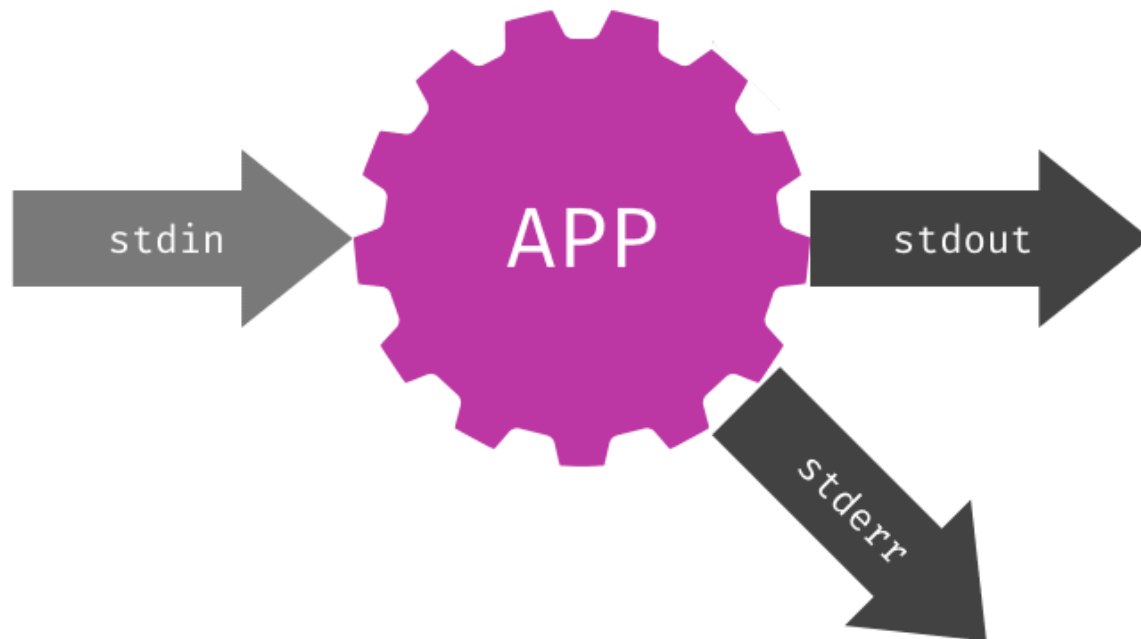


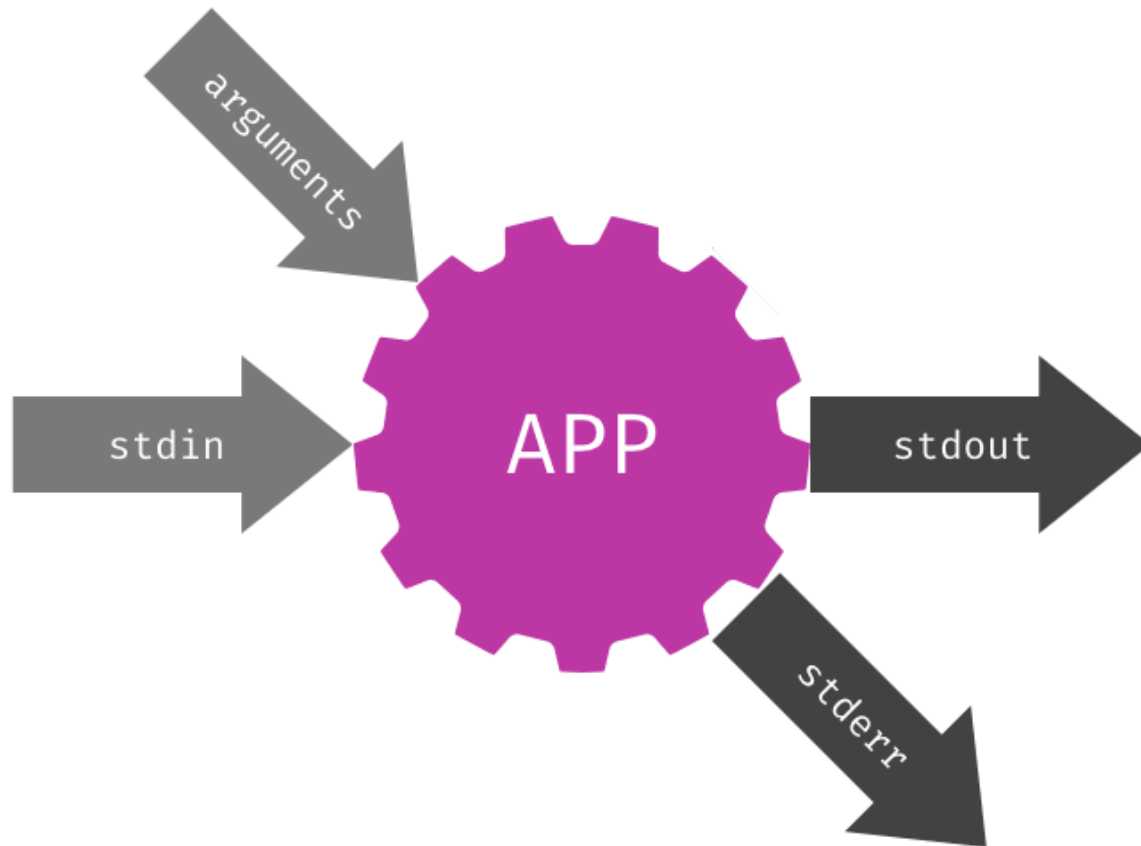
MASTERING APPLICATION/SERVICE CONFIGURATION

Moisés Guimarães - Red Hat









ARGPARSE

```
import argparse

parser = argparse.ArgumentParser(description='Process some integers.')

parser.add_argument('integers', metavar='N', 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()
print(args.accumulate(args.integers))
```



```
$ python prog.py -h
usage: prog.py [-h] [--sum] N [N ...]
```

Process some integers.

positional arguments:

N an integer for the accumulator

optional arguments:

-h, --help show this help message and exit

--sum sum the integers (default: find the max)


```
$ python prog.py 1 2 3 4
```

```
4
```

```
$ python prog.py 1 2 3 4 --sum
```

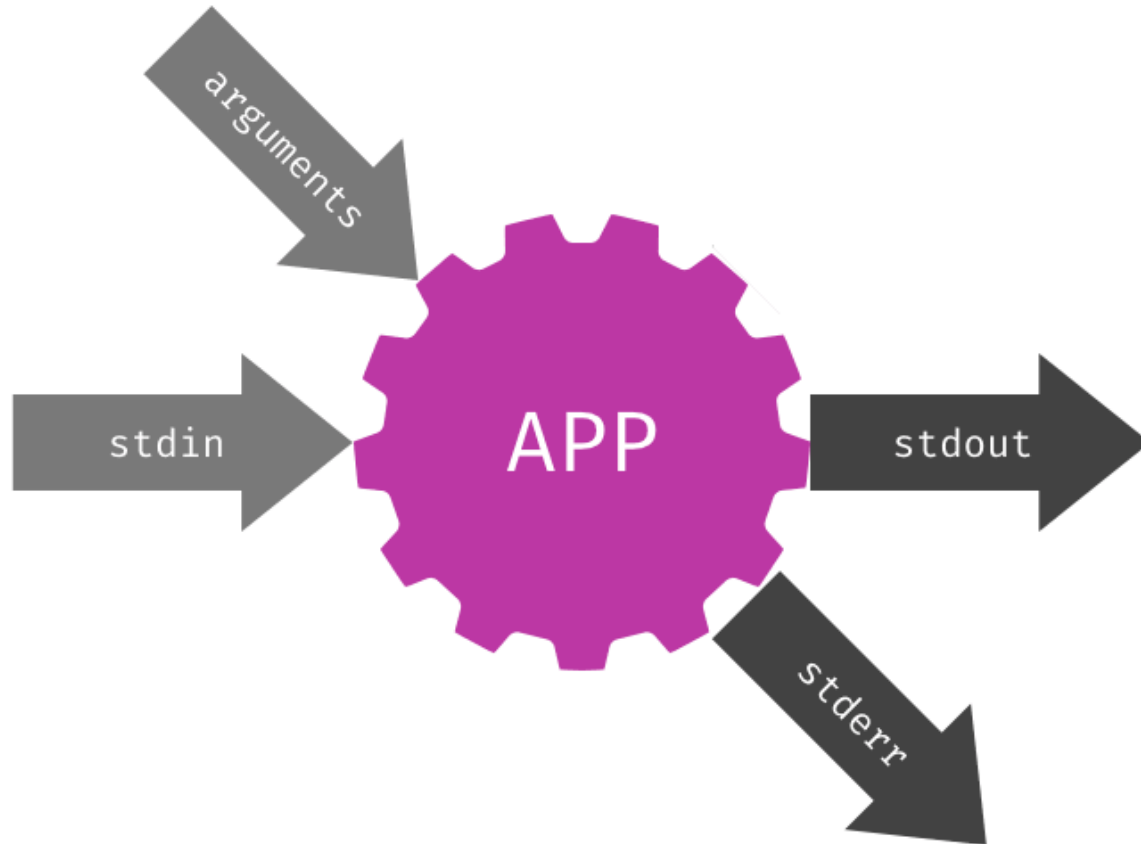
```
10
```

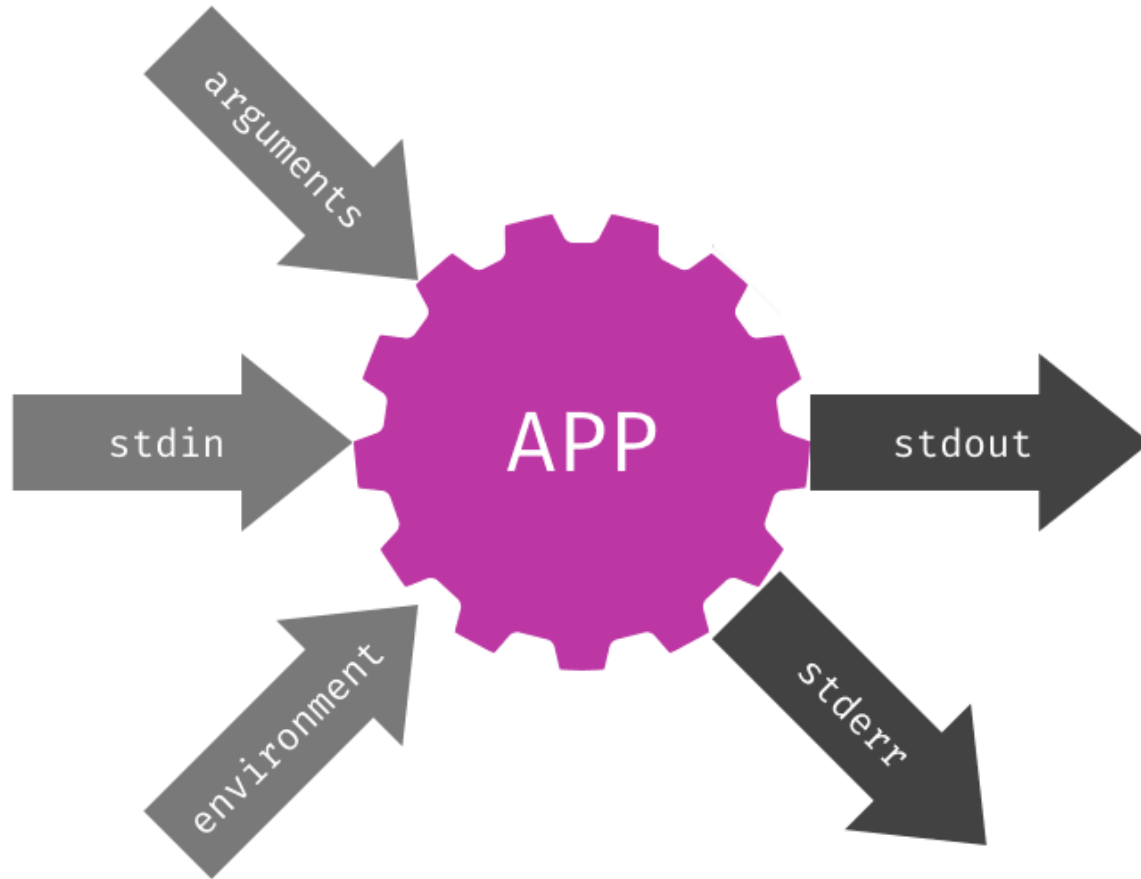
```
$ python prog.py a b c
```

```
usage: prog.py [-h] [--sum] N [N ...]
```

```
prog.py: error: argument N: invalid int value: 'a'
```

```
$
```





OS.ENVIRON


```
>>> import os
>>> os.environ["HOME"]
```

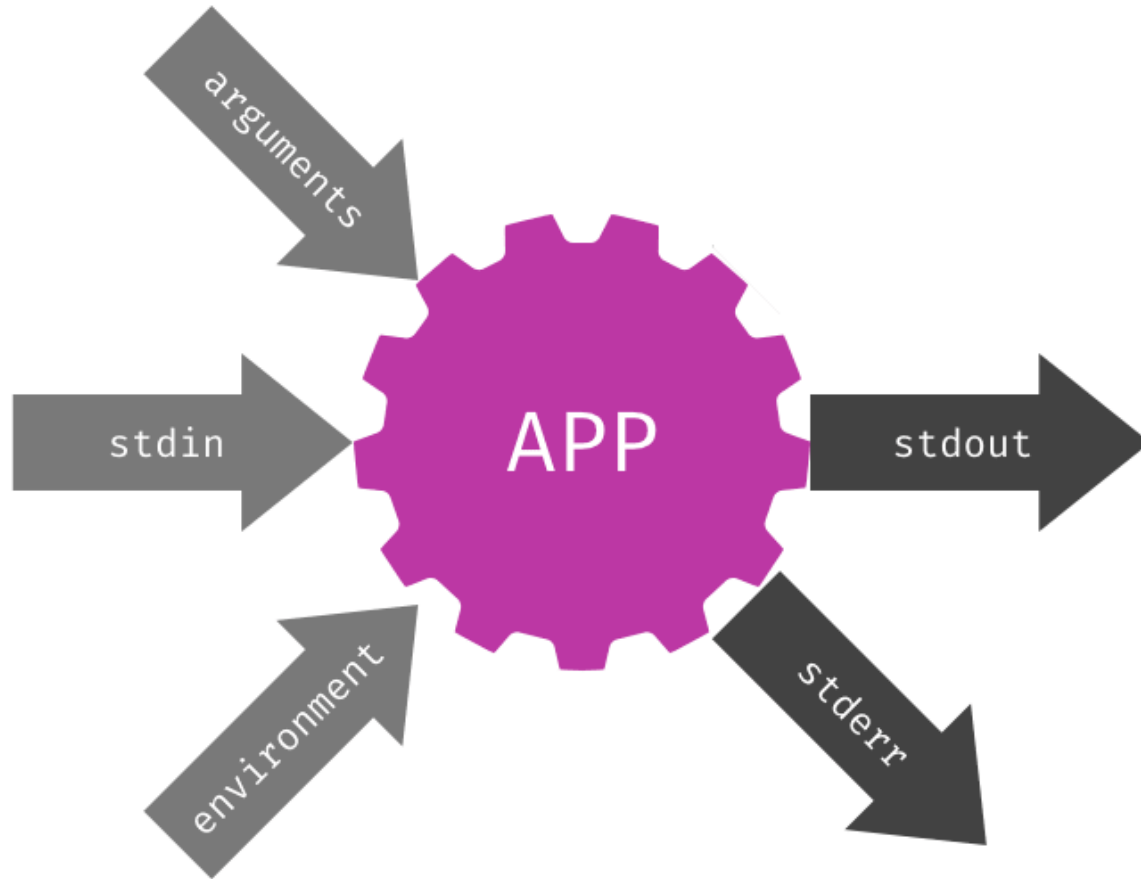


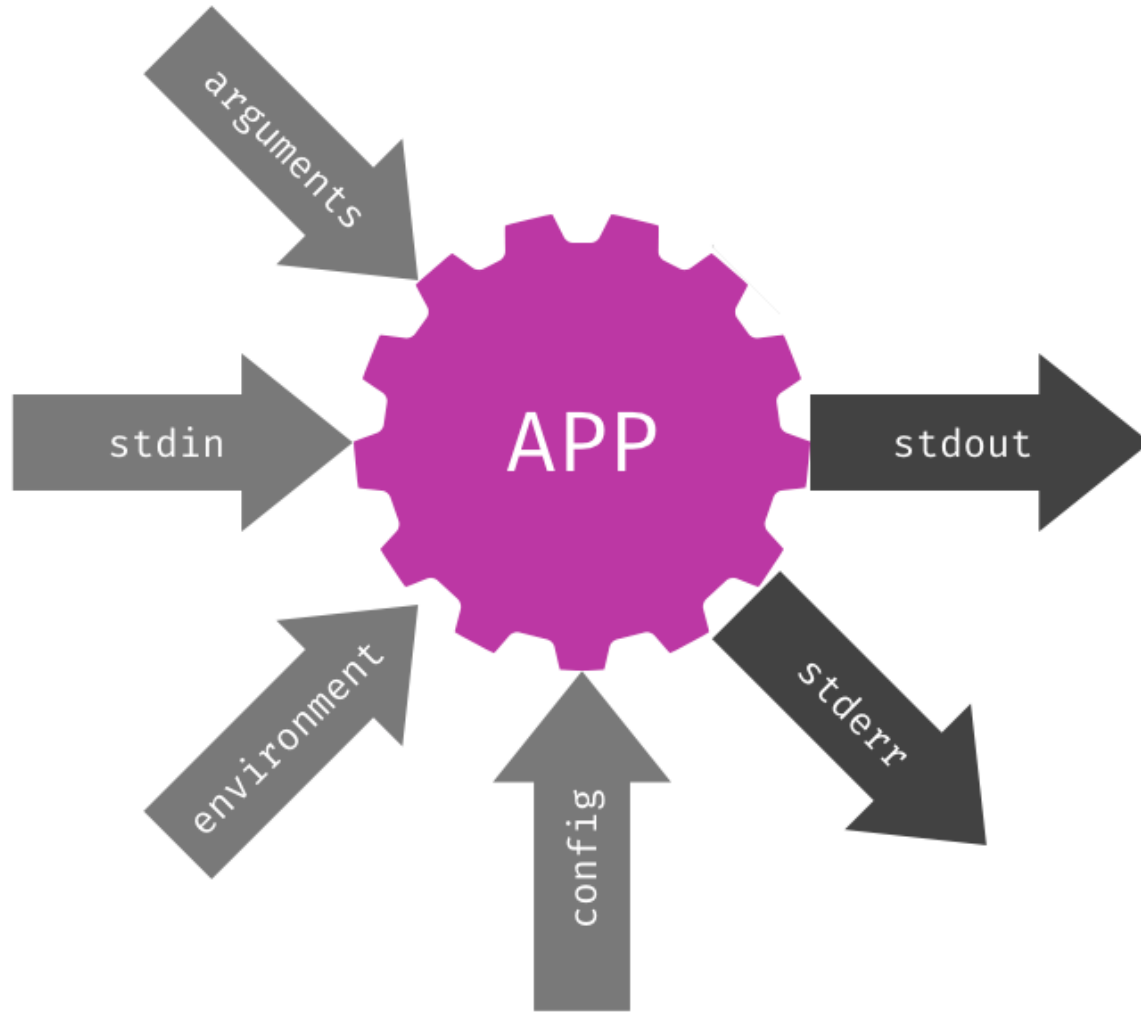
```
>>> import os
>>> os.environ["HOME"]
'/Users/moisesguimaraes'
>>>
```

```
>>> import os
>>> os.environ["HOME"]
'/Users/moisesguimaraes'
>>> os.environ["HOMEBREW_GITHUB_API_TOKEN"]
```

```
>>> import os
>>> os.environ["HOME"]
'/Users/moisesguimaraes'
>>> os.environ["HOMEBREW_GITHUB_API_TOKEN"]

>>>
```





CONFIGPARSER

[DEFAULT]

Compression = yes

CompressionLevel = 9

ForwardX11 = yes

[bitbucket.org]

User = hg

[topsecret.server.com]

Port = 50022

ForwardX11 = no


```
>>> import configparser
>>> config = configparser.ConfigParser()
```

```
>>> import configparser
>>> config = configparser.ConfigParser()
>>>
```

```
>>> import configparser
>>> config = configparser.ConfigParser()
>>> config.read('example.ini')
```

```
>>> import configparser
>>> config = configparser.ConfigParser()
>>> config.read('example.ini')
>>>
```

```
>>> import configparser
>>> config = configparser.ConfigParser()
>>> config.read('example.ini')
>>> config.sections()
```

```
>>> import configparser
>>> config = configparser.ConfigParser()
>>> config.read('example.ini')
>>> config.sections()
['bitbucket.org', 'topsecret.server.com']
>>>
```



```
>>> import configparser
>>> config = configparser.ConfigParser()
>>> config.read('example.ini')
>>> config.sections()
['bitbucket.org', 'topsecret.server.com']
>>> config['bitbucket.org']['User']
```

```
>>> import configparser
>>> config = configparser.ConfigParser()
>>> config.read('example.ini')
>>> config.sections()
['bitbucket.org', 'topsecret.server.com']
>>> config['bitbucket.org']['User']
'hg'
>>>
```

```
>>> import configparser
>>> config = configparser.ConfigParser()
>>> config.read('example.ini')
>>> config.sections()
['bitbucket.org', 'topsecret.server.com']
>>> config['bitbucket.org']['User']
'hg'
>>> config['DEFAULT']['Compression']
```

```
>>> import configparser
>>> config = configparser.ConfigParser()
>>> config.read('example.ini')
>>> config.sections()
['bitbucket.org', 'topsecret.server.com']
>>> config['bitbucket.org']['User']
'hg'
>>> config['DEFAULT']['Compression']
'yes'
>>>
```

```
>>> import configparser
>>> config = configparser.ConfigParser()
>>> config.read('example.ini')
>>> config.sections()
['bitbucket.org', 'topsecret.server.com']
>>> config['bitbucket.org']['User']
'hg'
>>> config['DEFAULT']['Compression']
'yes'
>>> for key in config['bitbucket.org']:
```

```
>>> import configparser
>>> config = configparser.ConfigParser()
>>> config.read('example.ini')
>>> config.sections()
['bitbucket.org', 'topsecret.server.com']
>>> config['bitbucket.org']['User']
'hg'
>>> config['DEFAULT']['Compression']
'yes'
>>> for key in config['bitbucket.org']:
...     print(key)
...
```

```
>>> import configparser
>>> config = configparser.ConfigParser()
>>> config.read('example.ini')
>>> config.sections()
['bitbucket.org', 'topsecret.server.com']
>>> config['bitbucket.org']['User']
'hg'
>>> config['DEFAULT']['Compression']
'yes'
>>> for key in config['bitbucket.org']:
...     print(key)
...
user
compressionlevel
compression
forwardx11
>>>
```

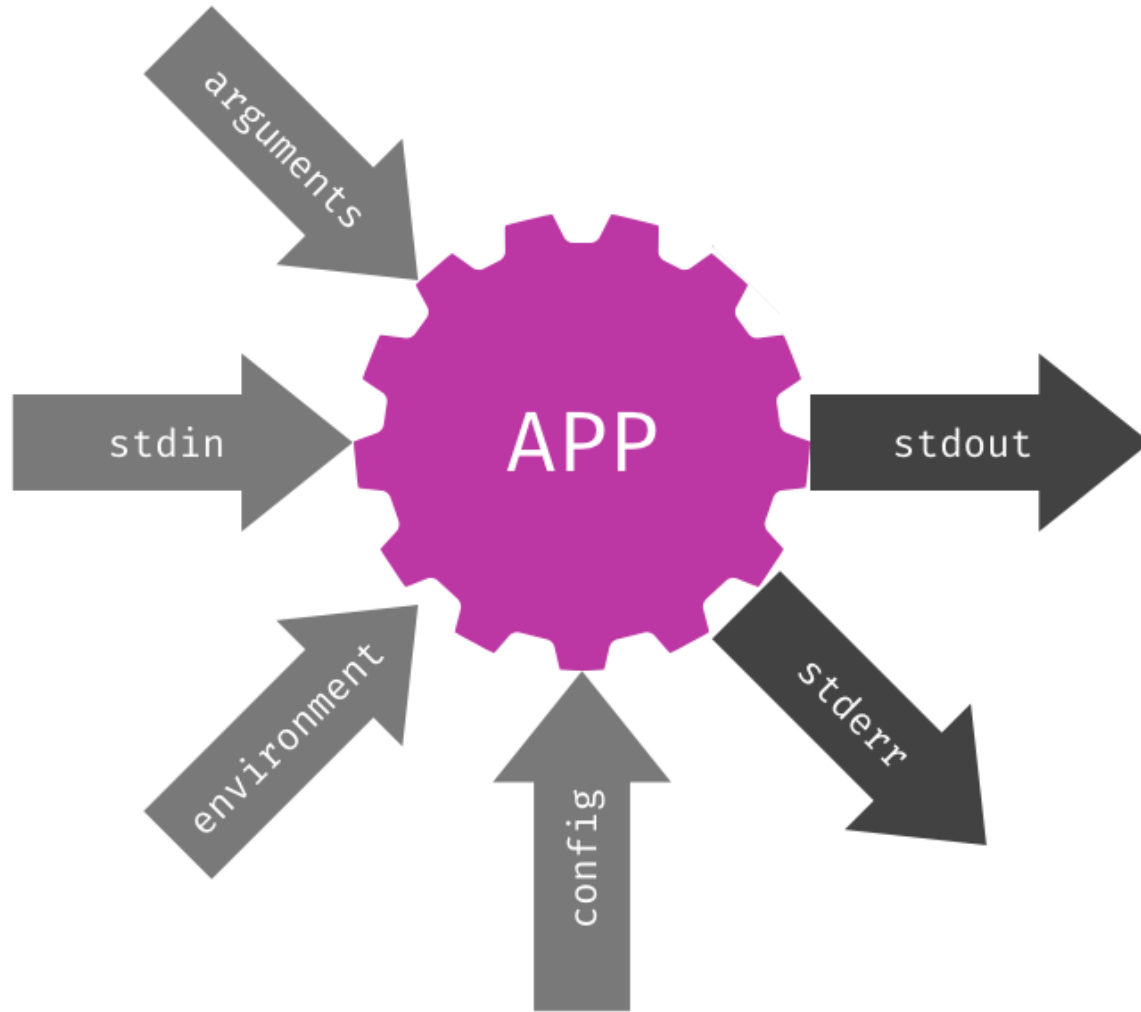
```
>>> import configparser
>>> config = configparser.ConfigParser()
>>> config.read('example.ini')
>>> config.sections()
['bitbucket.org', 'topsecret.server.com']
>>> config['bitbucket.org']['User']
'hg'
>>> config['DEFAULT']['Compression']
'yes'
>>> for key in config['bitbucket.org']:
...     print(key)
...
user
compressionlevel
compression
forwardx11
>>> config['bitbucket.org']['ForwardX11']
```

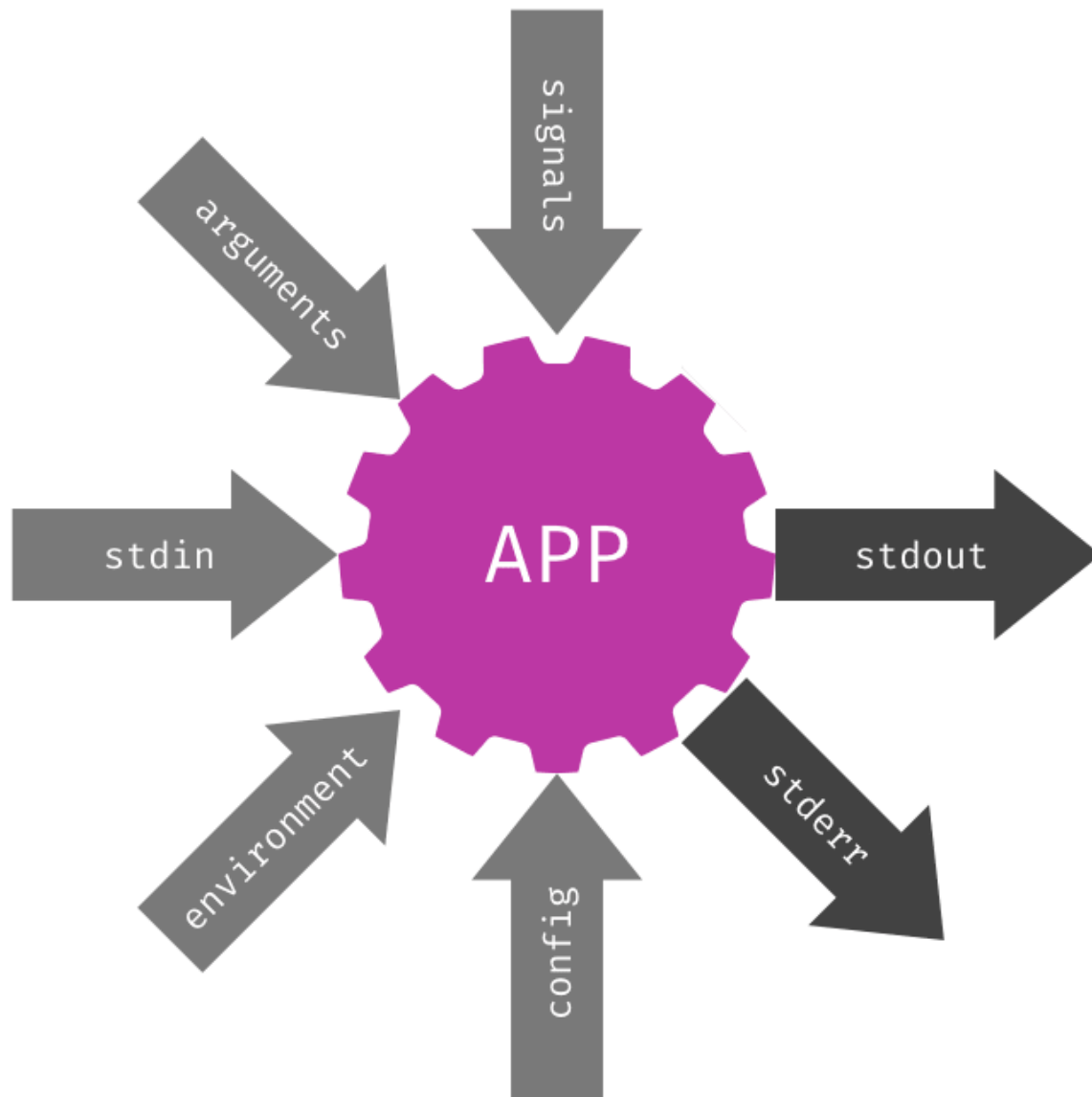


```
>>> import configparser
>>> config = configparser.ConfigParser()
>>> config.read('example.ini')
>>> config.sections()
['bitbucket.org', 'topsecret.server.com']
>>> config['bitbucket.org']['User']
'hg'
>>> config['DEFAULT']['Compression']
'yes'
>>> for key in config['bitbucket.org']:
...     print(key)
...
user
compressionlevel
compression
forwardx11
>>> config['bitbucket.org']['ForwardX11']
'yes'
>>>
```

```
>>> import configparser
>>> config = configparser.ConfigParser()
>>> config.read('example.ini')
>>> config.sections()
['bitbucket.org', 'topsecret.server.com']
>>> config['bitbucket.org']['User']
'hg'
>>> config['DEFAULT']['Compression']
'yes'
>>> for key in config['bitbucket.org']:
...     print(key)
...
user
compressionlevel
compression
forwardx11
>>> config['bitbucket.org']['ForwardX11']
'yes'
>>> topsecret = config['topsecret.server.com']['ForwardX11']
```

```
>>> import configparser
>>> config = configparser.ConfigParser()
>>> config.read('example.ini')
>>> config.sections()
['bitbucket.org', 'topsecret.server.com']
>>> config['bitbucket.org']['User']
'hg'
>>> config['DEFAULT']['Compression']
'yes'
>>> for key in config['bitbucket.org']:
...     print(key)
...
user
compressionlevel
compression
forwardx11
>>> config['bitbucket.org']['ForwardX11']
'yes'
>>> topsecret = config['topsecret.server.com']['ForwardX11']
'no'
>>>
```





UNIX SIGNALS

\$

UNIX SIGNALS

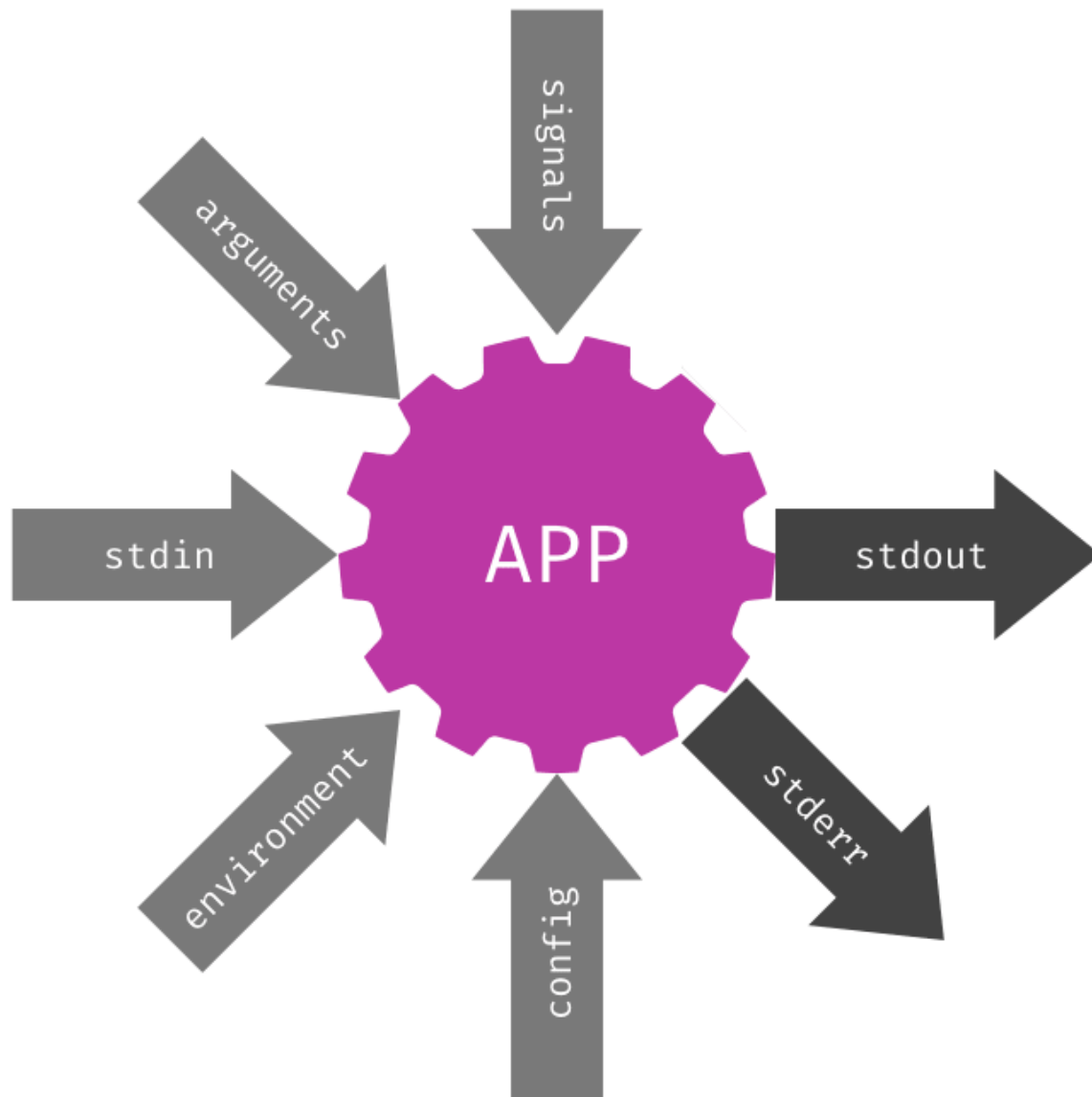
```
$ kill -1
```

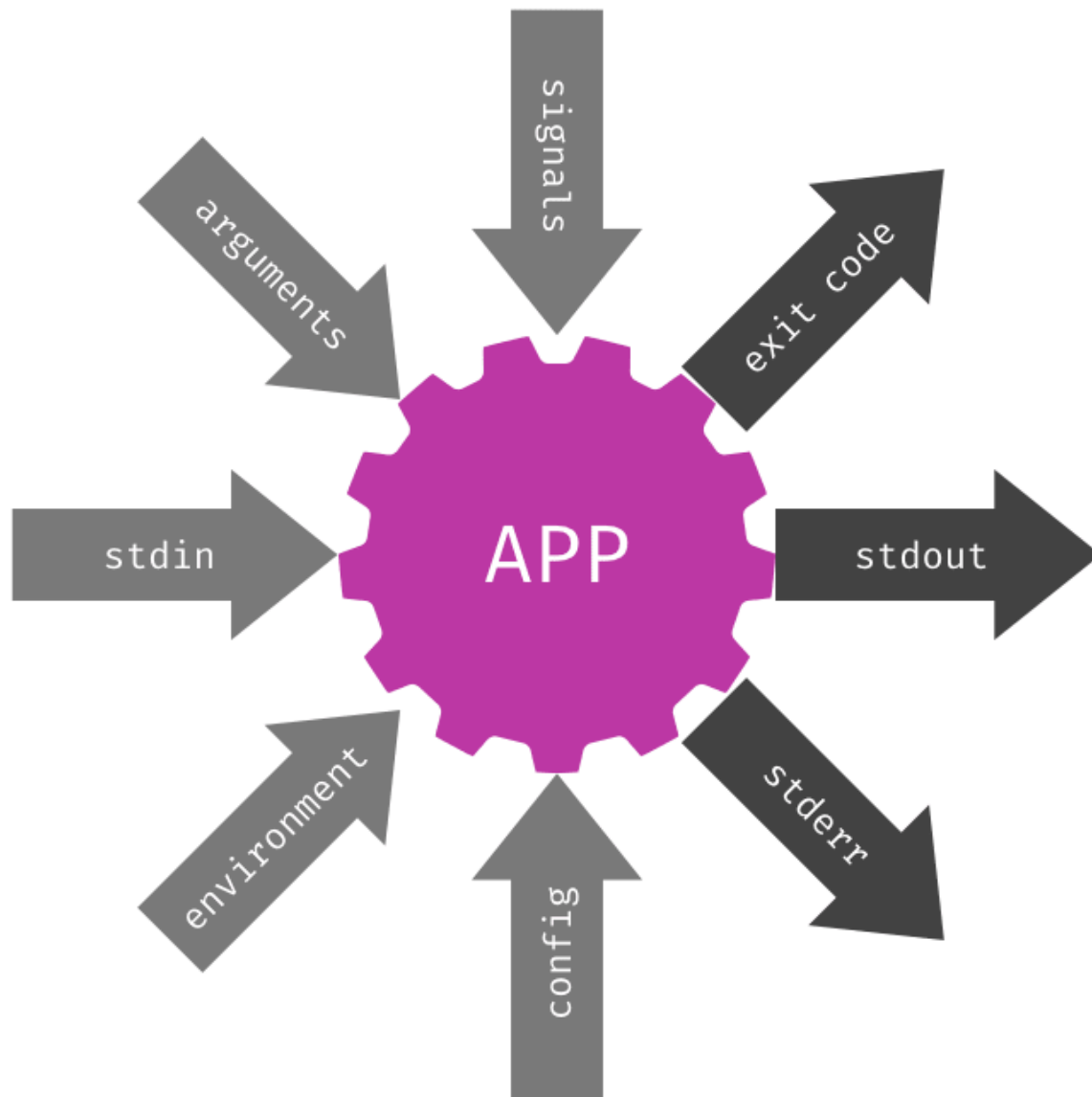
UNIX SIGNALS

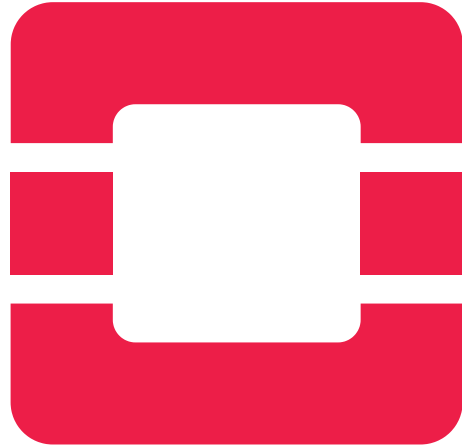
```
$ kill -l
```

1) SIGHUP	2) SIGINT	3) SIGQUIT	4) SIGILL
5) SIGTRAP	6) SIGABRT	7) SIGBUS	8) SIGFPE
9) SIGKILL	10) SIGUSR1	11) SIGSEGV	12) SIGUSR2
13) SIGPIPE	14) SIGALRM	15) SIGTERM	16) SIGSTKFLT
17) SIGCHLD	18) SIGCONT	19) SIGSTOP	20) SIGTSTP
21) SIGTTIN	22) SIGTTOU	23) SIGURG	24) SIGXCPU
25) SIGXFSZ	26) SIGVTALRM	27) SIGPROF	28) SIGWINCH
29) SIGIO	30) SIGPWR	31) SIGSYS	34) SIGRTMIN
35) SIGRTMIN+1	36) SIGRTMIN+2	37) SIGRTMIN+3	38) SIGRTMIN+4
39) SIGRTMIN+5	40) SIGRTMIN+6	41) SIGRTMIN+7	42) SIGRTMIN+8
43) SIGRTMIN+9	44) SIGRTMIN+10	45) SIGRTMIN+11	46) SIGRTMIN+12
47) SIGRTMIN+13	48) SIGRTMIN+14	49) SIGRTMIN+15	50) SIGRTMAX-14
51) SIGRTMAX-13	52) SIGRTMAX-12	53) SIGRTMAX-11	54) SIGRTMAX-10
55) SIGRTMAX-9	56) SIGRTMAX-8	57) SIGRTMAX-7	58) SIGRTMAX-6
59) SIGRTMAX-5	60) SIGRTMAX-4	61) SIGRTMAX-3	62) SIGRTMAX-2
63) SIGRTMAX-1	64) SIGRTMAX		

```
$
```





openstack.®



OSLO.CONFIG

OSLO.CONFIG

CONFIG

OSLO.CONFIG

ARGS +

CONFIG

OSLO.CONFIG

ARGS + ENV + CONFIG

OSLO.CONFIG

ARGS + ENV + CONFIG +

...

OSLO.CONFIG TYPES

String | Boolean | Integer | Float | List | Dict

OSLO.CONFIG TYPES

URI | Hostname | IPAddress | HostAddress | Port

OSLO.CONFIG

```
#!/usr/bin/env python
from oslo_config import cfg

common_opts = [
    cfg.StrOpt('name',
               positional=True,
               default='world',
               help='Name to greet'),
]

greeting_opts = [
    cfg.StrOpt('greeting',
               default='Hello',
               help='Greeting to use.'),

    cfg.IntOpt('times',
               short='n',
               default=1,
               help='Times to greet.'),
]
```

OSLO.CONFIG

```
def main():
    conf = cfg.ConfigOpts()

    conf.register_cli_opts(common_opts)

    conf.register_opt(greeting_opts[0], "greeting")
    conf.register_cli_opt(greeting_opts[1], "greeting")

    conf()

    for i in range(conf.greeting.times):
        print("{} {}!".format(
            conf.greeting.greeting,
            conf.name.capitalize()))

if __name__ == "__main__":
    main()
```



```
$ python hello.py --help
```

```
usage: hello [-h] [--config-dir DIR] [--config-file PATH]
            [--greeting-times GREETING_TIMES]
            [name]
```

positional arguments:

name	Name to greet
------	---------------

optional arguments:

-h, --help	show this help message and exit
--config-dir DIR	Path to a config directory to pull `*.conf` files ...
--config-file PATH	Path to a config file to use. Multiple config files ...

greeting options:

--greeting-times GREETING_TIMES, -n GREETING_TIMES	Times to greet.
--	-----------------


```
$ python hello.py  
Hello World!  
$
```

```
$ python hello.py
```

```
Hello World!
```

```
$ python hello.py Brussels
```

```
$ python hello.py
```

```
Hello World!
```

```
$ python hello.py Brussels
```

```
Hello Brussels!
```

```
$
```

```
$ python hello.py
```

```
Hello World!
```

```
$ python hello.py Brussels
```

```
Hello Brussels!
```

```
$ python hello.py -n 3 Brussels
```

```
$ python hello.py
```

```
Hello World!
```

```
$ python hello.py Brussels
```

```
Hello Brussels!
```

```
$ python hello.py -n 3 Brussels
```

```
Hello Brussels!
```

```
Hello Brussels!
```

```
Hello Brussels!
```

```
$
```

```
$ python hello.py
```

```
Hello World!
```

```
$ python hello.py Brussels
```

```
Hello Brussels!
```

```
$ python hello.py -n 3 Brussels
```

```
Hello Brussels!
```

```
Hello Brussels!
```

```
Hello Brussels!
```

```
$ cat fosdem.conf
```

```
$ python hello.py
Hello World!
$ python hello.py Brussels
Hello Brussels!
$ python hello.py -n 3 Brussels
Hello Brussels!
Hello Brussels!
Hello Brussels!
$ cat fosdem.conf
[DEFAULT]
name = Brussels

[greeting]
greeting = Hallo
times = 3
$
```



```
$ python hello.py
```

```
Hello World!
```

```
$ python hello.py Brussels
```

```
Hello Brussels!
```

```
$ python hello.py -n 3 Brussels
```

```
Hello Brussels!
```

```
Hello Brussels!
```

```
Hello Brussels!
```

```
$ cat fosdem.conf
```

```
[DEFAULT]
```

```
name = Brussels
```

```
[greeting]
```

```
greeting = Hallo
```

```
times = 3
```

```
$ python hello.py --config-file fosdem.conf
```

```
$ python hello.py
```

```
Hello World!
```

```
$ python hello.py Brussels
```

```
Hello Brussels!
```

```
$ python hello.py -n 3 Brussels
```

```
Hello Brussels!
```

```
Hello Brussels!
```

```
Hello Brussels!
```

```
$ cat fosdem.conf
```

```
[DEFAULT]
```

```
name = Brussels
```

```
[greeting]
```

```
greeting = Hallo
```

```
times = 3
```

```
$ python hello.py --config-file fosdem.conf
```

```
Hallo Brussels!
```

```
Hallo Brussels!
```

```
Hallo Brussels!
```

```
$
```



```
$ python hello.py --config-file fosdem.conf -n 30 "FOSDEM'19!"
```

```
Hallo FOSDEM'19!
```

```
Hallo FOSDEM'19!
```

```
Hallo FOSDEM'19!
```

```
Hallo FOSDEM'19!
```

```
Hallo FOSDEM'19!
```

```
Hallo FOSDEM'19!
```

```
Hallo FOSDEM'19!
```

```
Hallo FOSDEM'19!
```

```
Hallo FOSDEM'19!
```

```
Hallo FOSDEM'19!
```

```
Hallo FOSDEM'19!
```

```
Hallo FOSDEM'19!
```

```
Hallo FOSDEM'19!
```

```
Hallo FOSDEM'19!
```

```
Hallo FOSDEM'19!
```

```
Hallo FOSDEM'19!
```

```
Hallo FOSDEM'19!
```

```
Hallo FOSDEM'19!
```

```
Hallo FOSDEM'19!
```

```
Hallo FOSDEM'19!
```

```
$
```



THANK YOU!

GUIMARAES+TALKS@PM.ME

[HTTPS://MOISESGUIMARAES.COM/TALKS](https://moisesguimaraes.com/talks)