Automation with Python

Isaiah Hull

Sveriges Riksbank

November 5, 2017

References

1. “Automate the Boring Stuff with Python”
    - Al Sweigart (2015)

Overview

1. File operations
2. Debugging
3. Timing, Scheduling, and Launching
4. Automating Emails and Text Messages
5. Controlling Keyboard and Mouse Movements
6. Additional Examples

File operations

Files and File Paths

- Three file properties:
  - a filename → name
  - a file path → location on computer
  - an extension → type
- Path conventions differ across operating systems:
  - Forward slashes (/) used on OS X and Linux
  - Backslashes (\) used on Windows

File operations

Platform-Neutrality

```
#lecture2_example1.py

$user python
>>> import os
>>> os.path.join(`usr′,`bin′,`python′)

#windows → usr\\bin\\python
#linux / os x → usr/bin/python
```

File operations

Platform-Neutrality

```
#lecture2_example1.py (continued)

>>> import platform
>>> platform.system()
Darwin
>>> platform.relase()
16.7.0
>>> if platform.system() == `Darwin':
        save_dir = os.path.
        join(`/',`Users',`user',`projects')
    if platform.system() == `Windows':
        save_dir = os.path.
        join(`C',`Documents',`Project')
```

File operations

Absolute and Relative Paths

```
#lecture2_example2.py

$user python
>>> import os
>>> os.getcwd()
`/Users/user’
>>> os.chdir(`/Users/user/Desktop’)
>>> os.getcwd()
`/Users/user/Desktop’
>>> os.makedirs(`/Users/user/Desktop/Projects’)
```

File operations

Absolute and Relative Paths

```
#lecture2_example2.py (continued)

>>> os.path.relpath(`/Users/user/Desktop/file.csv',
`/Users/user')
Desktop
>>> os.path.split(`/Users/user/Desktop/file.csv')
(`/Users/user/Desktop', `file.csv')
```

File operations

Absolute and Relative Paths

```
#lecture2_example2.py (continued)

>>> os.path.dirname(`/Users/user/
Desktop/file.csv’)
`/Users/user/Desktop’
>>> os.path.basename(`/Users/user/
Desktop/file.csv’)
`file.csv’
```

File operations

Additional Commands

- `os.path.getsize()` → get file size
- `os.listdir()` → list contents of directory
- `os.path.exists()` → tests if path exists
- `os.path.isdir()` → test if location is a directory
- `os.path.isfile()` → test if location is a file

File operations

Copying files and folders

```
$user python
>>> import shutil
>>> import os
```

File operations

Copying files and folders

- `shutil` → (sh)ell (util)ities
- `shutil.copy(s, d)` → copy file at location s to d
- `shutil.copytree(s, d)` → copy folder and contents from location s to d
- `shutil.move(s, d)` → move file from location s to d
- `shutil.rmtree(s)` → removes folders at s
- `os.unlink(p)` → deletes files at p
- `os.rmdir(p)` → removes directory at p

File operations

Zip files

```
$user python
>>> import zipfile
>>> os.chdir(...)
>>> zipFile = zipfile.ZipFile(`example.zip’)
>>> zipFile.extractall()
>>> zipFile.close()
```

Debugging

Overview

- Debugging important, but often overlooked
- Useful to know why program failed
- Important to provide clear exception messages for collaborative projects

Debugging

Exceptions

```
$user python
>>> def invertNumber(x):
           if x == 0:
                raise Exception(`Division by zero.')
           return 1.0/x
>>> invertNumber(2)
0.5
>>> invertNumber(0)
Exception: Division by zero.
```

Debugging

Assertions

```
$user python
>>> X = None
>>> ...
>>> assert X!=None, ’X cannot be None.’
AssertionError:  X cannot be None.
```

Debugging

Logging

- `import logging`
- `logging.debug()` → low level logging
- `logging.info()` → higher level logging
- `logging.warning()` → potential problems
- `logging.error()` → program failures
- `logging.critical()` → highest level logging

Timing, Scheduling, and Launching

Overview

- Many tasks are best performed on a schedule.
- Scraping may need to be done daily, weekly, or monthly.
- Data may be updated at a certain time or needed at a certain time.
- Scheduling can be done with Python's `time` and `datetime` modules.

Timing, Scheduling, and Launching

Time Keeping

```
$user python
>>> import time
>>> time.time()
>>> time.sleep(3)
```

Timing, Scheduling, and Launching

Time Keeping

```
#lecture2_example4.py

$user python
>>> import time
>>> def calcSum(X):
           s = 0
           for j in range(1,X+1):
                s += j
           return s
>>> startTime = time.time()
>>> output = calcSum(10000000)
>>> endTime = time.time()
>>> print endTime-startTime
0.691835165024
```

Timing, Scheduling, and Launching

Scheduling

```
#lecture2_example5.py

$user python
>>> import time
>>> import datetime
>>> delay = 60
>>> year = 2017
>>> month = 11
>>> day = 4
>>> hour = 10
>>> minute = 0
>>> second = 0
```

Timing, Scheduling, and Launching

Scheduling

```
#lecture2_example5.py (continued)
>>> runDate = datetime.datetime(year, month,
day, hour, minute, second)
>>> runDate.year
2017
>>> runDate.hour
10
>>> def doSomething():
        return
>>> while datetime.datetime.now() < runDate:
        time.sleep(delay)
>>> doSomething()
```

Timing, Scheduling, and Launching

Multithreaded Scheduling

- What if we don't want the program to remain idle until the execution date?
  - Use multithreading
  - One thread executes scheduling task
  - Other threads perform different tasks until execution date

Timing, Scheduling, and Launching

Multithreaded Scheduling

```
#lecture2_example6.py

$user python
>>> import threading
>>> import time
>>> def doSomething():
          for j in range(100):
               time.sleep(1)
               print(`Something.')
>>> def doSomethingElse():
          for j in range(100):
               time.sleep(1)
               print(`Something else.')
```

Timing, Scheduling, and Launching

Multithreaded Scheduling

```
#lecture2_example6.py (continued)
>>> thread1 = threading.
Thread(target=doSomething()).start()
>>> thread2 = threading.
Thread(target=doSomethingElse()).start()
```

Timing, Scheduling, and Launching

Multithreaded Scheduling

```
Something.
Something else.
Something.
Something else.
Something.
Something else.
Something.
Something else.
Something.
```

Timing, Scheduling, and Launching

Launching Applications

- Many automation tasks require programmatic interaction with an application's GUI
- If a task is scheduled, the application may not be open when it is executed
- Python offers libraries for operating system interaction
  - os
  - subprocess

Timing, Scheduling, and Launching

Launching Applications

```
$user python
>>> import os
>>> import subprocess
```

Timing, Scheduling, and Launching

Launching Applications

- `os.system(“open /Applications/R.app/”)` → launch the R application
- `subprocess.Popen(“open /Applications/R.app/”, shell=True)` → launch the R application

Sending Email and Text Messages

Overview

- Automated email
  - Highly-customizable autoresponder for projects
  - Transmit logs from remote server
  - Updates on request
- SMS messages
  - Program updates and completion
  - Error messages

Sending Email and Text Messages

Sending Automated Emails

```
#lecture2_example7.py

$user python
>>> import smtplib
>>> smtpObj = smtplib.SMTP(`smtp.gmail.com', 587)
>>> smtpObj.ehlo()
>>> smtpObj.starttls()
>>> smtpObj.login(`user@gmail.com', `password')
>>> smtpObj.sendmail(`user@gmail.com',
`other.user@gmail.com', `Subject:  Test.  Dear Other
User, This is a a test.  Sincerely, User')
>>> smtpObj.quit()
```

Sending Email and Text Messages

Sending Automated Emails

- smtplib.SMTP() → create SMTP object
- smtpObj.ehlo() → establish connection to email server
- smtpObj.starttls() → enable encrypted connection

Sending Email and Text Messages

Automated Email Retrieval

```
#lecture2_example8.py

$user pip install imapclient
$user python
>>> import imapclient
>>> imapObj = imapclient.\
IMAPClient(`imap.gmail.com', ssl=True)
>>> imapObj.login(`user@gmail.com', `password')
>>> imapObj.select_folder(`INBOX')
>>> messages = imapObj.search([`NOT',`DELETED'])
```

Sending Email and Text Messages

Automated Email Documentation

- SMTP
  - https://docs.python.org/2/library/smtplib.html
- IMAP
  - https://imapclient.readthedocs.io/en/stable/

Sending Email and Text Messages

Sending Text Messages

```
#lecture2_example9.py

$user pip install twilio
$user python
>>> from twilio.rest import Client
>>> account_sid = “ACXXXXXXXXXXXXXXXXX”
>>> auth_token = “YYYYYYYYYYYYYYYYYY”
>>> to_number = “+XXXXXXXXXXXX”
>>> twilio_number = “+YYYYYYYYYYYY”
>>> body_message = “No errors.”
>>> client = Client(account_sid, auth_token)
>>> client.api.account.messages.create(
          to = to_number,
          from_ = twilio_number,
          body = body_message)
```

Sending Email and Text Messages

Sending Text Messages

```
#lecture2_example9.py (continued)

>>> def generateMessage(errorStatus):
        if error_status == 0:
            body_message = “No errors.”
        else:
            body_message = “Program failed.”
        return body_message
```

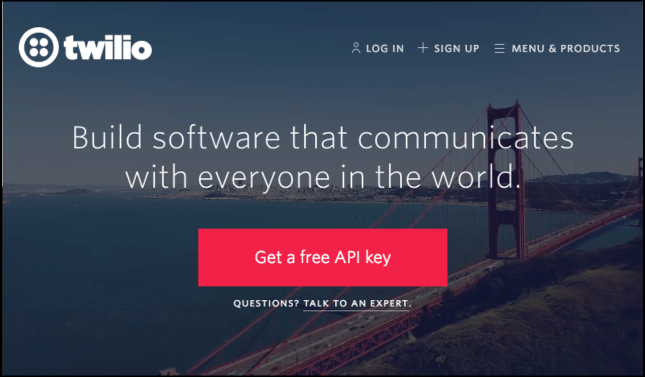
Sending Email and Text Messages

Sending Text Messages

```
#lecture2_example9.py (continued)

>>> def generateTarget(message_target):
        if message_target == 0:
             to_number = “+XXXXXXXXXXXX”
        else:
             to_number = “+ZZZZZZZZZZZZ”
        return to_number
```

Sending Email and Text Messages

Sending Text Messages


Build software that communicates with everyone in the world.

https://www.twilio.com/

Sending Email and Text Messages

Text Message Alternatives

- Pushbullet
  - https://github.com/randomchars/pushbullet.py
- WhatsApp
  - https://github.com/mukulhase/WhatsAPI

Controlling Keyboard and Mouse Movements

Overview

- Many applications do not have an API
- May need to collect data from or manipulate application
- Can program `pyautogui` to perform any task
  - Keyboard movements
  - Mouse clicks

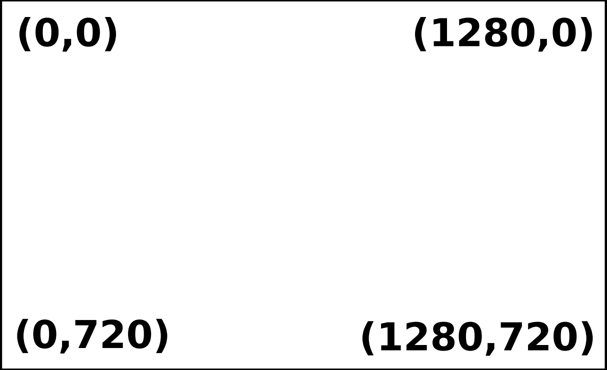
Controlling Keyboard and Mouse Movements

Introduction

```
#lecture2_example10.py

$user pip install pyautogui
$user python
>>> import pyautogui
>>> pyautogui.position()
(759, 429)
>>> pyautogui.size()
(1920, 1200)
>>> pyautogui.onScreen(3000, 3000)
False
>>> pyautogui.onScreen(500, 500)
True
```

Controlling Keyboard and Mouse Movements

Coordinate System



Controlling Keyboard and Mouse Movements

Delays and Failsafes

- `pyautogui.PAUSE = 3` → add delay between all calls
- `time.sleep(3)` → add delay before next call
- `pyautogui.FAILSAFE = True` → moving mouse to upper-left corner to generate exception

Controlling Keyboard and Mouse Movements

Mouse Movements

```
$user python
>>> import pyautogui
>>> pyautogui.moveTo(100, 100, duration=0.25)
>>> pyautogui.moveRel(100, 0, duration=0.25))
>>> pyautogui.click()
>>> pyautogui.click(500, 500)
>>> pyautogui.dragRel(100, 0, duration=0.2)
>>> pyautogui.scroll(100)
```

Controlling Keyboard and Mouse Movements

Additional Commands

- `im = pyautogui.screenshot()` → take screenshot
- `im.getpixel(x,y)` → get pixel color as RGB tuple
- `pyautogui.pixelMatchesColor(x, y, (r, g, b))` → check if pixel matches color
- `pyautogui.locateOnScreen(`button.png')` → finds location of image on screen

Controlling Keyboard and Mouse Movements

Additional Commands

- `pyautogui.locateAllOnScreen(`button.png')` → finds location of all identical images on screen
- `pyautogui.center((x, y, l, w))` → find center of image at coordinates (x, y) with dimensions (l, w)

Controlling Keyboard and Mouse Movements

Keyboard Controls

- `pyautogui.typewrite(`Hello world!′)` → Type “Hello world!” into active window
- `pyautogui.typewrite(`Hello world!′, 0.25)` → Add 0.25 second delay between key presses
- `pyautogui.keyDown(`shift′); pyautogui.press(`7′)` → &

Controlling Keyboard and Mouse Movements

Keyboard Controls

- pyautogui.hotkey(`ctrl′, ′c′) → hotkey for copying
- pyautogui.keyDown(`shift′) → press ‘shift’ key down
- pyautogui.keyUp(`shift′) → release ‘shift’ key

Controlling Keyboard and Mouse Movements

Additional Resources

- Documentation
  - https://pyautogui.readthedocs.io/en/latest/
- Cheat Sheet
  - https://pyautogui.readthedocs.io/en/latest/cheatsheet.html

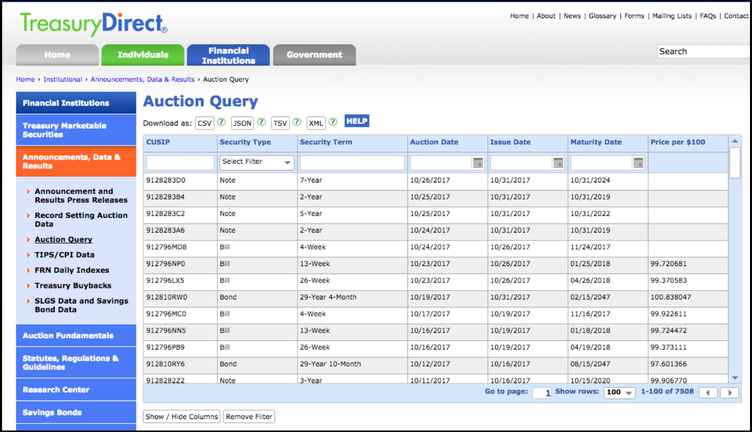
Additional Examples

Example: Treasury Auction Data

1. Load Treasury Direct Auctions.
2. Select today's auctions.
3. Save data with today's date as file name.
4. Send SMS message if successful.

Additional Examples

Example: Treasury Auction Data


TreasuryDirect®
Home | About | News | Glossary | Forms | Mailing Lists | FAQs | Contact
Home
Individuals
Financial Institutions
Government
Search
Home › Institutional › Announcements, Data & Results › Auction Query
Financial Institutions
Treasury Marketable Securities
Announcements, Data & Results
Announcement and Results Press Releases
Record Setting Auction Data
Auction Query
TIPS/CPI Data
FRN Daily Indexes
Treasury Buybacks
SLGS Data and Savings Bond Data
Auction Fundamentals
Statutes, Regulations & Guidelines
Research Center
Savings Bonds

Auction Query

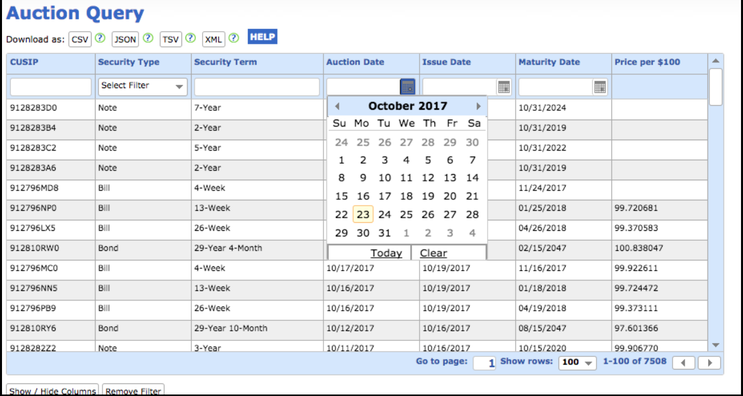
Download as: CSV ? JSON ? TSV ? XML ? HELP

CUSIP	Security Type	Security Term	Auction Date	Issue Date	Maturity Date	Price per $100
	Select Filter					
9128283D0	Note	7-Year	10/26/2017	10/31/2017	10/31/2024	
9128283B4	Note	2-Year	10/25/2017	10/31/2017	10/31/2019	
9128283C2	Note	5-Year	10/25/2017	10/31/2017	10/31/2022	
9128283A6	Note	2-Year	10/24/2017	10/31/2017	10/31/2019	
912796MD8	Bill	4-Week	10/24/2017	10/26/2017	11/24/2017	
912796NP0	Bill	13-Week	10/23/2017	10/26/2017	01/25/2018	99.720681
912796LX5	Bill	26-Week	10/23/2017	10/26/2017	04/26/2018	99.370583
912810RW0	Bond	29-Year 4-Month	10/19/2017	10/31/2017	02/15/2047	100.838047
912796MC0	Bill	4-Week	10/17/2017	10/19/2017	11/16/2017	99.922611
912796NN5	Bill	13-Week	10/16/2017	10/19/2017	01/18/2018	99.724472
912796PB9	Bill	26-Week	10/16/2017	10/19/2017	04/19/2018	99.373111
912810RY6	Bond	29-Year 10-Month	10/12/2017	10/16/2017	08/15/2047	97.601366
9128282Z2	Note	3-Year	10/11/2017	10/16/2017	10/15/2020	99.906770

Go to page: 1 Show rows: 100 1-100 of 7508

Additional Examples

Example: Treasury Auction Data



Auction Query

Download as: CSV ? JSON ? TSV ? XML ? HELP

CUSIP	Security Type	Security Term	Auction Date	Issue Date	Maturity Date	Price per $100
	Select Filter					
9128283D0	Note	7-Year			10/31/2024	
9128283B4	Note	2-Year			10/31/2019	
9128283C2	Note	5-Year			10/31/2022	
9128283A6	Note	2-Year			10/31/2019	
912796MD8	Bill	4-Week			11/24/2017	
912796NP0	Bill	13-Week			01/25/2018	99.720681
912796LX5	Bill	26-Week			04/26/2018	99.370583
912810RW0	Bond	29-Year 4-Month			02/15/2047	100.838047
912796MC0	Bill	4-Week	10/17/2017	10/19/2017	11/16/2017	99.922611
912796NN5	Bill	13-Week	10/16/2017	10/19/2017	01/18/2018	99.724472
912796PB9	Bill	26-Week	10/16/2017	10/19/2017	04/19/2018	99.373111
912810RY6	Bond	29-Year 10-Month	10/12/2017	10/16/2017	08/15/2047	97.601366
9128282Z2	Note	3-Year	10/11/2017	10/16/2017	10/15/2020	99.906770

October 2017

Su	Mo	Tu	We	Th	Fr	Sa
24	25	26	27	28	29	30
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

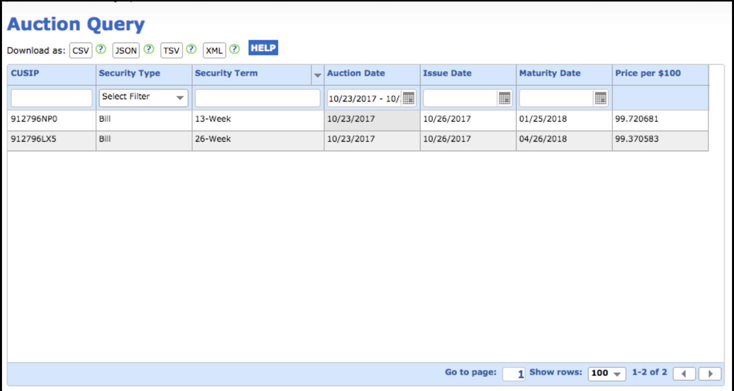
Today Clear

Go to page: 1 Show rows: 100 1-100 of 7508

Show / Hide Columns Remove Filter

Additional Examples

Example: Treasury Auction Data



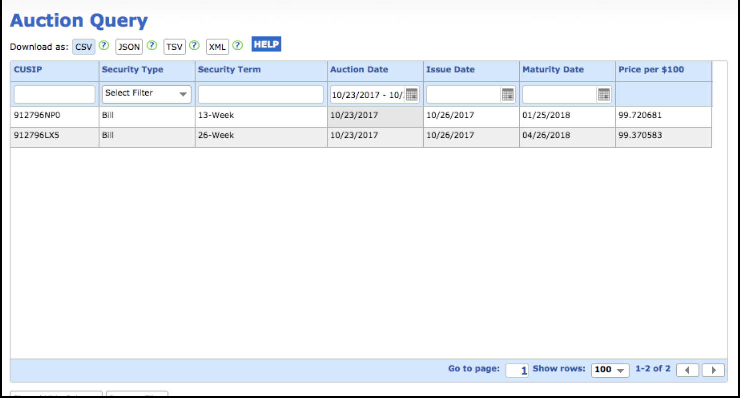
Auction Query

CUSIP	Security Type	Security Term	Auction Date	Issue Date	Maturity Date	Price per $100
	Select Filter		10/23/2017 - 10/			
912796NP0	Bill	13-Week	10/23/2017	10/26/2017	01/25/2018	99.720681
912796LX5	Bill	26-Week	10/23/2017	10/26/2017	04/26/2018	99.370583

Go to page: 1 Show rows: 100 1-2 of 2

Additional Examples

Example: Treasury Auction Data



Auction Query

Download as: CSV ? JSON ? TSV ? XML ? HELP

CUSIP	Security Type	Security Term	Auction Date	Issue Date	Maturity Date	Price per $100
	Select Filter		10/23/2017 - 10/			
912796NP0	Bill	13-Week	10/23/2017	10/26/2017	01/25/2018	99.720681
912796LX5	Bill	26-Week	10/23/2017	10/26/2017	04/26/2018	99.370583

Go to page: 1 Show rows: 100 1-2 of 2

Additional Examples

Example: Treasury Auction Data

```
#lecture2_example11.py

$user python
>>> import time
>>> import pyautogui
>>> import numpy as np
>>> import pandas as pd
>>> from twilio.rest import Client
>>> downloadPath = `../Downloads'
>>> imagePath = `../images/'
>>> url = `https://www.treasurydirect.gov
/instit/annceresult/annceresult_query.htm'
```

Additional Examples

Example: Treasury Auction Data

```
#lecture2_example11.py (continued)

>>> def returnCenter(filename, path=imagePath):
        path = imagePath
        location = pyautogui.\
        locateOnScreen(path+filename)
        center = pyautogui.center(location)
        return center
```

Additional Examples

Example: Treasury Auction Data

```
#lecture2_example11.py (continued)

>>> def sendMessage(bodyMessage):
          account_sid = “ACXXXXXXXXXXXXXXXXX”
          auth_token = “YYYYYYYYYYYYYYYYYY”
          to_number = “+XXXXXXXXXXXX”
          twilio_number = “+YYYYYYYYYYYY”
          body_message = bodyMessage
          client = Client(account_sid, auth_token)
          client.api.account.messages.create(
               to = to_number,
               from_ = twilio_number,
               body = body_message)
```

Additional Examples

Example: Treasury Auction Data

```
#lecture2_example12.py

>>> # Open firefox.

>>> firefoxCenter = returnCenter(`firefox.png')
>>> pyautogui.moveTo(firefoxCenter, duration=0.25)
>>> pyautogui.doubleClick()

# Wait for firefox to load.

>>> time.sleep(2)
```

Additional Examples

Example: Treasury Auction Data

```
#lecture2_example12.py (continued)

# Load Treasury Auction website.

>>> addressCenter = returnCenter(`enter_
address.png')
>>> pyautogui.moveTo(addressCenter, duration=0.25)
>>> pyautogui.click()
>>> pyautogui.typewrite(url, interval=0.05)
>>> pyautogui.press(`enter')

# Wait for page to load.

>>> time.sleep(3)
```

Additional Examples

Example: Treasury Auction Data

```
#lecture2_example12.py (continued)

# Select the auction date.

>>> dateCenter = returnCenter(`auction_date.png')
>>> pyautogui.moveTo(dateCenter, duration=0.25)
>>> pyautogui.moveRel(50, 10)
>>> pyautogui.doubleClick()

# Wait for calendar to load.

>>> time.sleep(2)
```

Additional Examples

Example: Treasury Auction Data

```
#lecture2_example12.py (continued)

# Select today's date.

>>> todayCenter = returnCenter(`today.png')
>>> pyautogui.moveTo(todayCenter, duration=0.25)
>>> pyautogui.click()

# Wait for auction list to load.

>>> time.sleep(2)
```

Additional Examples

Example: Treasury Auction Data

```
#lecture2_example12.py (continued)

# Download data in csv format.

>>> csvCenter = returnCenter(`csv.png′)
>>> pyautogui.moveTo(csvCenter, duration=0.25)
>>> pyautogui.doubleClick()

# Wait for save dialog box.

>>> time.sleep(2)
```

Additional Examples

Example: Treasury Auction Data

```
#lecture2_example12.py (continued)

# Save csv data.

>>> pyautogui.press(`enter')

# Load csv data.

>>> data = pd.read_csv(downloadPath+
`Securities.csv')
```

Additional Examples

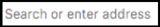
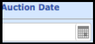
Example: Treasury Auction Data

```
#lecture2_example12.py (continued)

# Compute mean price.

>>> try:
        meanPrice = str(np.mean(data[`Price
        per $100’]))
      except:
        meanPrice = `No auction data available.’

>>> bodyMessage = `Price per $100:  ’+meanPrice+`.’

# Send message.

>>> sendMessage(bodyMessage)
```

Additional Examples

Example: Treasury Auction Data

#lecture2_example12.py (continued)

- ‘firefox.png’ →
- ‘enter_address.png’ → 
Search or enter address
- ‘auction_date.png’ → 
Auction Date
- ‘today.png’ →
Today
- ‘csv.png’ →
CSV

Additional Examples

Example: Writing and Compiling LaTeX Documents

```
#lecture2_example12.py (continued)

$user python
>>> import subprocess
>>> import pyautogui
>>> import time
>>> imagePath = `../images/'
```

Additional Examples

Example: Writing and Compiling LaTeX Documents

```
#lecture2_example12.py (continued)

>>> def returnCenter(filename, path=imagePath):
        path = imagePath
        location = pyautogui.
        locateOnScreen(path+filename)
        center = pyautogui.center(location)
        return center
```

Additional Examples

Example: Writing and Compiling LaTeX Documents

```
#lecture2_example12.py (continued)

>>> subprocess.Popen(`open
/Applications/TeXShop.app', shell=True)
```

Additional Examples

Example: Writing and Compiling LaTeX Documents

```
#lecture2_example12.py (continued)

>>> time.sleep(2)
>>> pyautogui.
typewrite(`\\documentclass[12pt]article’,
interval=0.05)
>>> pyautogui.press(`enter’)
>>> pyautogui.press(`enter’)
>>> pyautogui.typewrite(`\\begin{document}’,
interval=0.05)
>>> pyautogui.press(`enter’)
>>> pyautogui.press(`enter’) >>>
pyautogui.typewrite(`\\title{The Fisher Equation}’,
interval=0.05)
>>> pyautogui.press(`enter’)
```

Additional Examples

Example: Writing and Compiling LaTeX Documents

```
#lecture2_example12.py (continued)
>>> pyautogui.press(`enter’)
>>> pyautogui.typewrite(`\\author{document}’,
interval=0.05)
>>> pyautogui.press(`enter’)
>>> pyautogui.press(`enter’)
>>> pyautogui.typewrite(`\\date{}’, interval=0.05)
>>> pyautogui.press(`enter’)
>>> pyautogui.press(`enter’)
>>> pyautogui.typewrite(`\\maketitle{}’,
interval=0.05)
>>> pyautogui.press(`enter’)
>>> pyautogui.press(`enter’)
```

Additional Examples

Example: Writing and Compiling LaTeX Documents

```
#lecture2_example12.py (continued)

>>> pyautogui.typewrite(`\\begin{equation}',
interval=0.05)
>>> pyautogui.press(`enter')
>>> pyautogui.press(`enter')
>>> pyautogui.typewrite(`i \\approx r + \pi',
interval=0.05)
>>> pyautogui.press(`enter')
>>> pyautogui.press(`enter')
>>> pyautogui.typewrite(`\\end{equation}',
interval=0.05)
>>> pyautogui.press(`enter')
>>> pyautogui.press(`enter')
>>> pyautogui.typewrite(`\\end{document}'
```

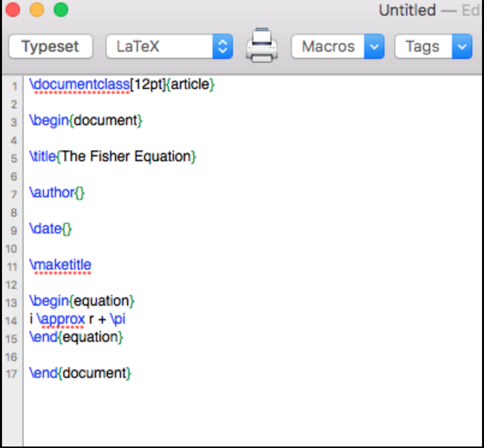
Additional Examples

Example: Writing and Compiling LaTeX Documents

```
#lecture2_example12.py (continued)

>>> typesetCenter = returnCenter(`typeset.png')
>>> pyautogui.moveTo(typesetCenter, duration=0.25)
>>> pyautogui.click()
>>> time.sleep(2)
>>> pyautogui.press(`enter')
```

Additional Examples

Example: Writing and Compiling LaTeX Documents



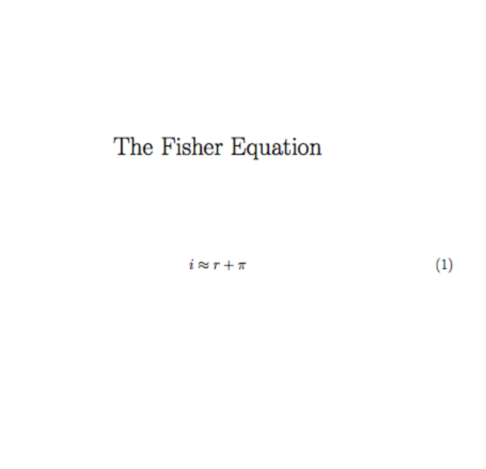
```
\documentclass[12pt]{article}

\begin{document}

\title{The Fisher Equation}

\author{}

\date{}

\maketitle

\begin{equation}
i \approx r + \pi
\end{equation}

\end{document}
```

Additional Examples

Example: Writing and Compiling LaTeX Documents

Additional Examples

Example: Writing and Compiling LaTeX Documents



The Fisher Equation

$$i \approx r + \pi \tag{1}$$

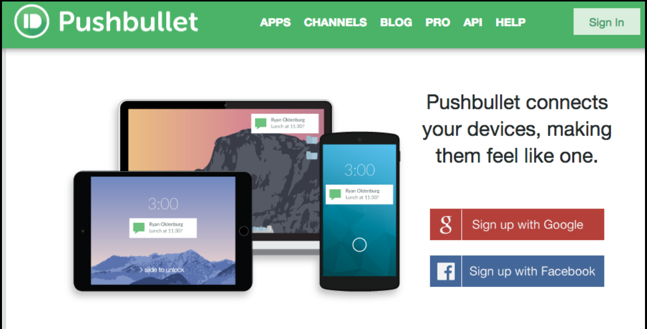
Additional Examples

Example: Send Messages to Any Device with Pushbullet

1. Create a Pushbullet account
2. Install Pushbullet on devices
3. Recover Pushbullet credentials
4. Install Pushbullet module for Python
5. Send status messages via Python

Additional Examples

Example: Send Messages to Any Device with Pushbullet

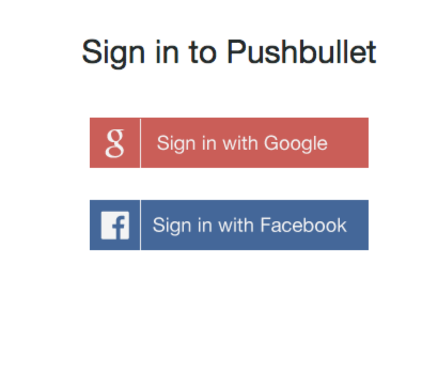


Pushbullet connects your devices, making them feel like one.

https://www.pushbullet.com/

Additional Examples

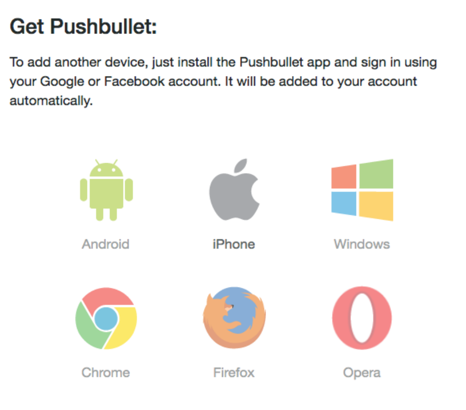
Example: Send Messages to Any Device with Pushbullet



Sign in to Pushbullet

Additional Examples

Example: Send Messages to Any Device with Pushbullet

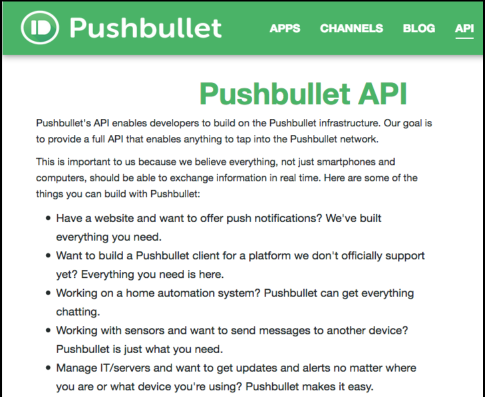


Get Pushbullet:

To add another device, just install the Pushbullet app and sign in using your Google or Facebook account. It will be added to your account automatically.

Additional Examples

Example: Send Messages to Any Device with Pushbullet



Pushbullet API

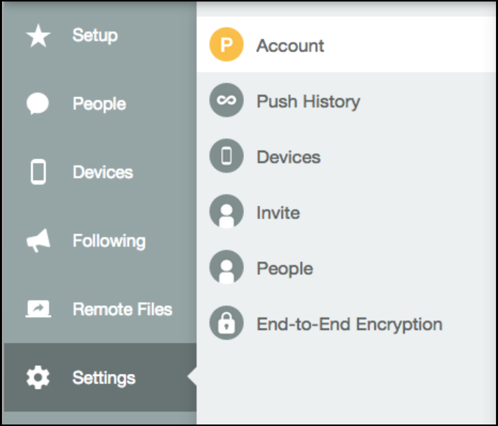
Pushbullet's API enables developers to build on the Pushbullet infrastructure. Our goal is to provide a full API that enables anything to tap into the Pushbullet network.

This is important to us because we believe everything, not just smartphones and computers, should be able to exchange information in real time. Here are some of the things you can build with Pushbullet:

- Have a website and want to offer push notifications? We've built everything you need.
- Want to build a Pushbullet client for a platform we don't officially support yet? Everything you need is here.
- Working on a home automation system? Pushbullet can get everything chatting.
- Working with sensors and want to send messages to another device? Pushbullet is just what you need.
- Manage IT/servers and want to get updates and alerts no matter where you are or what device you're using? Pushbullet makes it easy.

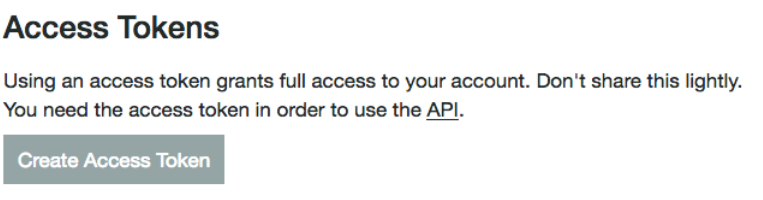
Additional Examples

Example: Send Messages to Any Device with Pushbullet



Additional Examples

Example: Send Messages to Any Device with Pushbullet



Access Tokens

Using an access token grants full access to your account. Don't share this lightly. You need the access token in order to use the API.

Additional Examples

Example: Send Messages to Any Device with Pushbullet

```
#lecture2_example13.py

$user pip install pushbullet.py
$user python
>>> import numpy as np
>>> import datetime
>>> from pushbullet import Pushbullet
```

Additional Examples

Example: Send Messages to Any Device with Pushbullet

```
#lecture2_example13.py (continued)

# Enter Pushbullet credentials.

>>> apiKey = `REPLACE_WITH_YOUR_API_KEY'
>>> pb = Pushbullet(apiKey)

# Construct today's date string.

>>> body = str(datetime.datetime.now())
```

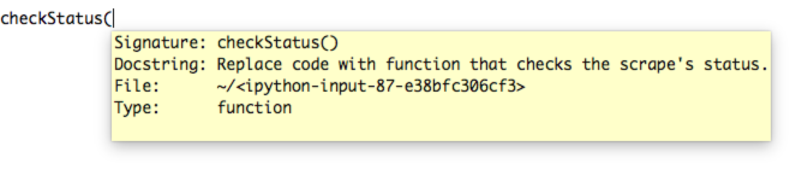
Additional Examples

Example: Send Messages to Any Device with Pushbullet

```
#lecture2_example13.py (continued)

# Check scrape status
>>> def checkStatus():
            """
            Replace code with function that checks
            the scrape’s status.
            """
            random_number = np.random.rand(1)[0]
            if(random_number<=.5)
                status = `Scrape status:  Succeeded.’
            if(random_number>.5)
                status = `Scrape status:  Failed.’
            return status
```

Additional Examples

Example: Send Messages to Any Device with Pushbullet



```
checkStatus(
           Signature: checkStatus()
           Docstring: Replace code with function that checks the scrape's status.
           File:      ~/<ipython-input-87-e38bfc306cf3>
           Type:      function
```

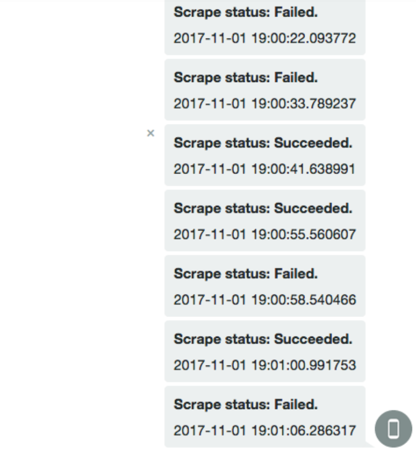
Additional Examples

Example: Send Messages to Any Device with Pushbullet

```
#lecture2_example13.py (continued)
# Check scrape status
>>> title = checkStatus()
# Send status message to all devices.
>>> pb.push_note(title, body)
```

Additional Examples

Example: Send Messages to Any Device with Pushbullet



Scrape status: Failed.

2017-11-01 19:00:22.093772

Scrape status: Failed.

2017-11-01 19:00:33.789237

Scrape status: Succeeded.

2017-11-01 19:00:41.638991

Scrape status: Succeeded.

2017-11-01 19:00:55.560607

Scrape status: Failed.

2017-11-01 19:00:58.540466

Scrape status: Succeeded.

2017-11-01 19:01:00.991753

Scrape status: Failed.

2017-11-01 19:01:06.286317

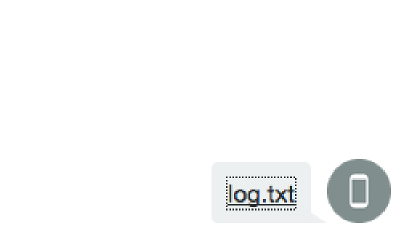
Additional Examples

Example: Send Messages to Any Device with Pushbullet

```
>>> with open(“log.txt”, “r”) as doc:
          file_data = pb.upload_file(doc, “log.txt”)
          push = pb.push_file(**file_data)
```

Additional Examples

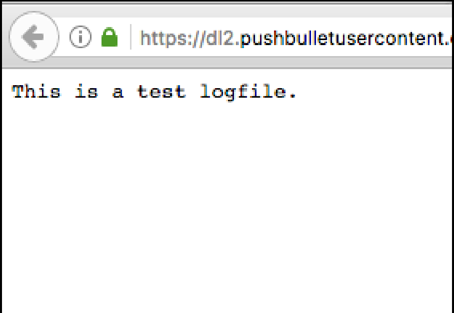
Example: Send Messages to Any Device with Pushbullet



log.txt

Additional Examples

Example: Send Messages to Any Device with Pushbullet



https://dl2.pushbulletusercontent.

```
This is a test logfile.
```

Additional Examples

Example: Queuing Up Tasks with Pushbullet

```
#lecture2_example14.py

$user python
>>> import numpy as np
>>> import pandas as pd
>>> from pushbullet import Pushbullet
>>> import time
>>> import datetime
>>> import re
```

Additional Examples

Example: Queuing Up Tasks with Pushbullet

```
#lecture2_example14.py (continued)

# Generate and save sample data.

>>> def generateRandomWalk():
        days = 10000
        start_date = datetime.datetime.now() +
        datetime.timedelta(-days)
        datelist = pd.date_range(start_date, periods=days).tolist()
        shocks = np.random.normal(0,1,days)
        randomWalk = np.zeros((days,1))
        randomWalk[0] = shocks[0]
        for t in range(1,len(randomWalk)):
           randomWalk[t] = randomWalk[t -1] + shocks[t]
        data = pd.DataFrame(randomWalk,columns=[`random_walk'])
        data.index = datelist
        return data
```

Additional Examples

Example: Queuing Up Tasks with Pushbullet

```
#lecture2_example14.py (continued)

# Generate and save sample data.

>>> data = generateRandomWalk()
>>> data.to_csv(`../data.csv')
>>> print data
```

Additional Examples

Example: Queuing Up Tasks with Pushbullet

```
                          random_walk
1990-06-16 22:37:21.792702 1.747387
1990-06-17 22:37:21.792702 0.209479
1990-06-18 22:37:21.792702 1.061580
1990-06-19 22:37:21.792702 2.010485
1990-06-20 22:37:21.792702 2.379742
...
2017-10-28 22:37:21.792702 52.869201
2017-10-29 22:37:21.792702 53.321906
2017-10-30 22:37:21.792702 52.720864
2017-10-31 22:37:21.792702 55.108963
```

Additional Examples

Example: Queuing Up Tasks with Pushbullet

```
#lecture2_example14.py (continued)

# Define function to compute mean between selected
dates.

>>> def computeMeans(date):
        data = pd.read_csv(`../data.csv',
        index_col=0)
        mean = np.mean(data[dates[0]:dates[1]])[0]
        return mean
```

Additional Examples

Example: Queuing Up Tasks with Pushbullet

```
#lecture2_example14.py (continued)

# Enter PushBullet credentials.

>>> apiKey = `REPLACE_WITH_YOUR_API_KEY'
>>> pb = Pushbullet(apiKey)

# Pull the latest push.

>>> latestPush = pb.get_pushes()[0]
```

Additional Examples

Example: Queuing Up Tasks with Pushbullet

```
#lecture2_example14.py (continued)

# Get Unix time.

>>> unixTime = time.time()

# Get time since latest message sent.

>>> timeSinceLatest =
unixTime-latestPush[`created’]

# Define regular expression.

>>> pattern = re.compile(`\d{4}-\d{1,2}-\d{1,2}’)
```

Additional Examples

Example: Queuing Up Tasks with Pushbullet

```
#lecture2_example14.py (continued)

# Check if dates have been pushed within the last 10
minutes.

>>> body = latestPush[`body']
>>> dates = re.findall(pattern, latestPush[`body'])
>>> if timeSinceLatest<=600 and len(dates)==2:
        mean = computeMean(dates)
```

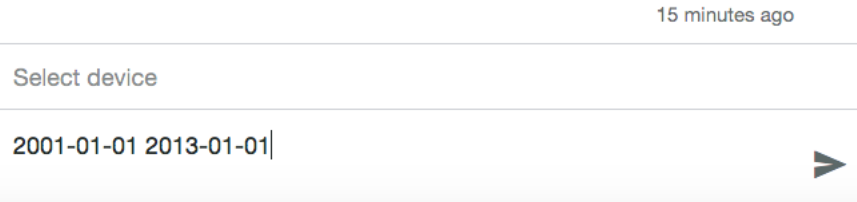
Additional Examples

Example: Queuing Up Tasks with Pushbullet

```
#lecture2_example14.py (continued)

# Send status message to all devices.

>>> pb.push_note(`Mean:  '+str(mean), `Start
'+dates[0]+`, '+'End '+dates[1])
```

Additional Examples

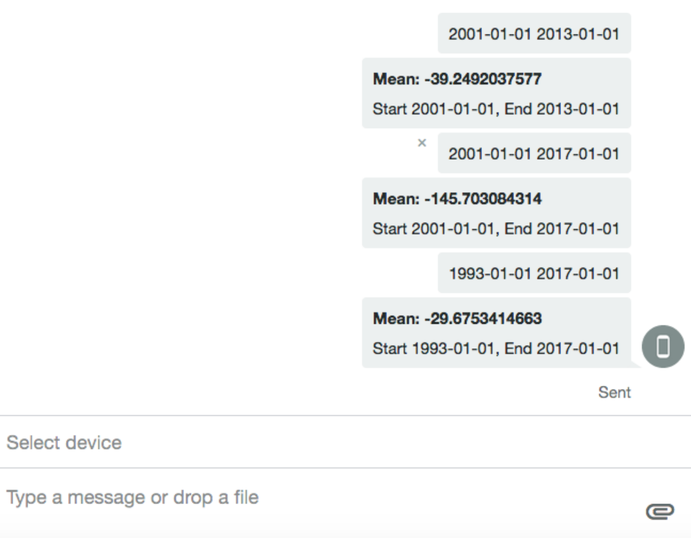
Example: Queuing Up Tasks with Pushbullet



2001-01-01 2013-01-01

Additional Examples

Example: Queuing Up Tasks with Pushbullet



2001-01-01 2013-01-01

Mean: -39.2492037577

Start 2001-01-01, End 2013-01-01

×

2001-01-01 2017-01-01

Mean: -145.703084314

Start 2001-01-01, End 2017-01-01

1993-01-01 2017-01-01

Mean: -29.6753414663

Start 1993-01-01, End 2017-01-01

Additional Examples

Example: Running Python on a Linux Virtual Private Server

1. Select a Linux VPS service.

2. Generate VPS with Ubuntu 16.04.

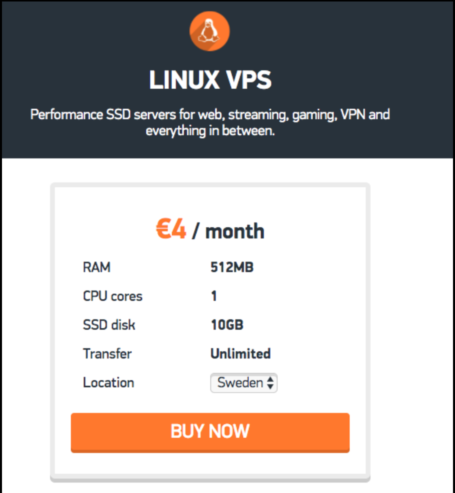
3. Get IP address and password.

4. Access VPS terminal via ssh.

5. Install Anaconda.

Additional Examples

Example: Running Python on a Linux Virtual Private Server



LINUX VPS

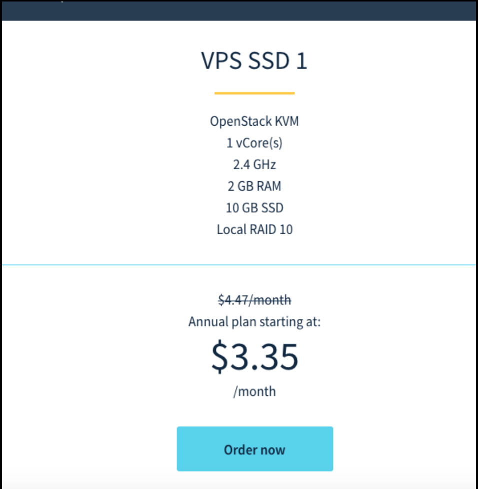
Performance SSD servers for web, streaming, gaming, VPN and everything in between.

€4 / month

RAM	512MB
CPU cores	1
SSD disk	10GB
Transfer	Unlimited
Location	Sweden

Additional Examples

Example: Running Python on a Linux Virtual Private Server



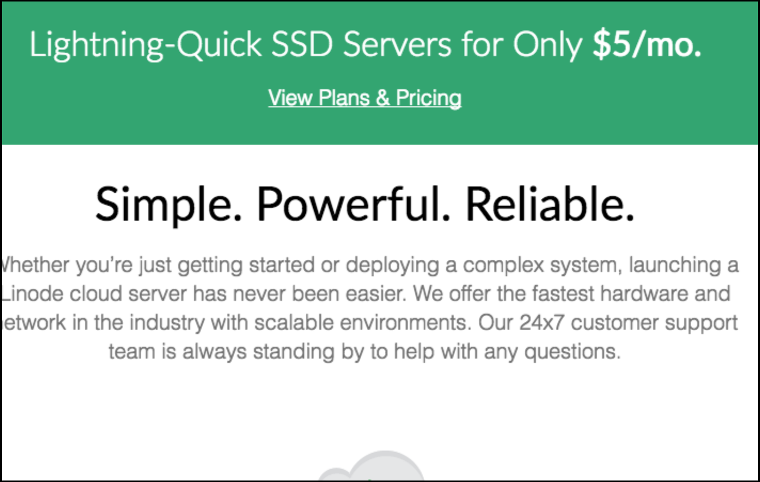
VPS SSD 1

OpenStack KVM
1 vCore(s)
2.4 GHz
2 GB RAM
10 GB SSD
Local RAID 10

~~$4.47/month~~
Annual plan starting at:
$3.35
/month

Additional Examples

Example: Running Python on a Linux Virtual Private Server



Lightning-Quick SSD Servers for Only **$5/mo.**

View Plans & Pricing

Simple. Powerful. Reliable.

Vhether you're just getting started or deploying a complex system, launching a Linode cloud server has never been easier. We offer the fastest hardware and etwork in the industry with scalable environments. Our 24x7 customer support team is always standing by to help with any questions.

Additional Examples

Example: Running Python on a Linux Virtual Private Server

```
# Access server via ssh and install text editor.

$user ssh root@x.x.x.x
root@x.x.x.x’s password:  *****
$root apt-get install nano
```

Additional Examples

Example: Running Python on a Linux Virtual Private Server

```
# Download and install Anaconda.

$root cd /tmp
$root curl -O
https://repo.continuum.io/archive/Anaconda34.2.0-
Linuxx86_64.sh
$root sha256sum Anaconda34.2.0Linuxx86_64.sh
$root bash Anaconda34.2.0Linuxx86_64.sh
$root source ∼.bashrc
```

Additional Examples

Example: Running Python on a Linux Virtual Private Server

```
# Upload scripts to server.

$root exit
Connection to x.x.x.x closed.
$user scp -r ../code/* root@x.x.x.x:/code/
```

Additional Examples

Example: Running Python on a Linux Virtual Private Server

```
# Reconnect to server.

$user ssh root@x.x.x.x
root@x.x.x.x's password:  *****
```

Additional Examples

Example: Running Python on a Linux Virtual Private Server

```
# Access crontabs.

$root crontab -e
```

Additional Examples

Example: Running Python on a Linux Virtual Private Server

```
# Setup cronjob for lecture3_example14.py

*/10 * * * * /root/anaconda/bin/python
/root/code/lecture3_example14.py
```

Additional Examples

Example: Running Python on a Linux Virtual Private Server

```
# Setup cronjob for lecture2_example14.py
```

- Use ctrl+x and then y to save and exit.
- * * * * *
  - Minute (0-59)
  - Hour (0-23)
  - Day of Month (1-31)
  - Month (1-12)
  - Day of week (0-6)
- Don't use root

Additional Examples

Example: Running Python on a Linux Virtual Private Server

```
# Setup cronjob for lecture2_example14.py

$root crontab -l
```

Additional Examples

Example: Running Python on a Linux Virtual Private Server

```
*/10 * * * * /root/anaconda/bin/python
/root/code/lecture2_example14.py
```

Summary

- File operations
  - Files and File paths
  - Platform-Neutrality
  - Absolute and Relative Paths
  - Copying Files and Folders
  - Zip Files

Summary

- Debugging
  - Exceptions
  - Assertions
  - Logging

Summary

- Timing, Scheduling, and Launching
  - Time keeping
  - Scheduling
  - Multithreaded Scheduling
  - Launching Applications

Summary

- Automating Emails and Text Messages
  - Sending automated emails
  - Automated email retrieval
  - Sending text messages

Summary

- Controlling Keyboard and Mouse Movements
  - Coordinate system
  - Delays and failsafes
  - Mouse movements
  - Keyboard controls

Summary

- Additional Examples
    - Treasury auction data
    - Writing and compiling LaTeX documents
    - Sending messages to any device with Pushbullet
    - Queuing up tasks with Pushbullet