Scrapy 101

A python crawler

MIGUEL ANGEL TORRES GOVEA

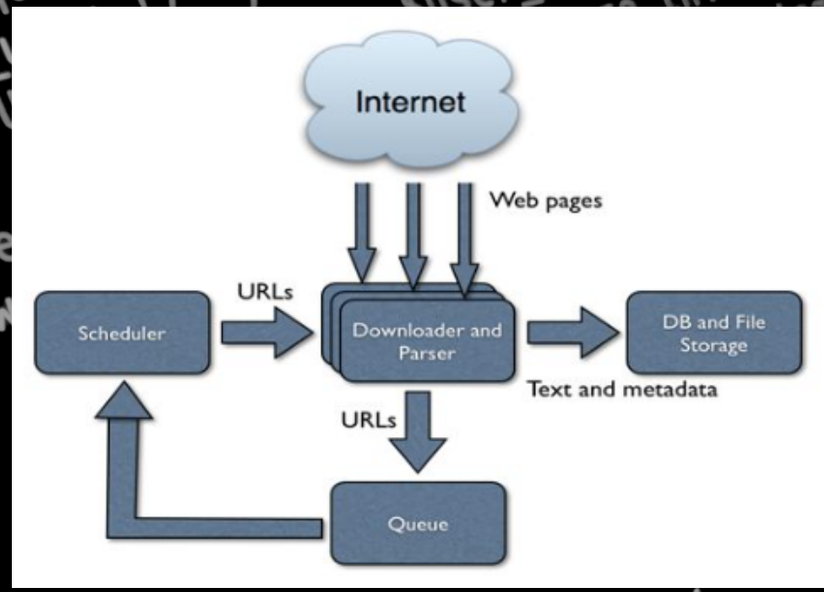
miguel.torres@pricetravel.com - miguel@maf.mx -mafairnet

Acerca de Mi

Y mi trabajo

Ingeniero en Sistemas Computacionales

VoIP/CTI Senior Leader
System Administrator
En PriceTravel Holdings



Crawler

=> Conocido como Webcrawler, web spider o web robot

=> Script o porgrama automatizado que busca contenido en internet

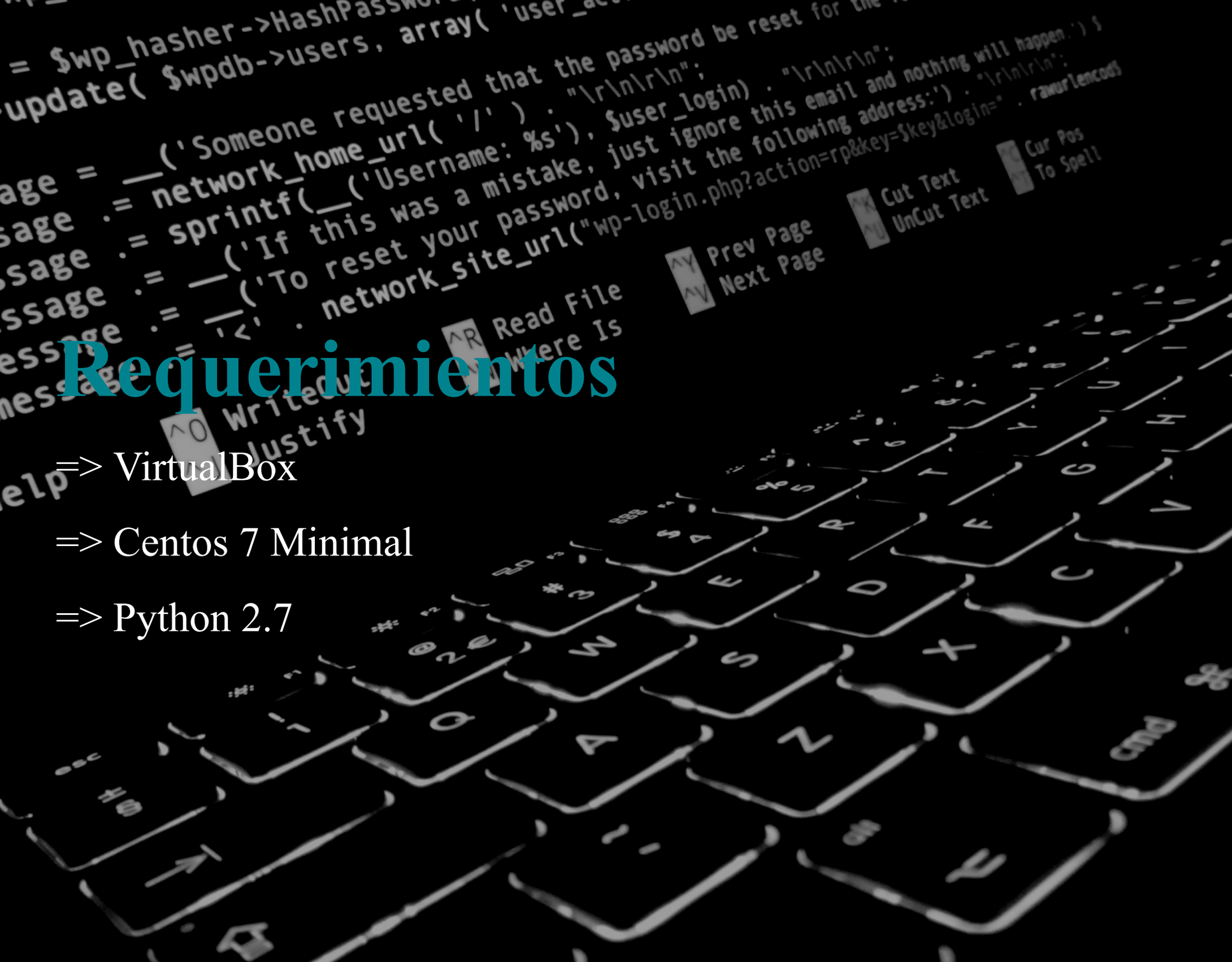
=> Ejemplos mas conocido: Google, Bing

=> Uso fundamental: obtener informacion o datos de sitios

Scrappy

=> Framework que ayuda a crawlear sitios

=> Requiere Python 2.7 en adelante

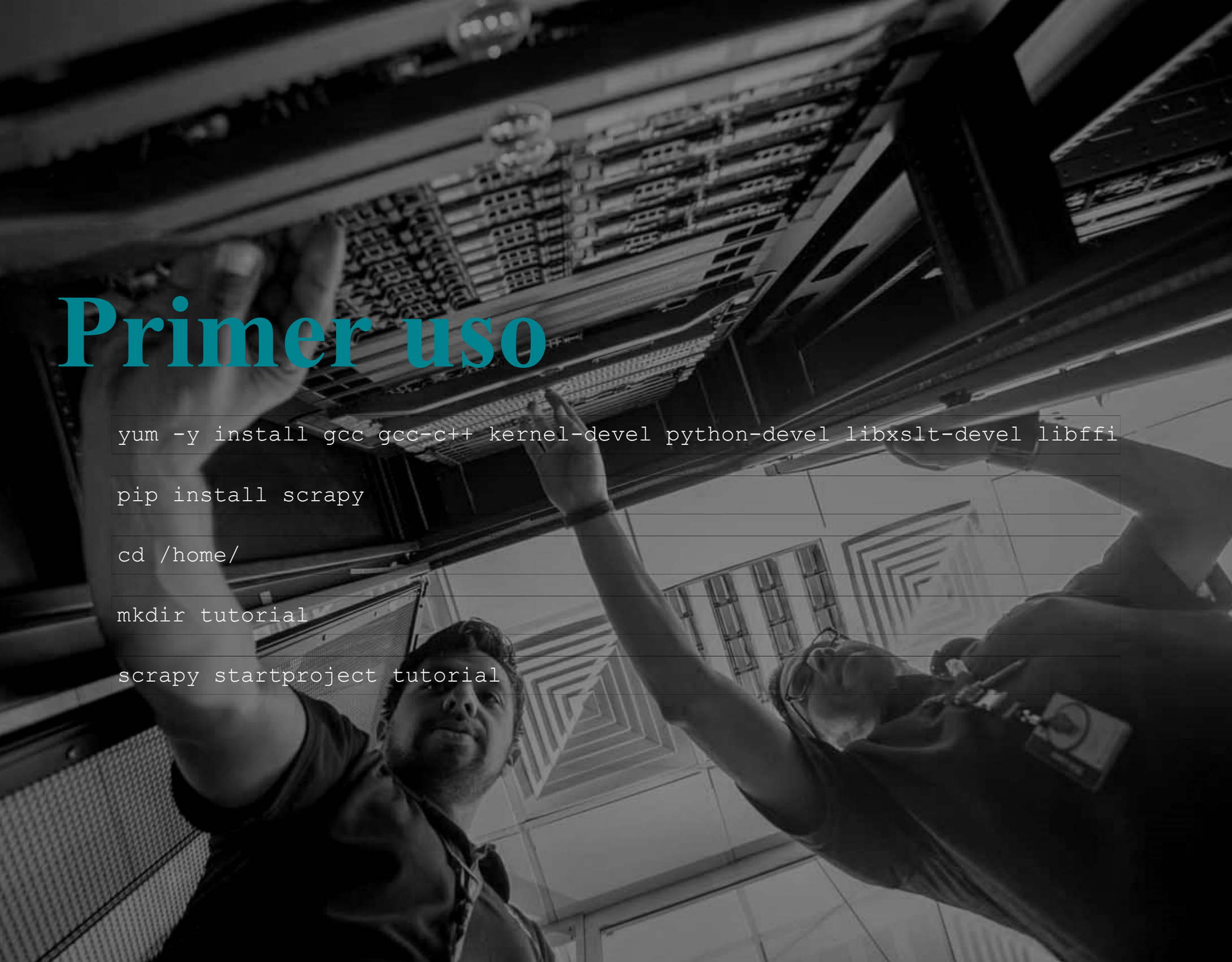


Requerimientos

=> VirtualBox

=> Centos 7 Minimal

=> Python 2.7



Primer uso

```
yum -y install gcc gcc-c++ kernel-devel python-devel libxslt-devel libffi
```

```
pip install scrapy
```

```
cd /home/
```

```
mkdir tutorial
```

```
scrapy startproject tutorial
```

Primer ejemplo

```
nano /home/tutorial/tutorial/tutorial/spiders/quotes
```

```
import scrapy

class QuotesSpider(scrapy.Spider):
        name = "quotes"

        def start_requests(self):
                urls = [
                        'http://quotes.toscrape.com/page/1/',
                        'http://quotes.toscrape.com/page/2/',
                ]
                for url in urls:
                        yield scrapy.Request(url=url, callback=self.parse

        def parse(self, response):
                page = response.url.split("/")[-2]
        filename = 'quotes-%s.html' % page
```

```
scrapy crawl quotes
```

Segundo ejemplo

```
nano /home/tutorial/tutorial/tutorial/spiders/quotescli
```

```
import scrapy

class QuotesSpider(scrapy.Spider):
        name = "quotescli"
        start_urls = [
                'http://quotes.toscrape.com/page/1/',
                'http://quotes.toscrape.com/page/2/',
        ]

def parse(self, response):
        for quote in response.css('div.quote'):
                yield {
                        'text': quote.css('span.text::text').extract_firs
                        'author': quote.css('small.author::text').extract
                        'tags': quote.css('div.tags a.tag::text').extract
```

```
scrapy crawl quotescli
```

Conclusiones

=> Crawler es una herramienta util para obtener informacion

=> Se puede usar en cualquier tipo de pagina

=> Nos ayuda automatizar el proceso tedioso de obtener informacion


Q&A

C'est fini

Muchas gracias

miguel.torres@pricetravel.com
miguel@maf.mx
mafairnet

https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-c

https://lintut.com/how-to-setup-network-after-rhelcentos-7-minimal-installation/

https://serverfault.com/questions/626521/centos-7-save-iptables-settings

https://stackoverflow.com/questions/5768369/drupal-7-file-system-error-the-directory-sites-default-files-exists-b

https://www.liquidweb.com/kb/how-to-install-pip-on-centos-7/

http://flask.pocoo.org/