote-approval/
kick-off-meeting-mindmap-external/     report/
kick-off-meeting-mindmap-internal/     report.zip
leave-application-form/                request-for-information/
letterhead/                            risk-assessment/
letterhead.zip                         sample-analysis-request/
mandate-agreement/                     scenario-simulations-report/
mathematical-model-manual/             short-proposal/
meeting-agenda-minutes/                 short-proposal-new-branding/
meeting-agenda-minutes.zip             site-and-travel-claim-form/
memo/                                   templates/
mindmaps/                              work-authorisation/
multiphysics-equations-and-nomenclature/
stefanvdm@pyconza:~/ex-mente/client/somecompany/projects/investigations/103-thermcalc$ cp -r ~/somepath/where/i/can/always/find/my/templates/investigation-report-new-branding report
stefanvdm@pyconza:~/ex-mente/client/somecompany/projects/investigations/103-thermcalc$ ll
total 2
```




LOOK MOM! NO FILE MANAGER!

Bash scripting



Bash



```
start-document.sh - pycon2022 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
start-document.sh U X
bash > start-document.sh
1  #!/bin/bash
2
3  nargs=$#
4
5  case $nargs in
6  1)
7      template=$1
8      dest="tex"
9      ;;
10 2)
11     template=$1
12     dest=$2
13     ;;
14 *)
15     echo "Usage: $0 <TEMPLATE> [DEST]"
16     exit 1
17     ;;
18 esac
19
20 echo "Template: $template"
21 echo "Destination: $dest"
22 |
```

Bash



```
em-install-tl - pycon2022 - Visual Studio Code
File Edit Selection View Go Run Terminal Help

em-install-tl U x
bash > em-install-tl
1  #!/bin/bash
2
3  Help() {
4      echo 'Usage: em-install-tl [-h] [-s] [-r] [-v VERSION]'
5      echo
6      echo 'Ex Mente TeX Live installion'
7      echo
8      echo 'This script will attempt to uninstall any apt packages'
9      echo 'related to texlive and install the selected version of'
10     echo 'TeX Live from the local mirror.'
11     echo
12     echo 'NOTE: Even though the local mirror installation is more'
13     echo 'reliable than the online installer, it will still take'
14     echo 'a long time to complete (approximately 30 min).'
15     echo
16     echo 'Options:'
17     echo '-s          Do not create symbolic links to new installation.'
18     echo '-r          RSYNC (mirror to this system before install) - EXPERIMENTAL.'
19     echo '-v VERSION  Version number.'
20     echo '-h          Show this help message.'
21 }
22
23 SYMLINK=true
24 VERSION_PROVIDED=false
25 RSYNC=false
```



PYTHON HAS ENTERED THE CHAT

Python: sys.argv



Python: sys.argv

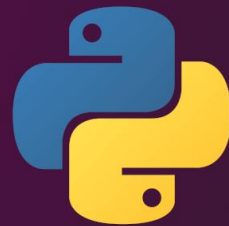


```
start-document.py - pycon2022 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
start-document.py U X
pungi-sys.argv > pungi > start-document.py > ...
1 import sys
2
3 if len(sys.argv) == 2:
4     template = sys.argv[1]
5     dest_dir = "tex"
6 elif len(sys.argv) == 3:
7     template = sys.argv[1]
8     dest_dir = sys.argv[2]
9 else:
10    print("Usage: start-document.py <TEMPLATE> [DEST]")
11    sys.exit(1)
12
13
14 print("Template: {}".format(template))
15 print("Destination: {}".format(dest_dir))
16
```



I'LL TELL YOU WHAT I WANT

Python: argparse



argparse ()

`program.py -x arg1 -y arg2`

Python: argparse



```
start-document.py - pycon2022 - Visual Studio Code
File Edit Selection View Go Run Terminal Help

start-document.py U X
pungi-argparse > pungi > start-document.py > ...
1  import argparse
2
3  parser = argparse.ArgumentParser(description="Start a new document from a template.")
4
5  parser.add_argument(
6      "template",
7      metavar="TEMPLATE",
8      type=str,
9      help="The template to use.",
10 )
11 parser.add_argument(
12     "--dest",
13     dest="dest_dir",
14     metavar="DEST",
15     type=str,
16     default="tex",
17     help="The destination directory to create the document in.",
18 )
19
20 args = parser.parse_args()
21 print("Template: {}".format(args.template))
22 print("Destination: {}".format(args.dest_dir))
23
```



```
$ click _
```

JUST CLICK IT

Python: Click
Command Line Interface Creation Kit

Python: Click



```
start-document.py - pycon2022 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
start-document.py U x
pungi-click > pungi > start-document.py > ...
1 import click
2
3
4 def start_document(template: str, dest: str):
5     """Start a new document from a template."""
6     print("Template: {}".format(template))
7     print("Destination: {}".format(dest))
8
9
10 if __name__ == "__main__":
11     start_document()
12
```

Python: Click



```
start-document.py - pycon2022 - Visual Studio Code
File Edit Selection View Go Run Terminal Help

start-document.py U x
pungi-click > pungi > start-document.py > ...
1  import click
2
3
4  @click.command()
5  def start_document(template: str, dest: str):
6      """Start a new document from a template."""
7      print("Template: {}".format(template))
8      print("Destination: {}".format(dest))
9
10
11 if __name__ == "__main__":
12     start_document()
13
```

Python: Click



```
start-document.py - pycon2022 - Visual Studio Code
File Edit Selection View Go Run Terminal Help

start-document.py U X
pungi-click > pungi > start-document.py > ...
1  import click
2
3
4  @click.command()
5  @click.argument("template")
6  def start_document(template: str, dest: str):
7      """Start a new document from a template."""
8      print("Template: {}".format(template))
9      print("Destination: {}".format(dest))
10
11
12 if __name__ == "__main__":
13     start_document()
14
```

Python: Click



```
start-document.py - pycon2022 - Visual Studio Code
File Edit Selection View Go Run Terminal Help

start-document.py U x
pungi-click > pungi > start-document.py > ...
1  import click
2
3
4  @click.command()
5  @click.argument("template")
6  @click.option(
7      "--dest",
8      default="tex",
9      help="The destination directory to create the document in.",
10 )
11 def start_document(template: str, dest: str):
12     """Start a new document from a template."""
13     print("Template: {}".format(template))
14     print("Destination: {}".format(dest))
15
16
17 if __name__ == "__main__":
18     start_document()
19
```



BUT TYPER IS MORE FUN!

Python: Typer



Typer

Python: Typer



```
start-document.py - pycon2022 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
start-document.py U X
pungi-typer > pungi > start-document.py > ...
1 import typer
2
3
4 def start_document(template: str, dest: str = "tex"):
5     """Start a new document from a template."""
6     print("Template: {}".format(template))
7     print("Destination: {}".format(dest))
8
9
10 if __name__ == "__main__":
11     typer.run(start_document)
12
```

Python: Typer



```
start-document.py - pycon2022 - Visual Studio Code
File Edit Selection View Go Run Terminal Help

start-document.py U X
pungi-typer > pungi > start-document.py > ...
1  import typer
2
3
4  def start_document(
5      template: str = typer.Argument(..., help="Template name."),
6      dest: str = typer.Option("tex", help="Destination directory."),
7  ):
8      """Start a new document from a template."""
9      print("Template: {}".format(template))
10     print("Destination: {}".format(dest))
11
12
13  if __name__ == "__main__":
14     typer.run(start_document)
15
```



BUT WAIT, THERE'S MORE!

Multiple commands in Typer

Multiple commands in Typer



```
Visual Studio Code
pungi.py - pycon2022 - Visual Studio Code
File Edit Selection View Go Run Terminal Help

pungi.py U x
pungi-typer > pungi > pungi.py > ...
1 import typer
2
3 app = typer.Typer()
4
5
6 @app.command("start-document")
7 def start_document(
8     template: str = typer.Argument(..., help="Template name."),
9     dest: str = typer.Option("tex", help="Destination directory."),
10 ):
11     """Start a new document from a template."""
12     print("Template: {}".format(template))
13     print("Destination: {}".format(dest))
14
15
16 @app.command("start-project")
17 def start_project(
18     template: str = typer.Argument(..., help="Template name."),
19     dest: str = typer.Option("tex", help="Destination directory."),
20 ):
21     """Start a new project from a template."""
22     print("Template: {}".format(template))
23     print("Destination: {}".format(dest))
24
25
```



THE PRETTY ONE

CLI apps do not have to be ugly or boring

Pretty CLI apps



```
pycon2022
File Edit View Bookmarks Plugins Settings Help
pycon2022 : pipenv
(pycon2022) stefanvdm@stefanvdm-Vostro-3500 ~/c/s/p/p/pungi (main)> python start-document.py --help
Usage: start-document.py [OPTIONS] TEMPLATE

Start a new document from a template.

Options:
  --dest TEXT  The destination directory to create the document in.
  --help      Show this message and exit.
(pycon2022) stefanvdm@stefanvdm-Vostro-3500 ~/c/s/p/p/pungi (main)> 
```

Pretty CLI apps



pipenv — Konsole



pycon2022 : pipenv



```
(pycon2022) stefanvdm@stefanvdm-Vostro-3500 ~/c/s/p/p/pungi (main)> python start-document.py --help
```

Usage: start-document.py [OPTIONS] TEMPLATE

Start a new document from a template.

Arguments

* template **TEXT** Template name. [default: None] [required]

Options

--dest **TEXT** Destination directory. [default: tex]
--install-completion Install completion for the current shell.
--show-completion Show completion for the current shell, to copy it or customize the installation.
--help Show this message and exit.

```
(pycon2022) stefanvdm@stefanvdm-Vostro-3500 ~/c/s/p/p/pungi (main)> 
```

Pretty CLI apps



```
pycon2022: pipenv — Konsole
File Edit View Bookmarks Plugins Settings Help
pycon2022: pipenv
(pycon2022) stefanvdm@stefanvdm-Vostro-3500 ~/c/s/p/p/pungi (main)> python start-document.py --help
Usage: start-document.py [OPTIONS] TEMPLATE

Start a new document from a template.

Options:
  --dest TEXT  The destination directory to create the document in.
  --help      Show this message and exit.
(pycon2022) stefanvdm@stefanvdm-Vostro-3500 ~/c/s/p/p/pungi (main)>

pycon2022: pipenv
(pycon2022) stefanvdm@stefanvdm-Vostro-3500 ~/c/s/p/p/pungi (main)> python start-document.py --help

Usage: start-document.py [OPTIONS] TEMPLATE

Start a new document from a template.

Arguments
  * template      TEXT  Template name. [default: None] [required]

Options
  --dest          TEXT  Destination directory. [default: tex]
  --install-completion  Install completion for the current shell.
  --show-completion    Show completion for the current shell, to copy it or customize the
                        installation.
  --help            Show this message and exit.

(pycon2022) stefanvdm@stefanvdm-Vostro-3500 ~/c/s/p/p/pungi (main)>
```



TL;DR: (or listen?)

Summary

Summary



PRO	CON
sys.argv and argparse	
<ul style="list-style-type: none">• No external dependencies	<ul style="list-style-type: none">• Cumbersome• Not always straight forward
Click	
<ul style="list-style-type: none">• Highly configurable	<ul style="list-style-type: none">• Too many steps per configuration
Typer	
<ul style="list-style-type: none">• User and developer friendly	<ul style="list-style-type: none">• Incomplete documentation

How do you choose?



- External dependencies
- Developer experience
- User experience



THANK YOU FOR LISTENING!