Node.js 크롤링 환경 구축 가이드

‘자바스크립트와 Node.js를 이용한 웹 크롤링 테크닉’ 서적을 구매하신 여러분, 역자 이동규입니다. 책이 발간된 후 독자분들이 보내주신 문의 메일을 바탕으로 입문자를 위한 개발 환경 구축 가이드를 작성하게 되었습니다.

이 책의 가장 큰 장점은 실습 위주로 데이터 수집과 처리 방법에 대해 학습할 수 있다는 점입니다. 그런데 비전공자분들이나 초보자분들은 첫 스텝인 환경 구축에서 좌절하시는 분들이 많으신 거 같습니다.

컴퓨터학을 전공했고 7년 이상 현업에서 일한 저의 눈높이에서는 실습을 위한 모든 내용이 책에 잘 담겨있다고 생각하며 번역을 했는데, 독자분들이 보내주신 문의 메일을 통해 입문자의 시각을 다시 한번 확인할 수 있었습니다. 그래서 이 문서는 철저히 초보자의 관점에서 작성하기 위해 노력하였습니다.

특히, 가상머신과 리눅스 시스템의 개념이 부족하여 실습을 진행하지 못한 경우가 많았기 때문에 해당 내용에 대해 집중 기술하였습니다. 사실 원저자가 가상환경과 리눅스 환경을 채택한 이유는 저자와 독자의 시스템 환경 차이로 인해 발생하는 문제를 원천 봉쇄하기 위한 굉장히 좋은 선택이었지만, 윈도우만 사용해 본 입문자의 입장에서는 오히려 진입 장벽을 높인 감도 있습니다.

일종의 AS(After Service)와 같은 이 문서를 통해 다시 한번 많은 분들이 데이터 수집과 분석의 재미를 느끼는 단계에 이르게 되기를 기대해봅니다.

1) 개발 환경 구축에 관하여
개발 환경 구축 과정 중에 좌절하신 분들께 먼저 용기를 북돋아 드리고 싶습니다. 개발 환경 구축이 곧 개발의 반이라는 말이 있을 정도로 개발 환경 구축은 만만치 않은 일입니다. 개발 환경 구축이 어려운 이유는 각자가 가지고 있는 컴퓨터의 하드웨어, 운영체제의 종류 및 버전, 설치된 라이브러리에 따라 에러가 발생할 수 있기 때문입니다. 하지만 몇 번 개발 환경을 구축하다 보면 자신감이 생깁니다. 대부분의 경우 에러 메시지를 구글링해 보면 원인과 조치법을 알 수 있기도 합니다.

2) 운영체제, 리눅스, 가상머신에 대하여
책의 내용을 실습하기 위해서는 위 세 가지 내용에 대한 어렴풋한 개념은 알고 있어야 합니다. 먼저 운영체제는 컴퓨터의 하드웨어 바로 윗 단에 설치되는 첫 소프트웨어입니다. CPU, 메모리, 디스크, 모니터 등의 하드웨어를 컨트롤하고 그 위에 여러 프로그램들이 돌아갈 수 있도록 관리하며, 계정 관리 기능도 포함이 되어있지요. 쉽게 생각하면 여러분들이 컴퓨터를 켜면 ‘윈도우’나 ‘MAC OS X’ 등에 로그인하여 ‘인터넷 브라우저’나 ‘워드’, ‘엑셀’등의 프로그램을 사용하는데 ‘윈도우’ 및 ‘MAC OS X’를 운영체제라고 합니다.

여러분들이 알만한 저 두 운영체제 외에도 개발자들이 반드시 알아야 할 운영체제가 있는데 바로 리눅스입니다. 리눅스도 운영체제이기 때문에 저 두 운영체제와 동일한 역할을 수행합니다. 그런데 리눅스는 마우스로 아이콘을 클릭하는 GUI(Graphical User Interface) 방식이 아니라 문자를 입력하여 명령을 내리는 CUI(Character User Interface)방식에

특화되어 있습니다. 이를테면 윈도우에서는 메모장 아이콘을 더블 클릭하면 메모장 프로그램이 기동되는데, 리눅스에서는 시커먼 화면(콘솔)에서 vi 라는 글자를 입력한 후 엔터를 치면 vi라는 문서 편집 프로그램이 기동됩니다.

앞서 운영체제가 하드웨어 바로 위에 설치되는 소프트웨어라고 했습니다. 그런데, 요즘에는 가상화 기술을 통해 소프트웨어적으로 가상의 머신을 만들고 그 위에 새로운 운영체제를 설치할 수 있게 되었습니다. 새로운 컴퓨터를 한 대 더 사지 않아도 가상의 기계를 만들어서 그 위에 원하는 운영체제를 설치할 수 있게 된 것입니다. 이러한 가상머신은 원하는 시점에서 만들고 삭제할 수 있어 편리합니다. 또한 새 컴퓨터 샀을 때 아무런 프로그램이 설치되지 않은 것처럼 깨끗한 상태의 시스템을 쉽게 얻을 수 있습니다.

가상 머신을 사용하면 책을 집필한 저자와 독자가 동일한 운영체제를 사용할 수 있게 됩니다. 따라서 환경의 차이로 인한 에러를 원천 봉쇄할 수 있습니다. 그러므로 독자 여러분들께서 가상 머신을 만들고 접속하는것 까지만 성공하면 책의 내용을 실습하는데 아무런 문제가 없게 되는 것입니다.

책에서는 VirtualBox와 Vagrant라는 소프트웨어를 사용하여 가상 머신 위에 centos라는 운영체제를 설치하여 실습을 진행합니다. centos란 리눅스의 여러 종류 중 하나입니다.

책 12페이지의 내용에 따라 두 소프트웨어를 설치하시기 바랍니다. 그런데 혹시 소유하신 컴퓨터가 매우 오래되었거나 기능이 축소, 특화된 컴퓨터인 경우 설치가 안 되거나 설치가 되어도 정상적으로 동작하지 않을 수 있습니다. 요즘 시대에 흔한 케이스는 아닌데 혹시라도 이에 해당되신다면 세 가지 제안을 드리고 싶습니다.

첫 번째 제안은 개발을 위한 노트북을 하나 구매하는 것입니다. 개발을 위한 컴퓨터는 SSD장착, 메모리 4GB 이상을 추천드리며 싸게는 40만원 정도에 구매가 가능합니다.

두 번째 제안은 클라우드 컴퓨팅 서비스를 이용하는 겁니다. 클라우드 컴퓨팅 서비스란 인터넷을 통해 원하는 성능의 컴퓨터를 원하는 시간동안 사용하고 사용한만큼만 돈을 내면 되는 서비스를 말합니다. 유명한 업체로는 Amazon AWS, MS Azure, Google 등이 있고, 가입 후 일정 기간 동안 공짜로 사용할 수 있는데 책의 내용을 실습하기에 충분한 시간이 제공됩니다. 다만, 사용한 머신을 끄는 것을 잊어서 무료 사용 시간을 초과 사용하여 과금되지 않도록 주의해야 합니다. 클라우드 컴퓨팅을 사용하는 방법은 방대한 내용이기 때문에 웹 상에 공개한 다음 URL을 참고하여 진행하시기를 바랍니다.

아마존 웹 서비스를 다루는 기술

https://www.pyrasis.com/aws.html

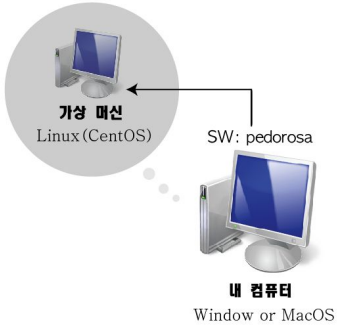
마지막 방법은 가상 환경을 포기하고 사용하고 계신 머신에다 바로 nodejs를 설치하여 실습을 진행하는 것입니다. 교재 475쪽의 내용을 참고하여 설치를 진행하면 됩니다. 그러나 이 경우, 책의 내용을 그대로 실습하기 힘들어 예기치 못한 동작의 차이가 발생할 수 있어 초보자들께는 권하지 않습니다.

3) 리눅스 터미널 접속

독자분들께 많이 문의 받은 질문입니다. pedorosa란 무엇인가? pedorosa는 리눅스 시스템에 접속하기 위한 윈도우용 프로그램입니다. 리눅스 시스템에 접속한다는 말은 하나의 컴퓨터에서 네트워크를 통해 저 멀리 있는 다른 컴퓨터의 리눅스 시스템에 로그인하여 명령을 내리는 것을 말합니다. 가상머신의 경우 네트워크 넘어에 있지 않고 현재 컴퓨터에 있기 때문에 IP주소를 127.0.0.1 (localhost: 현재 컴퓨터를 의미)로 지정하면 되고 만약에 클라우드 컴퓨팅 서비스를 사용한다면 원격지의 공인 IP를 입력해야 합니다.

여러분이 사용하는 컴퓨터에 윈도우가 설치되어 있다면 pedorosa를 설치하고 가상 머신의 리눅스에 접속하여 실습을 진행하면 됩니다. 비슷한 프로그램으로 putty라는 프로그램도 있습니다.

한편 Mac OS에서는 pedorosa라는 프로그램 없이도 기본으로 깔려있는 터미널이라는 앱을 통해 리눅스에 접속할 수 있습니다. Spotlight 검색(control + space) 에서 '터미널'이라고 입력하면 터미널 앱을 기동할 수 있습니다.



4) 정말 가상머신에 접속한게 맞나요?
간혹 독자분들중에서는 가상 머신을 만들고 나서도 가상 머신에 접속하지 않고 윈도우에서 명령어를 시도하는 경우도 있었습니다. 마치 영화 인셉션에서 꿈과 현실을 구분지 못하는 것처럼 혼동을 일으키신 경우입니다. 다시 한번 정리하지만 저희의 실습은 가상 머신을 만들고 그 위에 설치한 centos라는 리눅스 운영체제에서 진행을 합니다.

마치 꿈 속에서 볼을 꼬집어 보는 것처럼 현재 가상머신에 접속한게 맞는지 확인하는 방법을 알려드리겠습니다.

아래 명령어를 입력했을 때
grep . /etc/issue*

다음과 같이 출력되면 현재 centos에 접속된 상태인 것입니다.

/etc/issue:CentOS release 6.8 (Final)
/etc/issue:Kernel \r on an \m
/etc/issue.net:CentOS release 6.8 (Final)
/etc/issue.net:Kernel \r on an \m

그런데 전혀 다른 에러 메시지가 출력된다면 윈도우의 pedorosa나 Max OS의 터미널 앱을 통해 centos에 접속부터 해야 합니다.

5) 리눅스에 접속 후
가상머신을 만들고 리눅스에 접속했다면 이제부터는 리눅스 시스템에 대한 기본적인 조작법을 익히셔야 합니다. 리눅스에 대한 이해가 부족하여 실습을 진행하지 못한 경우도 너무 많았습니다. 리눅스 운영체제는 앞서 설명했듯이 명령어를 글자로 일일이 입력해서 프로그램을 수행하게 됩니다. 다음 몇 가지 명령어를 직접 실습하면서 확실히 익히시길 바랍니다.

리눅스에 로그인했으면 현재 있는 위치를 확인해 봅니다.

$ pwd
/home/vagrant

한 폴더 위로 이동해 봅니다.

$ cd ..

여기서 다시 한번 현재 위치를 확인해 봅니다.

$ pwd
/home

현재 폴더에 있는 파일이나 폴더가 어떤 게 있는지 확인해 봅니다.

$ ls
vagrant

다시 vagrant라는 폴더로 이동합니다.

$ cd vagrant

간단히 현재 폴더를 확인하고, 이동하는 방법을 알아봤습니다. 위 명령어를 포함하여 자주 사용하는 명령어들을 정리하면 다음과 같습니다.

명령어	의미	사용예
pwd	현재 디렉토리 출력	pwd
ls	현재 디렉토리에 있는 파일, 폴더 리스트 출력	ls
cd	현재 디렉토리 변경	cd .. 한 폴더 위로 이동 cd / 루트 폴더로 이동 cd vagrant 현재 폴더 밑에 있는

		vagrant라는 폴더로 이동
cp	파일 복사	cp a.txt b.txt 이미 있는 a.txt와 동일한 내용의 b.txt를 생성
rm	파일 삭제	rm a.txt a.txt를 삭제
mv	파일 이동	mv a.txt ../a.txt 현재 폴더에 있는 a.txt를 한 폴더 위로 이동
date	현재 시간 확인	date
cat	파일 출력	cat a.txt a.txt의 모든 내용이 출력됨
tail	파일의 마지막 부분 출력	tail a.txt a.txt의 마지막 내용이 출력됨

지금까지 윈도우만 사용해 온 독자의 경우, 꺼먼 화면 상에 명령어를 입력하고 그 결과를 확인하는 것이 어색하고 불편하게 느껴질 수도 있습니다. 그러나 본질적으로는 GUI나 CUI나 운영체제에 명령을 내린다는 측면에서 동일합니다. 콘솔을 통해 명령어를 입력하는 CUI가 익숙해지면 GUI보다 훨씬 더 생산성 높게 컴퓨터를 다룰 수 있게 됩니다.

6) node.js설치

centos가 설치된 가상머신에 접속하는 것에 성공했다면 교재 19쪽의 내용을 그대로 입력하기 시작하면 nodejs를 무사히 설치할 수 있습니다.

7) vi 사용법

19페이지의 내용을 따라하다보면 vi를 사용하게 됩니다. vi 에디터는 리눅스 초보자에게 큰 어려움일 수 있어 여기서 조금 더 보충 설명을 합니다.

vi에디터는 윈도우의 메모장처럼 파일을 읽고 편집할 수 있는 프로그램입니다. 하지만 마우스를 조작하지 않기 때문에 다소 독특한 편집 인터페이스가 존재합니다.

vi 로 a.txt라는 파일을 편집하려면 다음과 같이 입력합니다.

vi a.txt

그러면 a.txt의 내용이 화면에 보일 것이고, 커서를 이동하면서 파일의 내용을 살펴볼 수 있지만 키보드를 눌러봐도 생각처럼 편집이 되지 않을 것입니다. vi에서 파일에 편집하기 시작하려면 i 라는 키를 눌러야 합니다. insert의 약어입니다. 편집이 끝났으면 esc키를 누른 후 :w 라고 입력하면 파일이 저장됩니다. 그리고 :q 라고 입력하면 vi를 빠져나오게 됩니다.

vi 에는 훨씬 더 많은 명령어가 존재하지만 기본적인 파일을 조작하기 위한 필수적인 내용만 언급하였습니다.

8) vagrant 설정 파일 관련
책 21페이지를 보면 가상 머신과 호스트 머신간의 폴더 공유 관련 내용이 있습니다. 호스트 머신과 가상 머신간에 폴더를 공유할 수 있는데 이 내용을 비유하자면 꿈을 꾸러 가는데 현실의 물건을 들고 들어갈 수 있고, 꿈에서 만든 물건을 다시 현실로 가져올 수 있는 것이라 할 수 있습니다. 꿈에서 마음껏 돈을 만들어서 현실로 가져올 수 있다면 참 신나겠지요.

하지만 이때 주의해야 할 점이 있습니다. “가상 머신의 경로"를 “/home/vagrant” 와 같이 하면 로그인할 수 없게 되기 때문입니다. 그 이유는 /home/vagrant에는 로그인 인증 정보가 담긴 .ssh 폴더가 있기 때문입니다. 그래서 보통 다음과 같이 설정을 합니다.

config.vm.synced_folder "./", "/home/vagrant/sync"

9) node.js를 설치했다면
node.js를 설치했다면 교재에 나와있는 것처럼 리눅스 커맨드 라인에서 node라고 입력한 후 3+5 를 꼭 입력해 보도록 합니다. 그리고 8이 출력되는 것을 확인하도록 합니다.

10) 여기까지 진행했다면
지금까지 설명한 내용을 바탕으로 가상머신에 node.js까지 설치가 되었다면 책의 내용을 그대로 입력하면서 실습하는데 문제가 없을 것입니다. 책의 내용을 그대로 옮겨적을 때는 혹시 오타가 발생하지 않는지 반드시 주의를 기울여 주시기 바랍니다.

11) 입문자를 위한 책 보는 방법
‘자바스크립트와 Node.js를 이용한 웹 크롤링 테크닉’은 실로 방대한 내용을 담고 있습니다. 크롤링과 관련된 주변 기술들을 참 많이 소개하고 있습니다. 개발 경험이 어느 정도 있다면 빠르게 실습을 진행 해 보면서 다양한 기술들의 핵심적인 내용을 맛볼 수 있어 참으로 유익하고 즐거운 학습 체험을 할 수 있다고 생각합니다.

그러나 비전공자분이나 초보자분들이라면 자바스크립트 문법이나 프로그래밍의 기본적인 구조 및 리눅스 시스템에 대한 자세한 설명을 포함하고 있지 않기 때문에 책의 내용 전체를 이해하기는 어려울 수 있습니다. 그래도 이 책을 구매한 목적이 웹 상의 데이터를 수집하는 기술을 익히는 것에 있다면 소정의 목적을 달성할 수 있다고 생각합니다. 웹 상의 데이터를 수집하기 위해서는 html parsing, css선택자, 정규식, 파일 저장 등의 기술을 기본으로 알고 있어야 합니다. 거기에 웹 브라우저를 자동화하는 phantomjs, casperjs를 알면 더 강력하게 웹 데이터를 수집할 수 있습니다. 위에서 언급한 내용을 바탕으로 책을 읽어나가신다면 소정의 목적을 달성할 수 있다고 생각하며, 나중에 더 경험이 쌓인 상태에서 데이터베이스 사용법, 형태소 분석 및 머신 러닝 관련된 내용까지 실습한다면 더할 나위 없이 좋겠습니다.