Angel Pablo Hinojosa

Scraping y Obtención de Datos para Big (y no tan Big) Data (Parte III)

Trabajar en remoto

Conectando con servidores:

FTP
(con Filezilla)

`cabas.ugr.es`

Trabajar en remoto

(not in Kansas anymore)

Trabajar en remoto

Conectando con servidores:

SSH
(con PuTty)

Shell de UNIX básica:

- ls
- cd

- cat
- more
- less

- touch
- rm
- mv

Shell de UNIX básica:

- uname
- top

- grep
- wc
- less
- sort

- nano
- man

Python (aún más) básico

Hola mundo, abrir ficheros y poco más

```
python miprograma.py
```

Scrapy

Modo interactivo

```
scrapy shell "URL"
```

Scrapy

Modo interactivo:

```
response.xpath('//title')
response.xpath('//title').extract()
response.xpath('//h2')
response.xpath('//h2/text()').extract()
response.xpath('//a')
response.xpath('//a/@href')
```

Scrapy

Modo interactivo:

```
fetch(“URL”)
view(response)
quit
```

Scrapy

Nuestro primer scraper:

`http://doc.scrapy.org/en/latest/intro/tutorial.html`

Scrapy

Nuestro primer scraper:

```
scrapy startproject tutorial
```

Scrapy

Editamos items.py:

```
import scrapy


class DmozItem(scrapy.Item):
    title = scrapy.Field()
    link = scrapy.Field()
    desc = scrapy.Field()
```

Scrapy

Creamos dmoz_spider.py

```
import scrapy

class DmozSpider(scrapy.Spider):
    name = "dmoz"
    allowed_domains = ["dmoz.org"]
    start_urls = [
        "http://www.dmoz.org/Computers/Programming/Languages/Python/Books/",
        "http://www.dmoz.org/Computers/Programming/Languages/Python/Resources/"
    ]

    def parse(self, response):
        filename = response.url.split("/")[-2] + '.html'
        with open(filename, 'wb') as f:
            f.write(response.body)
```

Scrapy

Ejecutamos el scraper:

```
scrapy crawl dmoz
```

Scrapy

Ejecutamos el scraper:

```
scrapy crawl dmoz
```

Scrapy

segunda versión de dmoz_spider.py

```
import scrapy

class DmozSpider(scrapy.Spider):
    name = "dmoz"
    allowed_domains = ["dmoz.org"]
    start_urls = [
        "http://www.dmoz.org/Computers/Programming/Languages/Python/Books/",
        "http://www.dmoz.org/Computers/Programming/Languages/Python/Resources/"
    ]

    def parse(self, response):
        for sel in response.xpath('//ul/li'):
            title = sel.xpath('a/text()').extract()
            link = sel.xpath('a/@href').extract()
            desc = sel.xpath('text()').extract()
            print title, link, desc
```

Scrapy

Ejecutamos (otra vez) el scraper:

```
scrapy crawl dmoz
```

```
scrapy crawl dmoz > archivo
```

```
scrapy crawl --nolog dmoz
```

Scrapy

tercera versión de dmoz_spider.py

```
import scrapy, urlparse

from tutorial.items import DmozItem

class DmozSpider(scrapy.Spider):
    name = "dmoz"
    allowed_domains = ["dmoz.org"]
    start_urls = [
        "http://www.dmoz.org/Computers/Programming/Languages/Python/",
    ]

    def parse(self, response):
        for href in response.css("ul.directory.dir-col > li > a::attr('href')"):

            url = urlparse.urljoin(response.url, href.extract())
            yield scrapy.Request(url, callback=self.parse_dir_contents)

    def parse_dir_contents(self, response):
        for sel in response.xpath('//ul/li'):
            item = DmozItem()
            item['title'] = sel.xpath('a/text()').extract()
            item['link'] = sel.xpath('a/@href').extract()
            item['desc'] = sel.xpath('text()').extract()
            yield item
```

Scrapy

Ejecutamos (otra vez) el scraper:

```
scrapy crawl dmoz
```

Gracias

(Ruegos y preguntas)

http://www.psicobyte.com
@psicobyte_
psicobyte@gmail.com