[01.20] 빅데이터

1. 빅데이터

: 너무 크고 복잡해서 기존 RDBMS(관계형 데이터베이스)에서 다루기 어려운 데이터

1. 데이터 과학 : 대량의 데이터를 분석해서 지식을 추출하는 방법
2. 데이터 과학에서 파이썬의 장점 : 큰 성능 저하 없이 프로토타입을 빨리 수행
3. Scale Up & Scale Out

* Scale Up : 서버의 Spec을 높이는 방식 (사이즈 업)
* Scale Out : 서버의 대수를 늘리는 방식

1. 빅데이터 프로세스
2. 문제 정의
   * 무엇을 분석할 것인가?
   * 객관적이고 구체적인 분석 대상의 정의
3. 데이터 수집, 가공
   * 필요한 데이터가 무엇이고 어디서 수집할 것인가?
   * 데이터 구조 및 특성 변경
   * 데이터 오류 확인 및 수정
4. 데이터 모델링
   * 상관분석을 위한 데이터 간 관계를 설정
5. 데이터 시각과 및 분석
   * 데이터 분석을 위한 시각화
   * 데이터에 대한 통찰력 확보
6. Numpy 특징

* 수학 계산을 위한 라이브러리
  + 선형 대수
  + 벡터 및 행렬 연산에 특화 (머신러닝, 인공지능)
  + Array 단위로 데이터를 관리하고 연산

[02.04] HTML

1. HTML5이란?

* 웹 : 세계의 컴퓨터를 연결하여 정보 공유
* 태그 : HTML 문서는 태그로 만들어진다.
  + 시작 태그 : <p>
  + 종료 태그 : </p>
* 속성
* 태그 – 속성 – 속성 값
  + 테이블 태그
  + 타이틀 태그 : 웹사이트 제목
  + 앵커 태그 : 사이트 이동
  + 폼 태그 : 사용자의 정보를 서버로 전달하기 위한 태그
    - 인풋 태그
    - 데이터 전송 (GET : 페이지 받아오기, POST : 데이터 업로드하기)

|  |  |  |
| --- | --- | --- |
| 특징 | GET | POST |
| 기능 | 데이터를 조회 | 데이터를 전송 |
| 데이터 업로드 | url 주소  ? : 파라메타를 보내겠다  & : 그리고  ‘파라메타 이름 = 값’ | Body 안 |
| 용량 제한 | 제한 | 대용량 데이터 전송 가능 |
| 보안 | 취약 | 상대적 안전(그래도 취약) |

* http : 일반적
* https : 암호화

2. 웹 스크래핑이란?

|  |  |  |
| --- | --- | --- |
| 특징 | 스크래핑 | 크롤링 |
| 데이터 획득 | 필요한 데이터만 추출 | 페이지에서 데이터 참조 |
| 데이터 크기(스케일) | 필요한 만큼 | 대부분 크다 |
| 중복 제거 | 필요하다면 | 필수 |
| 필요 모듈 | agent & parser | agent |

* F12 누르면 개발자 툴 보여줌
* 여기서 네모+화살표 누르면 뭐가 뭔지 알려줌
* Xpath : 트리 구조
  + //\*[@id="account"]/a : 네이버 로그인 버튼의 xpath
  + //\*[@id="NM\_FAVORITE"]/div[1]/ul[2]/li[1]/a : 사전의 xpath

3. requests 모듈 : http client 모듈 (정적 데이터에 한에서만 동작함)

4. BeautifulSoup4 : html 구문 분석

5. Selenium + : 동작 페이지 client 모듈 (동적 데이터에 한에서만 동작함)

[02. 07] 정규 표현식 (Regular Expression)

1. 목적

* 패턴 검색 : 문자열에서 원하는 패턴을 찾을 때
* 문자열 변환 : 찾은 패턴을 다른 문자열로 변환
* 유효성 검사 : 이메일, 아이디/패스워드 입력

1. 형태

* /regexr/ i : 시작기호 + 패턴 + 종료기호 + 플래그
* 플래그
  + global
  + multiline : 문장 (해제하면 문단의 처음으로 찾음)
* 패턴
  + | : 또는
  + () : 그룹
  + [] : 문자셋
  + [^] : 부정 문자셋
  + (?:) : 찾지만 기억하지는 않음 (그룹으로 만들지 않음)
* 양
  + ? : 없거나 하나 있거나 (zero or one)
  + \* : 없거나 많거나 (zero or one or more)
  + + : 하나 또는 많이 (one or more)
  + {2} : 2개
  + {2, 3} : 2개랑 3개
* 경계
  + \b : 단어 경계 (공백 기준으로 나뉨, 문단의 시작만 보여줌, 뒤에 입력하면 끝에 보여줌)
  + \B : 단어 경계가 아님 (맨 앞이 아닌 것 보여줌, 뒤에 입력하면 맨 앞 제외하고 보여줌)
  + $ : 뒤에 입력. 맨 뒤에 보여줌
  + ^ : 앞에 입력. 맨 앞에 보여줌
  + . : 모든 문자
  + \[Ya\] : 이스케이프 문자처럼 []을 그대로 보여줌
  + \d : 숫자 보여줌 ([0-9] 해도 됨)
  + \D : 숫자가 제외 모두 출력 ([a-zA-Z] 해도 됨)
  + \w : 문자만 보여줌
  + \W : 문자 제외(특수문자)만 보여줌 (공백 포함)
  + \s : 공백 보여줌

https://regexr.com/

Hi there, Nice to meet you. And Hello there and hi.

I love grey(gray) color not a gry, graay and graaay.

Ya ya YaYaYa Ya

[Ya]

abcdefghijklmnopqrstuvwxyz

ABSCEFGHIJKLMNOPQRSTUVWZYZ

1234567890

.[]{}()\^$|?\*+

010-1234-5678

010 1234 5678

010.1234.5678

02-123-5678

hello@daum.net

[hello@naver@naver.com](mailto:hello@naver@naver.com)

* 전화번호 잡아내기 : \d{2,3}\W\d{3,4}\W\d{4}
* 이메일 잡아내기 : [a-zA-Z0-9.\_]+[@][a-zA-Z0-9.\_]+
* 차량 번호 : \d{2,3}[가-힣]\d{4}