DPUG: Argparse Demo

Viktoria Haghani

LaSalle Lab | Korf Lab

December 2, 2024

SCAN ME

What is `argparse`?

- `argparse` is a Python module that allows you to set up command-line arguments (i.e. flags) for your scripts

```
#!/usr/bin/env python3

import argparse

#####################
## Set Up Argparse ##
#####################

# Initialize argparse
parser = argparse.ArgumentParser(
    description='Print any number of smiley emoticons on the same or a different line')

'''
# Required arguments
parser.add_argument('--num', required=True, type=int,
    metavar='<int>', help='Required integer argument to choose number of smiley emoticons to print')
'''

# Optional arguments
parser.add_argument('--num', required=False, type=int, default = 5,
    metavar='<int>', help='Integer argument to choose number of smiley emoticons to print, default is [%(default)i]')

# Switch
parser.add_argument('--same', action="store_true",
   help='If used, the smiley emoticon are printed on the same line instead of different lines')

# Finalization of argparse
arg = parser.parse_args()

if arg.same:
    a = ':) '
    b = a * arg.num
else:
    a = ':) \n'
    b = a * arg.num
print(b)
```

Why use `argparse`?

```
vhaghani@DESKTOP$ python3 smiley_argparse.py --help
usage: smiley_argparse.py [-h] [--num <int>] [--same]

Print any number of smiley emoticons on the same or a different line

options:
  -h, --help   show this help message and exit
  --num <int>  Integer argument to choose number of smiley emoticons to print, default is [5]
  --same       If used, the smiley emoticon are printed on the same line instead of different
lines
```

`argparse` Set Up

```
vhaghani@DESKTOP$ touch smiley_argparse.py
```

```
#!/usr/bin/env python3

import argparse

#####################
## Set Up Argparse ##
#####################

# Initialize argparse
parser = argparse.ArgumentParser(
    description='Print any number of smiley emoticons on the same or a different line')

# Finalization of argparse
arg = parser.parse_args()
```

Required Arguments

```
#!/usr/bin/env python3

import argparse

####################
## Set Up Argparse ##
####################

# Initialize argparse
parser = argparse.ArgumentParser(
    description='Print any number of smiley emoticons on the same or a different line')

# Required arguments
parser.add_argument('--num', required=True, type=int,
    metavar='<int>', help='Required integer argument to choose number of smiley emoticons to print')

# Finalization of argparse
arg = parser.parse_args()

a = ':) '
b = a * arg.num
print(b)
```

```
vhaghani@DESKTOP$ python3 smiley_argparse.py --num 8
:) :) :) :) :) :) :) :)

vhaghani@DESKTOP$ python3 smiley_argparse.py --num 100
:) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :)
:) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :)
:) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :) :)
:) :) :) :)
```

Optional Arguments

```
#!/usr/bin/env python3

import argparse

####################
## Set Up Argparse ##
####################

# Initialize argparse
parser = argparse.ArgumentParser(
    description='Print any number of smiley emoticons on the same or a different line')

# Optional arguments
parser.add_argument('--num', required=False, type=int, default = 5,
    metavar='<int>', help='Integer argument to choose number of smiley emoticons to print, default is
[%(default)i] ')

# Finalization of argparse
arg = parser.parse_args()

a = ':) '
b = a * arg.num
print(b)
```

```
vhaghani@DESKTOP$ python3 smiley_argparse.py --num 5
:) :) :) :) :)

vhaghani@DESKTOP$ python3 smiley_argparse.py
:) :) :) :) :)
```

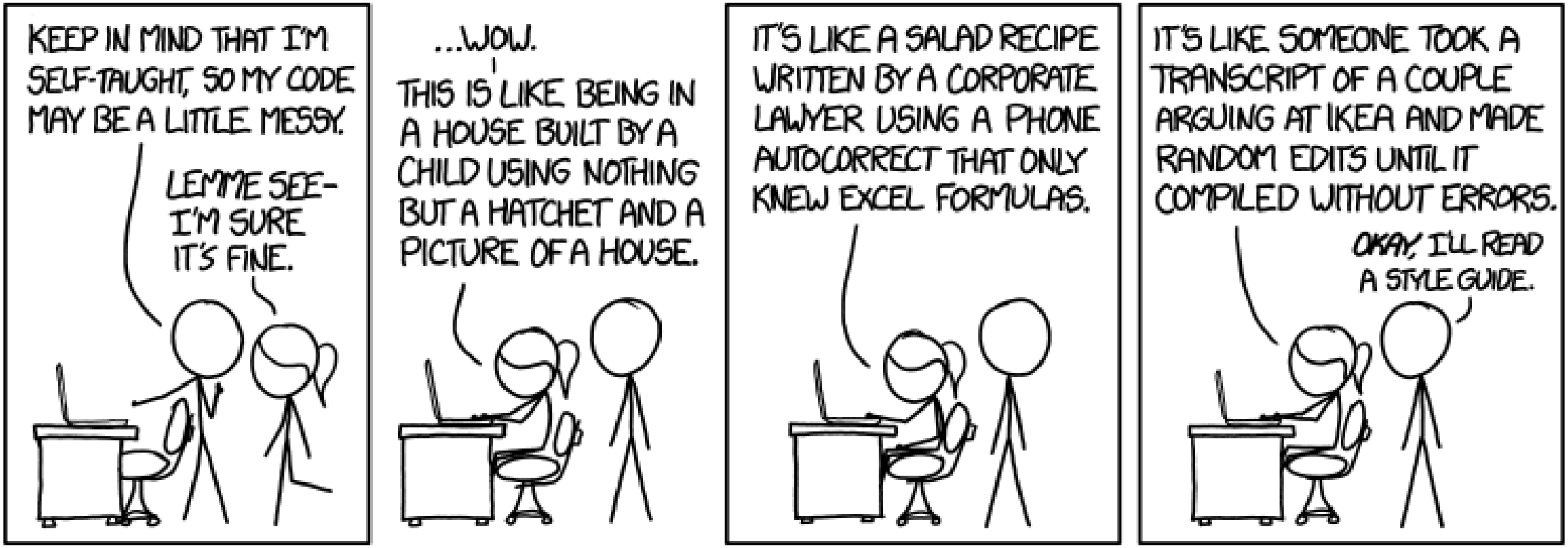
Switches

```
#!/usr/bin/env python3

import argparse

#####################
## Set Up Argparse ##
#####################

# Initialize argparse
parser = argparse.ArgumentParser(
    description='Print any number of smiley emoticons on the same or a different line')

# Optional arguments
parser.add_argument('--num', required=False, type=int, default = 5,
    metavar='<int>', help='Integer argument to choose number of smiley emoticons to print, default is [%(default)i] ‘)

# Switch
parser.add_argument('--same', action="store_true",
   help='If used, the smiley emoticon are printed on the same line instead of different lines')

# Finalization of argparse
arg = parser.parse_args()

if arg.same:
    a = ':) '
    b = a * arg.num
else:
    a = ':) \n'
    b = a * arg.num
print(b)
```

```
vhaghani@DESKTOP$ python3 smiley_argparse.py
:)
:)
:)
:)
:)

vhaghani@DESKTOP$ python3 smiley_argparse.py --same
:) :) :) :) :)

vhaghani@DESKTOP$ python3 smiley_argparse.py --num 3
:)
:)
:)

vhaghani@DESKTOP$ python3 smiley_argparse.py --num 3 --same
:) :) :)
```

Template

```
#!/usr/bin/env python3

import argparse

#####################
## Set Up Argparse ##
#####################

# Initialize argparse
parser = argparse.ArgumentParser(
    description='Description of program')

# Required argument(s)
parser.add_argument('--required_argument', required=True, type=str,
    metavar='<str>', help='Description of the argument')

# Optional argument(s)
parser.add_argument('--optional_argument', required=False, type=int, default = 5,
    metavar='<int>', help='Some optional argument that uses the default value [%(default)i]')

# Switch(es)
parser.add_argument('--switch1', action="store_true",
   help='An argparse switch that stores the value "True"')

# Finalization of argparse
arg = parser.parse_args()
```


KEEP IN MIND THAT I'M SELF-TAUGHT, SO MY CODE MAY BE A LITTLE MESSY.
LEMME SEE- I'M SURE IT'S FINE.
...WOW.
THIS IS LIKE BEING IN A HOUSE BUILT BY A CHILD USING NOTHING BUT A HATCHET AND A PICTURE OF A HOUSE.
IT'S LIKE A SALAD RECIPE WRITTEN BY A CORPORATE LAWYER USING A PHONE AUTOCORRECT THAT ONLY KNEW EXCEL FORMULAS.
IT'S LIKE SOMEONE TOOK A TRANSCRIPT OF A COUPLE ARGUING AT IKEA AND MADE RANDOM EDITS UNTIL IT COMPILED WITHOUT ERRORS.
OKAY, I'LL READ A STYLE GUIDE.