파이썬 라이브러리를 활용한 세상에서 제일 쉬운 데이터 분석 수업

실무에 바로 써먹는 파이썬 라이브러리부터
실무 데이터를 활용한 실전 분석 실습까지

Introduction

Data Science, Mathematic, Programming, Machine-Learning

01

Data Science Competency

Mathematic

- Linear Algebra
- Calculus
- Optimization
- Probability

Programming

- Python, R, C etc.
- OS
- DB
- Docker, VM etc.

Data Analysis

- Probability
- Test & Estimate
- Regression etc.
- ML

Domain Competency

- Data -Understanding
- Preprocessing
- Feature -engineering

02

How Start Data Analysis

파이썬

그리고, 실습

02

How Start Data Analysis

재밌는 것 부터!

딱딱한 기초 지식..
기초라지만, 몰라도 시작은 할 수 있다.

02

How Start Data Analysis

어려운 것도 때가 되면
결국 공부하고자 하는 동기부여가
생긴다.

(이상 개인피셜이었습니다.)

03

Data Analysis Process

Collection

- 정형/비정형
- Crawling

Preprocessing

- Cleaning
- Feature-Engineering

Analysis

- Modeling
- Train & Test
- Cross Validation

04

How Execute DA Process

Open Source

이 곳은 없는 거 빼고 다 있는
아름다운 세상

남들이 이미 만들어놓은 소스를
우리는 마음껏 쓸 수 있다.

04

How Execute DA Process

Open Source

=

Library(Package)

05

Using Library : Ex1) Datetime

질문1. 현재 날짜 및 시간을 파이썬으로 알아낼 수 있을까?

In [24]:
```
import datetime
```

In [25]:
```
now = datetime.datetime.now()
now
```

Out[25]: datetime.datetime(2019, 4, 3, 16, 47, 48, 337123)

In [26]:
```
date.year, date.month, date.day, date.hour, date.minute, date.second, date.microsecond
```

Out[26]: (2019, 4, 3, 16, 34, 39, 900974)

05

Using Library : Ex2) Dateutil

질문2. 파이썬으로 연인과의 기념일을 계산할 수 있을까?

In [62]:
```
from dateutil.relativedelta import *
```

In [63]:
```
first = datetime.datetime.strptime("2018-11-17 19:55:34", "%Y-%m-%d %H:%M:%S")
first
```

Out[63]: datetime.datetime(2018, 11, 17, 19, 55, 34)

In [64]:
```
Dday_200 = first + relativedelta(days=+200)
Dday_200
```

Out[64]: datetime.datetime(2019, 6, 5, 19, 55, 34)

05

Using Library : Ex3) Time

질문3. 파이썬 코드가 돌아가는 시간을 계산할 수 있을까?

```
In [74]: import time

In [76]: time.time() #현재 시각
Out[76]: 1554292054.34693

In [77]: time.time() #10초 후
Out[77]: 1554292078.1270132
```

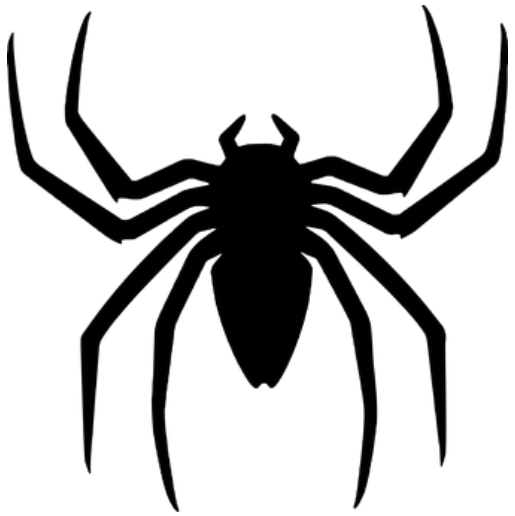
Web Crawler

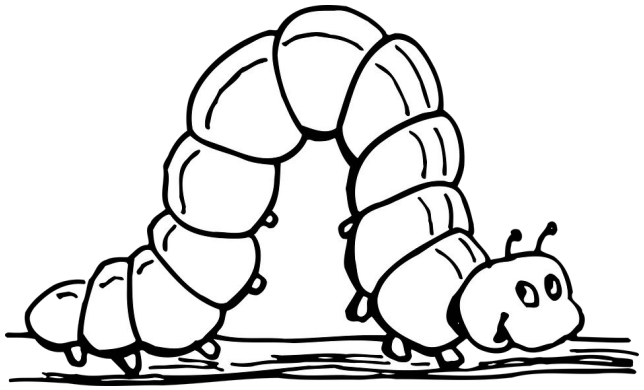
Scrapping, spider, bot, intelligent agent

01

What Crawler?

‘크롤러'를 검색해보면, 거미와 지렁이가 나온다. 이 둘의 공통점은?

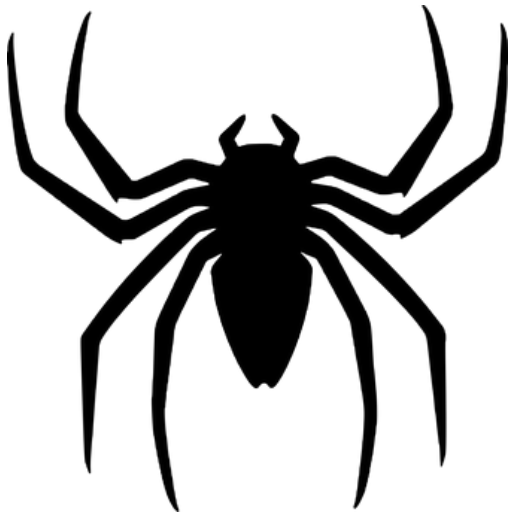


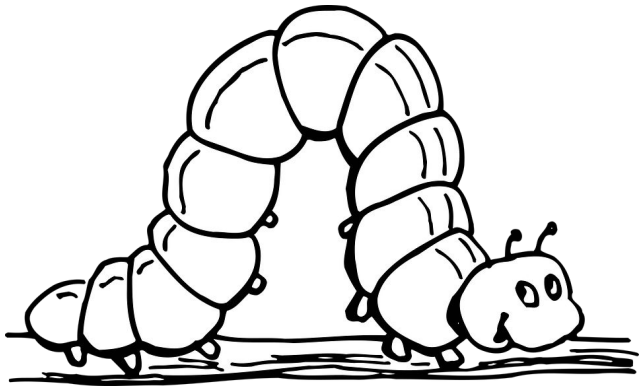


01

What Crawler?

자유로운 객체. 거미는 거미줄, 지렁이는 땅이라는 공간에서 자유롭게 움직이는 객체다.

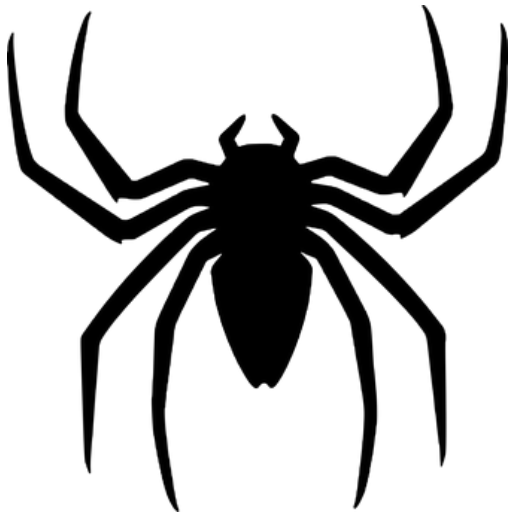


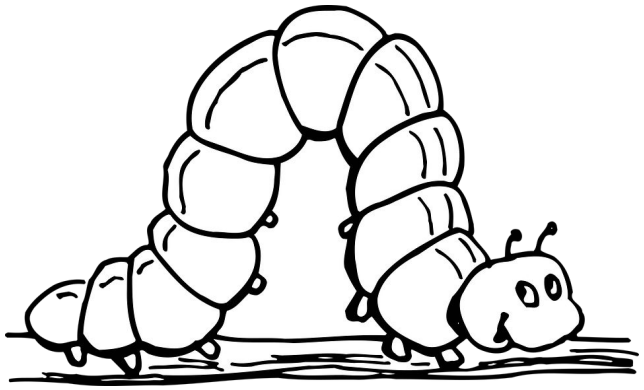


01

What Crawler?

이와같이 우리가 배울 (웹) 크롤러란 WWW(W3)에서 자유롭게 돌아다니는 프로그램이다.





01

What Crawler?

✧ 크롤러 VS 스크래퍼

앞서 말한대로, '크롤러'에는 '자유롭게 돌아다닌다'는 의미가 있다. 다시 말해, 크롤러는 웹이라는 공간에서 자유롭게 다니는 프로그램이고, 이 크롤러가 자유롭게 웹을 돌아다니는 것을 크롤링이라고 한다.

반면에, '스크래퍼'란 '수집한다'는 의미가 있다. 즉, 웹 상의 데이터를 수집하는 프로그램을 일컬어 스크래퍼라고 한다.

그러나, 이 둘을 합쳐 통상적으로는 '**크롤러**로 **크롤링**을 한다'고 표현한다.

02

Why Crawler?

✧ 크롤링을 해야 하는 이유

데이터의 이용 요금을 지불하는 시대. 그만큼 데이터를 얼마나 잘 사용하는지가 중요한 시대에 살고 있다.

✧ 크롤링으로 할 수 있는 것들

검색엔진 활용, 경쟁사 제품 실시간 분석, 온라인 쇼핑몰 최저가 검색, 인스타그램 태그 검색 등 다양하다.

02

Client & Server

크롤링을 이해하기 위해서는 가장 먼저 클라이언트와 서버의 개념을 알아야 한다.

✧ client(클라이언트, 고객)

데이터 혹은 서비스를 요청하는 프로그램. 우리가 사용하는 앱, 웹, 웹 브라우저가 이에 해당한다.

✧ Server(서버, 매장)

인터넷을 통해 연결된 클라이언트에 데이터 혹은 서비스를 제공하는 프로그램.
서버가 없다면 클라이언트도 존재하지 않는다.

02

Client & Server

크롤링을 이해하기 위해서는 가장 먼저 클라이언트와 서버의 개념을 알아야 한다.

✧ client(클라이언트, 고객)

데이터 혹은 서비스를 요청하는 프로그램. 우리가 사용하는 앱, 웹, 웹 브라우저가 이에 해당한다.

✧ Server(서버, 매장)

인터넷을 통해 연결된 클라이언트에 데이터 혹은 서비스를 제공하는 프로그램.
서버가 없다면 클라이언트도 존재하지 않는다.

02

Client & Server

크롤링을 이해하기 위해서는 가장 먼저 클라이언트와 서버의 개념을 알아야 한다.

✧ client(클라이언트, 고객)

데이터 혹은 서비스를 요청하는 프로그램. 우리가 사용하는 앱, 웹, 웹 브라우저가 이에 해당한다.

✧ Server(서버, 매장)

인터넷을 통해 연결된 클라이언트에 데이터 혹은 서비스를 제공하는 프로그램.
서버의 종류는 제공하는 데이터 혹은 서비스의 형태에 따라 다양하다.(영상, 파일, 채팅 등)

02

Client & Server

서버가 없다면 클라이언트도 존재할 수 없다.

데이터 요청

데이터 응답

02

Client & Server

✦ 영상

영상 저장 요청

영상 저장 완료!

영상 요청

영상 제공

02

Client & Server

✦ 채팅

dino에게 '뭐해?' 송신

B에게 '밥먹는 중 너는?' 수신

A에게 '밥먹고있어 너는? 송신

A에게 '뭐해?' 수신

03

How Communicate?

서버와 클라이언트가 서로 데이터를 주고 받는데에 규약이 있다. 이를 **프로토콜**이라고 한다.

✧ 요청 메서드(Client 가 Server 에게)

일명 CRUD. Create, Read, Update, Delete. 이를 직접 명시하지는 않는다.

대신, POST(데이터 생성), GET(데이터 요청), PUT(수정), DELETE(삭제)

물론 이외에도 여러 요청 메서드가 존재하지만, 거의 사용하지 않는다. 이 넷 중에서도 주로 사용하는 것은 POST와 GET 뿐이다.

03

How Communicate?

서버와 클라이언트가 서로 데이터를 주고 받는데에 규약이 있다. 이를 **프로토콜**이라고 한다.

✧ 응답 코드(Server 가 Client 에게)

1xx : 조건부 응답. 데이터 일부만 보내 더 보내야 한다는 신호

2xx : 성공했다는 신호 - 200(요청을 성공적으로 처리), 201(POST로 요청한 자료 성공적으로 저장), 204(반환데이터 없을때) 이 주로 사용

3xx : 새로고침 완료

4xx : 요청 오류 신호. 400(잘못 요청), 401(권한 없음), 402(결제 필요), 403(금지됨), 404(찾을 수 없음, 존재하지 않는 URL 요청), 405(잘못된 요청 메서드 사용) 주로 사용

5xx : 서버 오류 신호. 500(내부 서버 오류), 503(서비스 이용 불가) 주로 사용

03

How Communicate?

✧ **헤더(header)**

서버와 클라이언트 간 데이터를 주고받을 때, 헤더라는 정보를 포함하여 보냅니다.
이는 편지의 보낸 사람, 받는 사람 등의 정보에 비유할 수 있습니다.

클라이언트는 일반헤더 + 요청헤더 + 엔티티 헤더의 형태로 만들어 요청하고,
서버는 일반헤더 + 응답헤더 + 엔티티 헤더의 형태로 응답합니다.

03

How Communicate?

✧ **요청 헤더(Request header)**

클라이언트가 문서와 구성 형태 및 형식을 포함한 헤더.

✧ **응답 헤더(Response header)**

서버가 클라이언트에 응답할 때 요청과 서버의 구성을 포함한 헤더.

✧ **일반 헤더(General header)**

서버와 클라이언트에서 일반적인 정보를 포함하는 헤더

✧ **엔티티 헤더(Entity header)**

메시지에 대한 설명을 포함하는 헤더

04

What URL?

✧ URL 이란?

네트워크 상에서 자원을 요청하는 규약.

✧ URL 구조

프로토콜://주소(or IP):포트번호/리소스경로?쿼리스트링

04

What URL?

프로토콜://주소(or IP):포트번호/리소스경로?쿼리스트링

✧ 프로토콜

인터넷에서 주로 사용하는 프로토콜은 HTTP이다. FTP, SFTP, SSH 등이 이다.

✧ 주소 or IP

데이터를 요청하는 대상.

✧ 포트 번호

프로토콜마다 각자의 고유 포트 번호가 있다. HTTP는 80번이다.

04

What URL?

프로토콜://주소(or IP):포트번호/리소스경로?쿼리스트링

✧ **리소스경로**

해당 사이트에서의 페이지 값. 주소 다음에 /(슬래시)로 시작하는 부분.

✧ **쿼리스트링**

서버에 보내는 데이터 값. 리소스 경로 다음에 ?(물음표)로 시작하는 부분.

변수1=데이터1&변수2=데이터2 … 이런 형식으로 되어 있다.

05

What Web?

✧ WWW(W3)

우리가 말하는 웹이란 World Wide Web의 약자로서, 인터넷에 연결된 컴퓨터를 통해 정보를 공유할 수 있는 전세계적인 공간이다.

✧ 웹브라우저

이러한 공간을 사용할 수 있도록 해주는 것이 웹브라우저다.

✧ 웹사이트

우리가 보는 페이지 하나하나를 웹페이지라고 하며, 이러한 웹페이지의 묶음을 웹사이트라고 한다.

05

What Web?

✧ 웹페이지 구성요소

웹페이지는 크게 세가지로 이루어져 있다.

1. 레이아웃을 잡아주는 **HTML**
2. 기능을 불어넣어주는 **JavaScript**
3. 화면을 꾸며주는 **CSS**