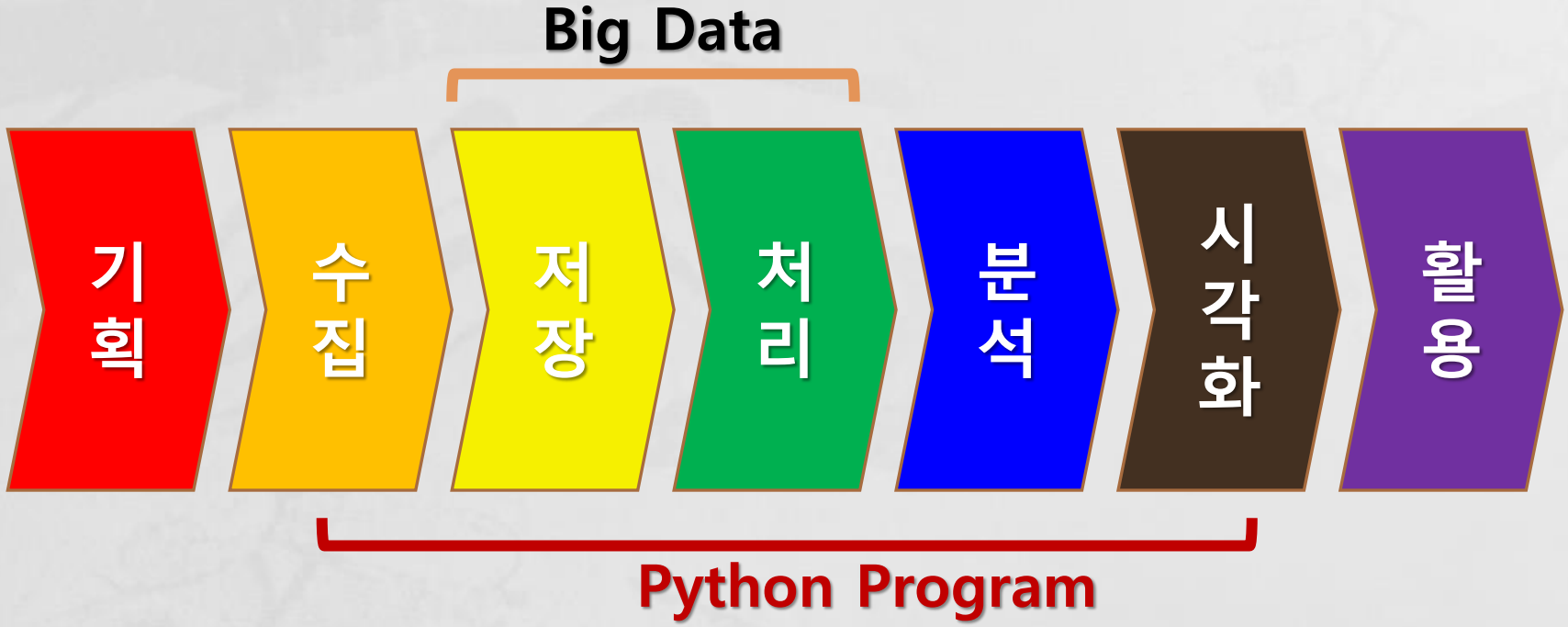
Python Web Crawling

Before we start

Big Data 활용



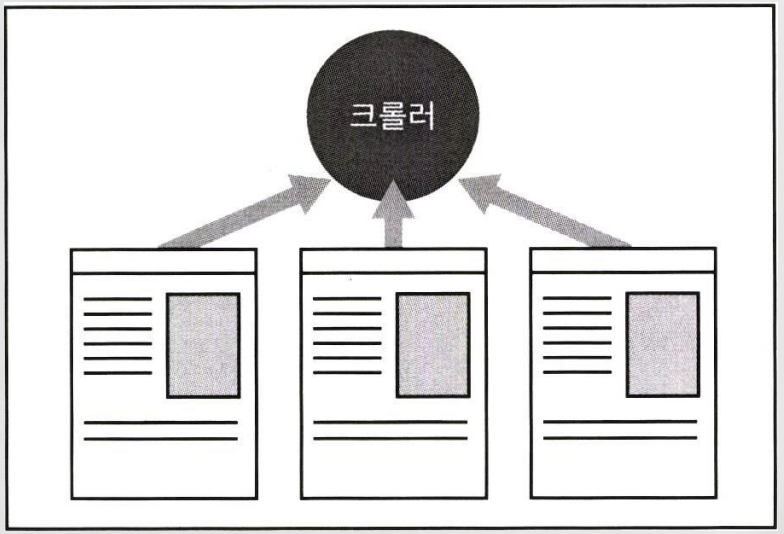
* 분석 목표 설정 -> 필요한 데이터 정의 -> 데이터 수집 및 전처리 -> 분석 -> 인사이트 도출

크롤링과 스크레이핑 개념

크롤링과 스크레이핑

크롤러와 크롤링

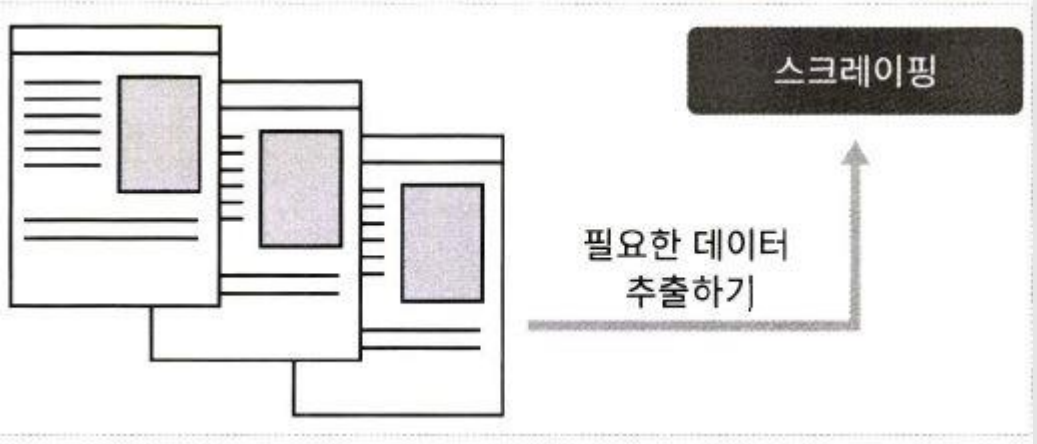
- 크롤러는 자동으로 웹 페이지에 있는 정보를 수집하는 프로그램
- 크롤러는 사람이 브라우저로 웹 페이지를 조회하고 정보를 수집하는 것과 비교할 수 없을 정도로 대규모의 정보를 단시간에 수집
- 크롤러로 정보를 수집하는 일을 '크롤링'



크롤링과 스크레이핑

스크레이핑

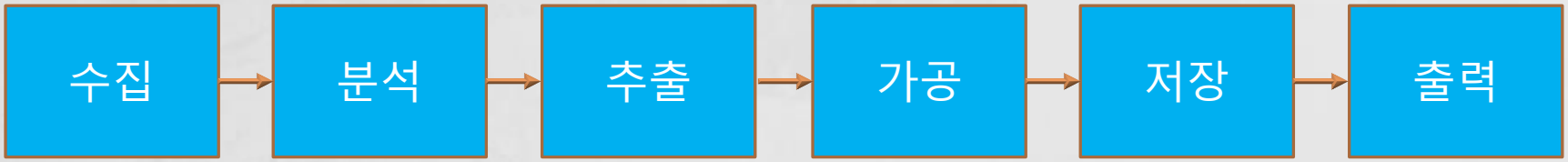
- 스크레이핑은 수집한 정보를 분석해서 필요한 정보를 추출하는 것



크롤링과 스크레이핑

크로링과 스크레이핑

- 크롤링과 스크레이핑
- 웹 페이지의 정보는 '수집 — 분석 — 추출 — 가공 —저장 — 출력'이라는 일련의 흐름



크롤링과 스크레이핑

크롤링과 스크레이이핑 할때의 주의 사항

- 웹 사이트에 접근할 때의 주의 사항
  - 웹사이트의 이용 규약을 확인하고 지킨다
  - robots.txt와 robots 메타 태그의 접근 제한 사항을 지킨다
  - 제한이 없더라도 상대 서버에 부하가 가지 않을 정도의 속도로 접근한다.
  - rel="nofollow"가 설정돼 있으면 크롤러로 접근하지 않는다.
  - 크롤링을 거부하는 조치가가 있으면 즉시 크롤링을 멈추고 이미 추출한 정보를 모두 삭제한다

크롤링과 스크레이핑

크롤링과 스크레이이핑 할때의 주의 사항

- 수집한 데이터를 다룰 때의 주의 사항
  - 수집한 데이터는 저작권을 지켜서 사용하야 함
  - 수집한 데이터는 저작권에 문제가 있으면 개인적인 용도로만 사용함
  - 수집한 데이터를 기반으로 검색 서비스를 제공하는 경우, 웹 사이트와 API등의 사용 규약을 확인하고 문제가 없을 때만 제공함
  - 이용 규약이 따로 없을 때도 상대방에서 확인한 뒤에 데이터를 공개

크롤러 설계 기본

목적과 대상을 명확하게 하기

- 개발 전에 목적을 명확하게 함
- 대상을 충분하게 분석하는 것

```
<ul>
    <li data-url="/category/1/2017-1202.html"><a href="javascript : jump ( ) ;">
    외부 행사 소식</a></ l i >
    <li data-url="/category/2/2017-1201.html"><a href="javascript : jump ( ) ;">
    국제 교류 정보</a></ l i >
</ul>
```

크롤러 설계 기본

URL 확인하기

- 사이트맵을 트리구조(페이지)로 제공하는 사이트
  - 사이트맵을 보면 어떤 정보가 어떤 URL 아래에 있는지 쉽게 확인
  - https://www.seoul.go.kr/helper/siteMap.do
- 사이트맵을 XML로 제공하는 사이트
  - https://www.usa.gov/sitemap.xml
- 사이트맵을 확인할 수 없을 때
  - 카테고리 목록 페이지로 이동하는 링크가 없는지, 사이트 내부에서 하나하나 찾아봄

크롤러 설계 기본

목적 데이터를 따로 제공하는지 확인하기

- 사이트에 따라서는 불특정 다수의 크롤러가 접근해서 부하를 발생시키는 것을 막기 위해 공식 아카이브 데이터를 제공하기도 함
  - https://ko.wikipedia.org/wiki/위키백과:데이터베이스_다운로드
- 아키이브 데이터 덤프를 제공해주지 않는 경우라도 웹 API와 피드를 제공해준다면 이를 활용

크롤러 설계 기본

웹 API

- 웹 API는 특정 URL에 정해진 매개 변수를 넣어 접근하면 XML 또는 JSON등의 구조화된 데이터를 제공해주는 기능
  - https://developers.naver.com/docs/common/openapiguide/apilist.md
  - https://developers.kakao.com/

크롤러가 가지는 설계할 때의 주의 사항

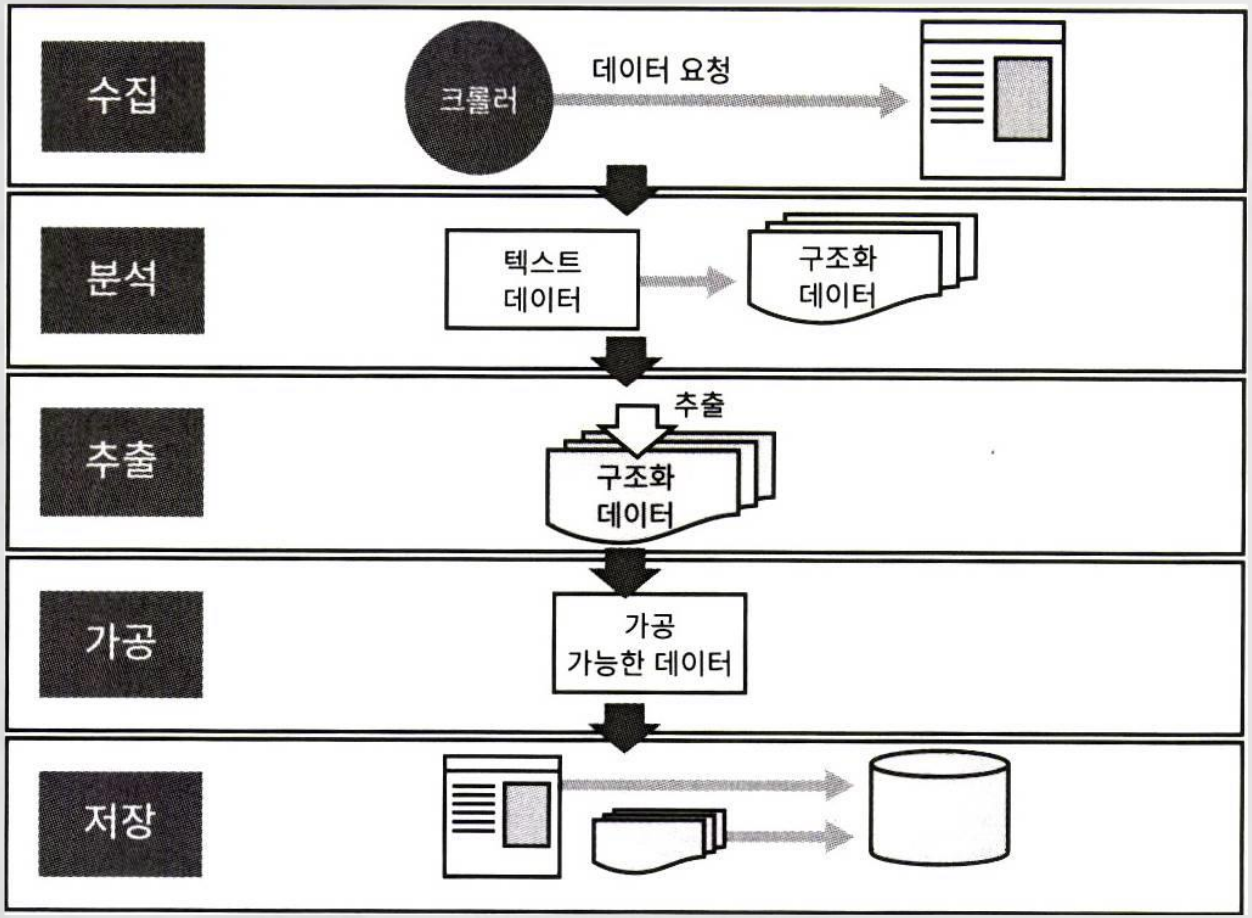
설계가 필요한 부분

- '출력 결과로 무엇이 필요한가'가 바로 '목적'
  - 스프레드시트의 특정 위치에 숫자를 반영
  - 다른 시스템과 연동할 수 있게 API를 제공
  - 자신의 사이트에서 읽어 들일 수 있는게 피드를 만듬
- 출력겨로가이 관련된 명확한 상세가 있어야 함

크롤러가 가지는 설계할 때의 주의 사항

설계가 필요한 부분

- 크롤로의 각 처리 공정



크롤러가 가지는 설계할 때의 주의 사항

네트워크 요청

- 간격 설정하기
  - 적어도 1초에 1번 정도만 요청할 수 있게 하는 것을 권장
- 타임아웃
  - 요청한 사이트로부터 응답이 오지 않는 경우에 타임아웃 설정
  - 3초동안 응답이 없으면 멈춤
- 재시도
  - 큰문제가 없는데도 오류를 응답하는 경우
  - 재시도할때는 어느 정보의 횟수 제한(1~3회 정도)이 있어야 함
  - 재시도 간격도 고려

크롤러가 가지는 설계할 때의 주의 사항

파싱(분석)

- 문자 코드
  - 대부분 UTP-8로 작성되어 있지만 HTML 소스 코드는 다양한 문자코드로 작성된 경우가 많음(EUC-KR 등)
- HTML/XML 파싱
  - 웹 페이지 중에는 태그가 잘못 구성돼 있거나 속성 값에 큰 따옴표가 처져 있지 않은 경우도 많음
- JSON 디코드
  - 대부분의 웹API는 JSON 형식으로 데이터를 응답

크롤러가 가지는 설계할 때의 주의 사항

스크레이핑과 정규 표현식

- URL 정규화
  - 링크를 추출할 때 링크가 상대 경로인 경우
- 테스트
  - 스크레이핑 라이브러리를 사용하거나 정규 표현식을 사용하더라도 한 번에 원하는 데이터 추출하는 경우가 적음
  - 테스트 코드를 사용하면 수집 처리와 스크레이핑 처리를 분리하기 쉬움

크롤러가 가지는 설계할 때의 주의 사항

데이터 저장소의 구조와 선택

- 데이터 저장소
  - 파일
  - 문서 데이터베이스
  - 관계형 데이터베이스
  - 객체 데이터데이스
  - 키-값 데이터베이스

크롤러가 가지는 설계할 때의 주의 사항

배치를 만들 때의 주의점

- 공정분리하기
- 중간 데이터 저장해두기
- 실행 시간 알아 두기
- 중지 조건을 명확하게 하기
- 함수의 매개 변수를 간단하게 하기
- 날짜를 다룰 때의 주의 사항

크롤러가 가지는 설계할 때의 주의 사항

설계(https://wikibook.co.kr/list/)

- 소스 확인하기

```
<html lang="ko-KR">
<!--<![endif]-->
<head>
<title>도서 목록 | 위키북스</title>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width" />
<meta property="twitter:account_id" content="4503599629654224" />
<!--[if lt IE 9]>
```

```
<li class="row unstyled book-list-item" target="https://wikibook.co.kr/python-crawler/">
<div class="col-md-1 book-list-image">
<a href="https://wikibook.co.kr/python-crawler/"><img src="https://wikibook.co.kr/images/cover/s/9791158391645.jpg"></a>
</div>
<div class="col-md-11 book-list-detail">
<a class="book-url" href="https://wikibook.co.kr/python-crawler/"><h4 class="main-title">파이썬을 활용한 크롤러 개발과 스크레이핑 입문</h4></a>
<div class="sub-title">크롤러 설계와 개발부터 수집 데이터 분석과 운용까지</div>
<div class="book-info">
<span class="author">카토 카츠야, 요코야마 유우키 <small>지음</small></span> |
<span class="translator">윤인성 <small>옮김</small></span> |
30,000원 |
<span class="pub-date">2019년 07월 24일 | </span>
<span class="isbn"><small>ISBN: </small>9791158391645</span>
<span class="tag" style="display:none">웹,크롤링,스크레이핑,웹 크롤링,크롤러,웹 수집</span>
</div>
<span id="tags">
<i class="fas fa-tags"></i>
<a href="https://wikibook.co.kr/tag/%ec%9b%b9/"><span class="label label-default">웹</span></a>
<a href="https://wikibook.co.kr/tag/%ed%81%ac%eb%a1%a4%eb%a7%81/"><span class="label label-default">크롤링</span></a>
<a href="https://wikibook.co.kr/tag/%ec%8a%a4%ed%81%ac%eb%a0%88%ec%9d%b4%ed%95%91/"><span class="label label-default">스크레이핑</span></a>
<a href="https://wikibook.co.kr/tag/%ec%9b%b9-%ed%81%ac%eb%a1%a4%eb%a7%81/"><span class="label label-default">웹 크롤링</span></a>
<a href="https://wikibook.co.kr/tag/%ed%81%ac%eb%a1%a4%eb%9f%ac/"><span class="label label-default">크롤러</span></a>
<a href="https://wikibook.co.kr/tag/%ec%9b%b9-%ec%88%98%ec%a7%91/"><span class="label label-default">웹 수집</span></a>
</span>
</div>
</li>
```

크롤러가 가지는 설계할 때의 주의 사항

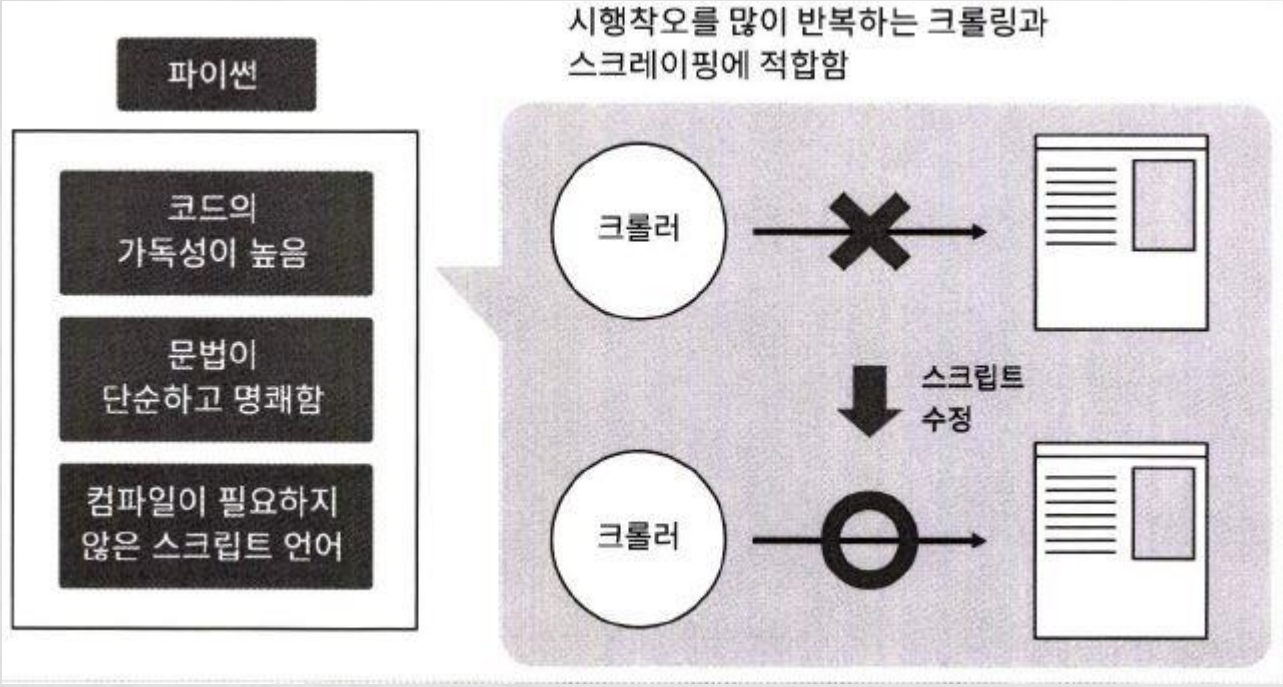
설계(https://wikibook.co.kr/list/)

- 저장방법
- 파일저장 형식
  - CSV(Comma-Separated Values)
  - TSV(Tab-Separated Values)
  - JSON(JavaScript Object Notation)

파이썬을 사용하는 이유

파이썬 언어의 특징

- 파이썬이 크롤링과 스크레이핑에 적합한 이유



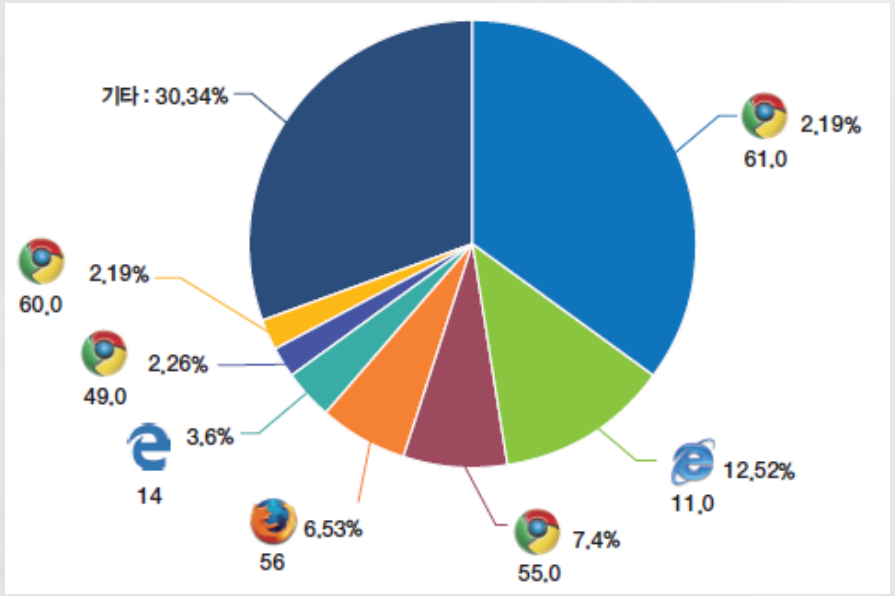
- 풍부한 라이브러리 https://pypi.org/

파이썬의 기초와 HTML5/CSS

HTML

HTML과 웹 브라우저

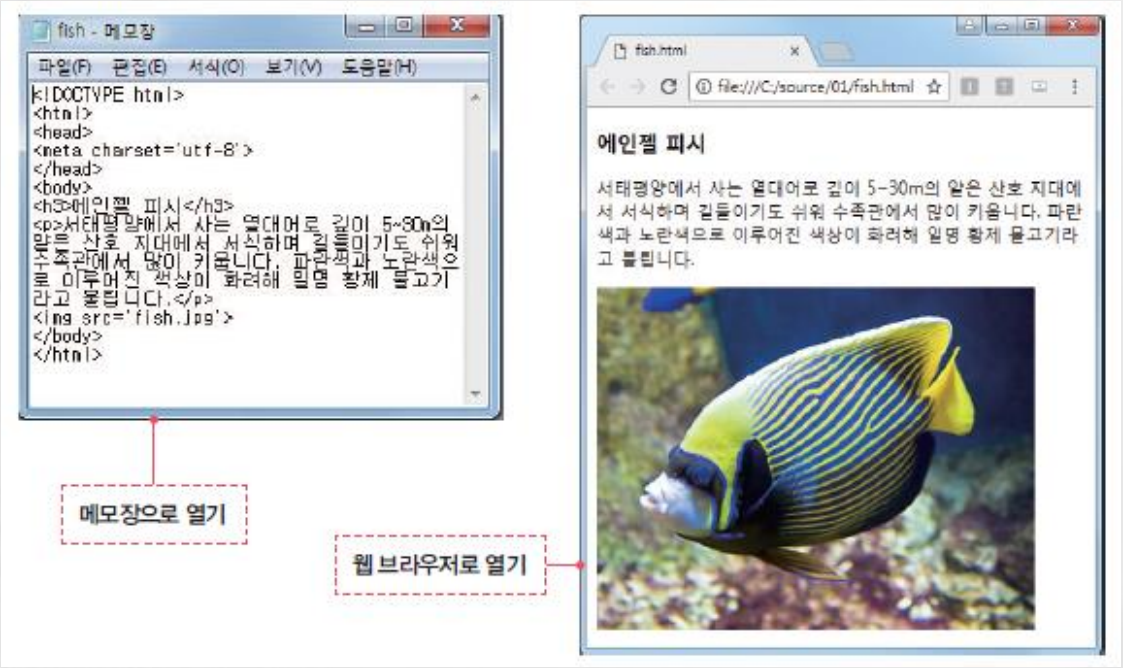
- HTML (HyperText Markup Language)
  - 웹 페이지 제작에 가장 기본적으로 사용되는 마크업 언어
  - HTML5
  - .html 또는 .htm 확장자
- 웹 브라우저
  - 인터넷 익스플로러, 크롬, 파이어폭스 등
  - 웹 페이지 접속 위해 사용



HTML

HTML 태그

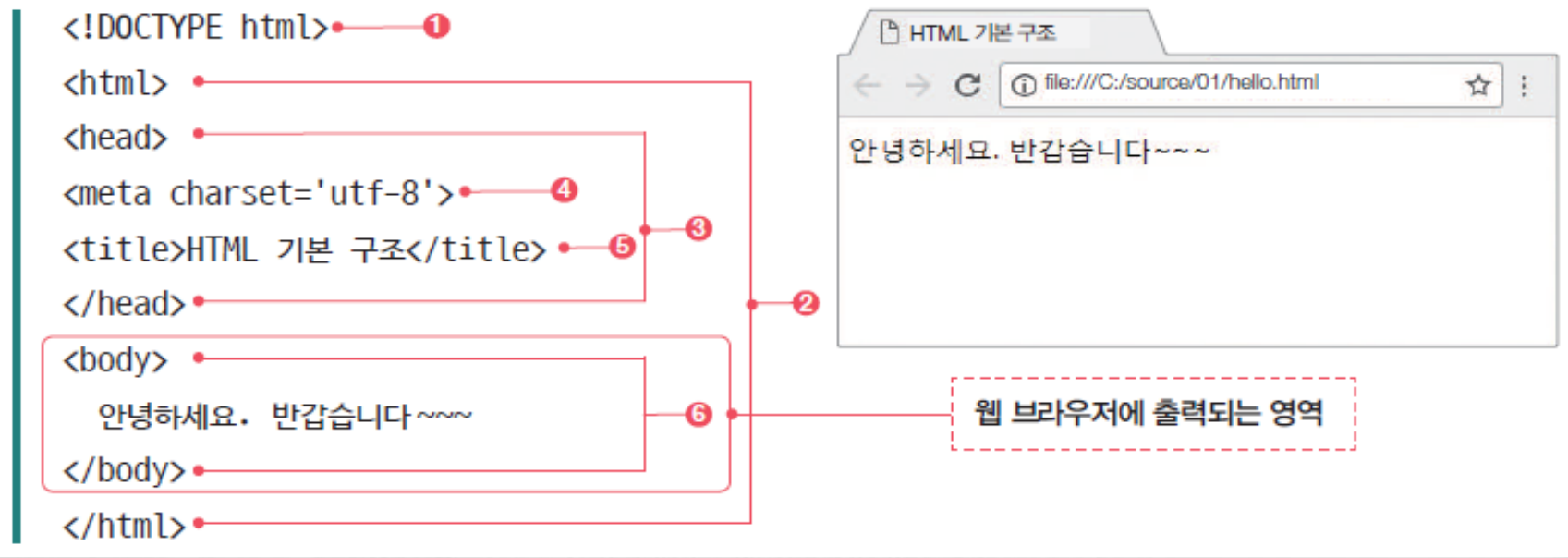
- HTML에서 사용되는 명령어
- 홑화살괄호(< >)로 표현
  - <h3> </h3> : 글 제목 만들기
  -   : 단락 만들기



HTML

HTML 문서 기본 구조

- <!D0CTYPE html>
- <html> </html>
- <head> </head>
- <meta>
- <title> </title>
- <body> </body>



HTML

HTML 태그

- 글 제목 만들기 : <h1>, <h2>, <h3>, <h4>, <h5>, <h6>
  - 위 태그 중에서 선택하여 글 제목의 크기 지정
- 단락 나누기 : <p>
- 글자 두껍게 하기 : <b>
- 줄바꿈과 공백 삽입 :
,
- 목록 만들기
  - 구성상 항목의 순서가 내용과 무관한 경우 : <ul>,
  - 항목의 순서가 중요한 경우 : <ol>,
- HTML 문서에 설명글 달기
  - 두 태그 사이 부분을 웹 브라우저가 해석하지 않도록 함 : <!-- , -->

HTML

HTML 멀티미디어와 링크

- HTML 문서에 이미지 넣기 : <img>
  - src 속성과 함께 사용 : src='dog.jpg'

이미지 파일 포맷	파일 확장자	주로 사용되는 곳
JPG	.jpg	사진 이미지
GIF	.gif	아이콘과 같은 컴퓨터 그래픽 이미지 간단한 애니메이션 이미지
PNG	.png	사진과 그래픽 이미지에 모두 사용 가능
SVG	.svg	컴퓨터 그래픽 이미지와 로고 이미지

- 이미지의 가로 및 세로 크기 조정 : width 및 height 속성
- 이미지 위 마우스 올릴 시 표시할 메시지 지정 : title 속성

속성	의미
src	삽입되는 이미지 파일의 이름과 경로
width	이미지의 너비 지정
height	이미지의 높이 지정
title	이미지 위에 마우스를 올렸을 때 나타나는 이미지 설명 글

HTML 멀티미어와 링크

- 오디오를 재생 : <audio>
  - controls : 오디오 플레이어를 화면에 표시
  - autoplay : 자동 재생
  - loop : 자동 반복

오디오 재생하기 03\audio.html

```
<audio ❶src='bass.mp3' ❷controls ❸autoplay ❹loop></audio>
```

- 비디오 파일 삽입 : <video>

비디오 재생하기 03\video.html

```
<video ❶width='320' ❷height='240' ❸autoplay ❹controls ❺loop>
  <source src='sample-video.mp4' type='video/mp4'>
  <source src='sample-video.webm' type='video/webm'>
  <source src='sample-video.ogg' type='video/ogg'>
</video>
```

HTML 멀티미어와 링크

- 링크 : 글자나 이미지에 다른 웹 페이지를 연결하는 것
  - 링크 거는 역할 : <a>
  - 클릭 시 이동할 주소 : href
- 새로운 탭으로 링크 걸기 : target
  - 속성값으로 _blank 지정
  - 웹 브라우저의 새로운 탭에 해당 웹 페이지 표시

새로운 탭으로 링크 걸기 03\new-tab.html

```
<h2>새로운 탭으로 링크 걸기</h2>
❶<h3><a href='http://www.facebook.com' target='_blank'>페이스북</a></h3>
❷<h3><a href='http://www.twitter.com' target='_blank'>트위터</a></h3>
❸<h3><a href='http://www.pinterest.com' target='_blank'>핀터레스트</a></h3>
```

HTML

테이블 삽입하기

- 웹 페이지에서 테이블 제작 시 사용하는 태그
  - 표에 넣고자 하는 내용 : <table>
  - 하나의 행 : <tr>
  - 테이블 각 열의 제목 “table header” : <th>
  - 열 제목 나타내는 첫 번째 행의 셀 제외한 각각의 셀 표현 : <td>

- 테이블의 행과 열 합치기
  - 행 합치는 데 사용, 합치고자 하는 셀의 개수를 속성값으로 지정 : rowspan
  - 열을 합치는 데 사용 : colspan

HTML

텍스트와 입력 창 만들기

- 텍스트 입력 창
  - 사용자가 텍스트 입력하는 폼 양식
  - <form> 으로 폼 양식 삽입
- 라디오 버튼
  - 원형의 선택 폼 양식
  - 여러 항목 중 단 하나만 선택
- 체크 박스
  - 사각형의 선택 폼 양식
  - 다수 항목 선택 가능

라디오 버튼과 체크 박스 04\radio-checkbox.html

```
<form>
정보공개 : ❶<input type='radio' checked> 공개
          ❷<input type='radio'> 비공개 <br>
취       미 : ❸<input type='checkbox'> 탁구
          ❹<input type='checkbox'> 배드민턴
          ❺<input type='checkbox' checked> 음악감상
          ❻<input type='checkbox' checked > 악기연주
</form>
```

HTML

파일 선택 창 만들기

- <input>의 type 속성값을 'file'로 지정
- 선택 박스
  - 선택 박스 생성 : <select>
  - 선택 박스에 들어갈 항목 : <option>
- 체크 박스
  - 사각형의 선택 폼 양식
  - 다수 항목 선택 가능

CSS(Cascading Style Sheets)

CSS(Cascading Style Sheets)란?

- HTML 보조하여 웹 페이지 글자, 이미지 등 요소 꾸미고 배치

CSS를 이용하여 글자 크기와 색상 바꾸기 — 01\hello-css.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset='utf-8'>
<title>HTML 기본 구조</title>
<style>
body {
  font-size: 20px;
  color: red;
}
</style>
</head>
<body>
  안녕하세요. 반갑습니다~~~
</body>
</html>
```

파일의 위치이며 C:\source\챕터\파일 이름입니다. C:\source는 생략하고 늘 챕터\파일 이름만 적혀있습니다.

CSS에요!

CSS(Cascading Style Sheets)

CSS(Cascading Style Sheets)란?

- <head> 태그 내 <style> 태그
  - HTML 문서에 CSS 삽입
- <style> 태그 내 h3
  - CSS 선택자
  - HTML 문서에서 꾸미고자 하는 영역 선택
- color; red;
  - CSS 명령
  - 선택자 h3가 선택한 글자의 색상 적색으로 변경

CSS(Cascading Style Sheets)

글자 스타일 지정하기

- CSS 이용하여 글자의 색상, 크기, 글꼴 지정 등

글자 스타일 지정하기 05\text-style.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset='utf-8'>
<style>
❶h2 {
    color: blue;
    text-shadow: 2px 2px 10px gray;
  }
❷p {
    color: #444444;
    font-size: 18px;
    font-family: '바탕';
    line-height: 150%;
  }
❸span {
    font-weight: bold;
    color: #0e9bdc;
    text-decoration: underline;
  }
</style>
</head>
```

<h2> </h2>로 둘러싸인 콘텐츠를 꾸며주는 부분입니다.

 로 둘러싸인 콘텐츠를 꾸며주는 부분입니다.

 으로 둘러싸인 콘텐츠를 꾸며주는 부분입니다.

CSS(Cascading Style Sheets)

글자 스타일 지정하기

- 글 제목의 글자 색상 지정과 그림자 넣기
  - text-shadow : 글자에 그림자 넣는 속성
  - 2px 2px 10px gray : 그림자 색상과 형태를 지정
- 단락의 글자 스타일 지정
  - #444444 : 색상 코드 (짙은 회색) color 속성값으로 색상 영문명 및 색상 코드 사용 가능
- 특정 영역의 글자에 스타일 지정
  - 특정 부분 글자를 CSS로 꾸미기 위해 영역 지정 :

CSS(Cascading Style Sheets)

목록 스타일 지정하기

- list-style-type
  - 각 항목에 붙는 글머리 형태 지정
  - square 속성값 - 각 항목 앞에 정사각형 포인트
- CSS 설명 글
  - /* 에서 시작하여 */로 종료
- HTML 설명 글
  - <!-- 에서 시작하여 --> 로 종료

CSS(Cascading Style Sheets)

CSS 선택자란?

- 태그 선택자
  - 태그의 영역 선택하고 이후에 오는 CSS 명령을 해당 영역에 적용 : p
- id 선택자
  - 웹 페이지에서 유일무이한 단 하나의 특정 영역 지정하여 CSS 명령 적용
  - id명 앞에 샵(#) 붙여야
- 클래스 선택자
  - 두 군데 이상의 특정 영역 지정하여 동일한 CSS 적용
  - 클래스명 앞에 점(.) 붙여야

CSS(Cascading Style Sheets)

태그 선택자

- 선택자에 태그명 사용하는 것
- 웹 페이지에서 태그 사용된 영역 선택, 해당 영역에 css 명령 적용

태그 선택자 — 06\tag-selector.html

style

```
❶body {
    font-family: '돋움';
  }
❷h3 {
    font-family: '맑은고딕';
    color: blue;
  }
❸p {
    font-size: 14px;
    line-height: 150%;
  }
❹li {
    list-style-type: square;
    font-size: 16px;
  }
❺span {
    font-weight: bold;
  }
```

body

```
<h3> - 배낭여행이란?</h3>
<p>여권, 항공권 등 여행 시 필요한 것만을 준비하고 현지에서
숙박, 식사 등을 해결하는 자유여행을 말합니다.</p>

<h3>- 배낭여행의 종류</h3>
<p>배낭여행에는 모든 일정을 자신이 정하는 자유 배낭, 여
러 명이 같이 출발 전 숙소와 교통편 등을 미리 예약하고 여행
하는 단체 배낭, 자유 배낭과 단체 배낭의 중간 형태인 패키지
배낭여행 등이 있습니다.</p>

<h3>- 배낭여행 준비</h3>
<ul>
  <li><span>여권 준비</span> : 여권이 없으면 신청하고
  여권 유효기간을 반드시 체크.</li>
  <li><span>비행기 예약</span> : 항공사의 예매 사이트
  나 예약 대행 사이트 이용.</li>
  <li><span>여행 스케줄</span> : 스케줄은 가능한 세부
  적으로 잘 짜야 함.</li>
  <li><span>짐싸기</span> : 꼭 필요한 물품만으로 최대
  한 간단하게 짐 준비.</li>
</ul>
```

CSS(Cascading Style Sheets)

태그 선택자

- body
  - <body> 태그 영역인 전체 웹 페이지 선택
  - font-family: '돋움' : 전체 웹 페이지 기본 글꼴을 '돋움'으로
- p
  - <p> 태그 영역 선택
  - font-size: 14px; : <p> 태그 영역 두 단락의 글자 크기를 14픽셀로

CSS(Cascading Style Sheets)

id 선택자

- 웹 페이지에서 하나만 존재하는 유일한 특정 영역 선택

특정 단락에 id 선택자 사용 — 06\id-selector.html

style

```
p {
  color: gray;
}
❸#p1 {
  color: blue;
}
❹#p2 {
  color: green;
}
```

body

```
<p ❶id='p1'>안녕하세요.</p>
<p ❷id='p2'>반갑습니다.</p>
<p>또 만났군요.</p>
<p>자주 만나네요!</p>
```

CSS(Cascading Style Sheets)

박스 모델

- 박스 모델(Box Model)의 구성요소
  - 박스 형태로 된 모든 HTML 요소
  - 경계선(border) 그리고 마진(margin)과 패딩(padding) 지정 가능
- border
  - 예시의 청색 경계선 등 그리는 데 사용하는 속성
- padding
  - 경계선 내부 간격
  - 예시의 콘텐츠 '웹이란?'과 경계선 사이의 간격
- margin
  - 경계선 외부 간격
  - 경계선과 외부의 요소 사이의 간격

CSS(Cascading Style Sheets)

경계선 그리기

- border
  - 경계선 스타일
    - solid (실선)
    - double (이중실선)
    - dotted (점선)
    - dashed (줄 선)
  - 경계선 두께
    - px 단위
  - 경계선 색상
    - 색상 이름 혹은 코드

```
border: 경계선 스타일 경계선 두께 경계선 색상;
```

- padding
  - 글자와 경계선 사이의 간격
- width / height
  - 박스의 너비 / 높이

CSS(Cascading Style Sheets)

패딩과 마진 설정하기

- padding 속성값을 상-우-하-좌단 순서로 적용

- *
  - 전체 선택자
  - 모든 HTML 태그 요소를 선택
- padding: 0; margin: 0;
  - 전체 선택자에 의해 선택된 모든 요소에서 패딩과 마진을 0으로 초기화

CSS(Cascading Style Sheets)

배경 색상 설정하기

- background-color: yellow;
  - 태그 선택자 body로 전체 웹 페이지 선택
  - background-color 속성
    - yellow 속성
- 자세히 보기 >
  -
    - 박스 형태 요소 만들기
- #button
  - id='button'의 영역을 선택

CSS(Cascading Style Sheets)

배경 이미지 삽입하기

- background-image: url('img/bg.jpg');
  - background-image 속성
    - 배경 이미지 삽입에 사용
    - url 뒤 괄호 안에 경로 포함한 배경 이미지 파일 이름 입력
- background-repeat: no-repeat;
  - no-repeat 속성값

CSS(Cascading Style Sheets)

테이블 경계선 그리기

- border: solid 1px #000000;
  - 태그 선택자 table, th, td
  - border 속성 이용하여 실선, 1픽셀 두께, 흑색 경계선 그림
- border-collapse: collapse;
  - border-collapse 속성
  - collapse 속성값
  - 테이블 경계선을 하나의 가는 실선으로 그림
    - 생략할 경우 이중실선

CSS(Cascading Style Sheets)

테이블 너비 지정과 텍스트 정렬하기

- width: 80px;
  - width 속성, 80px 속성값으로 너비 80픽셀로 확장
- text-align: center;
  - text-align 속성, center 속성값으로 테이블 셀 안 요소를 중앙정렬
- background-color: #adf0f4;
  - 셀의 배경 색상을 색상코드 #adf0f4 색상으로 지정

CSS(Cascading Style Sheets)

레이아웃

- 웹 페이지에 박스, 텍스트, 이미지 등 HTML 요소 배치하는 것
- 수평 방향 레이아웃 / 인라인 (inline)
- 수직 방향 레이아웃 / 블록 (block)
- 인라인 요소
  - 붉은색 표시 이미지, 텍스트 : 가로 방향으로 배치되는 HTML 요소
  - <img>, 태그 등
- 블록 요소
  - 파란색 표시 박스 : 세로 방향으로 배치되는 HTML 요소
  - <p>, 태그 등

CSS(Cascading Style Sheets)

display 속성

- 태그
- list-style-type: none;
  - 목록의 글머리 기호 제거
- #v_menu li / #h_menu li
  - 후손 선택자
  - 선택자 아래 다시 선택자 설정
- display: inline;
  - 태그가 기본으로 가지는 블록을 인라인으로 변경

CSS(Cascading Style Sheets)

float과 clear 속성

- float 속성
  - float: left; 해당 요소를 왼쪽에 배치
  - float: right; 해당 요소를 오른쪽에 배치
- clear 속성
  - clear: both;
  - 앞의 float: left; 와 float: right; 에서 사용된 float 속성 해제

크롤링과 스크레이핑

웹 브라우저 실행

웹브라우저 실행시키기(webbrowser)

- webbrower는 자신의 시스템에서 사용하는 기본 웹브라우저가 자동으로 실행되게 하는 모듈
- 웹브라우저를 자동으로 실행시켜고 해당 URL인 www.naver.com로 감

```
import webbrowser

webbrowser.open("http://www.naver.com")
```

웹 브라우저 실행

웹브라우저 실행시키기(webbrowser)

- webbrower의 open함수는 웹브라우저가 실행된 상태이면 해당 주소로 이동
- 웹브라우저가 실행되지 않은 상태이면 새로이 웹브라우저가 실행되어 해당 주소로 이동
- Open_new 함수는 이미 웹브라우저가 실행된 상태에서 새로운 창으로 해당 주소가 열리도록 함

```
import webbrowser

webbrowser.open_new("http://google.co.kr")
```

웹 페이지 추출하기

urllib 으로 웹 페이지 추출하기

- 페이지를 추출할 때는 표준 라이브러리 urilib.request 모듈을 사용
- urllib.request에 포함돼 있는 urlopen() 함수에 URL을 지정하면 웹 페이지를 추출

```
from urllib.request import urlopen
# urlopen()함수는 HTTPResponse 자료형의 객체를 반환합니다.
f = urlopen('http://hanbit.co.kr')
type(f)
f.read() # read() 메서드로 HTTP 응답 본문(bytes 자료형)을 추출합니다
f.status    # 상태 코드를 추출합니다.
f.getheader('Content-Type') # HTTP 헤더의 값을 추출합니다
```

웹 페이지 추출하기

meta 태그에서 인코딩 방식 추출하기

- HTML 내부의 meta 태그 또는 응답 본문의 바이트열도 확인해서 최종적 인

  인코 딩 방식을 결정하고 화면에 출력

```
import re
import sys
from urllib.request import urlopen
f = urlopen('http://www.hanbit.co.kr/store/books/full_book_list.html')
bytes_content = f.read()
scanned_text = bytes_content[:1024].decode('ascii', errors='replace')
match = re.search(r'charset=["₩']?([₩w-]+)', scanned_text)
if match:
    encoding = match.group(1)
else:
encoding = 'utf-8'
print('encoding:', encoding, file=sys.stderr)
text = bytes_content.decode(encoding)
print(text)
```

웹 페이지에서 데이터 추출하기

정규 표현식으로 스크레이핑하기

- 표준 라이브러리의 re 모듈을 사용

```
import re
from html import unescape
# 이전 절에서 다운로드한 파일을 열고 html이라는 변수에 저장합니다.
with open('dp.html') as f:
    html = f.read()
# re.findall()을 사용해 도서 하나에 해당하는 HTML을 추출합니다.
for partial_html in re.findall(r'<td class="left"><a.*?</td>', html, re.DOTALL):
    # 도서의 URL을 추출합니다.
    url = re.search(r'<a href="(.*?)">', partial_html).group(1)
    url = 'http://www.hanbit.co.kr' + url
    # 태그를 제거해서 도서의 제목을 추출합니다.
    title = re.sub(r'<.*?>', '', partial_html)
    title = unescape(title)
    print('url:', url)
    print('title:', title)
    print('---')
```

웹 페이지에서 데이터 추출하기

XML(RSS) 스크레이핑

- 블로그 또는 뉴스 사이트 등의 웹사이트는 변경 정보 등을 RSS라는 이름의 XML 형식으로 제공
- RSS는 XML을 기반으로 만들어졌으므로 HTML보다 간단하게 파싱
- rss라는 이름의 요소를 루트로 하는 트리 구조를 가지고 있음
- 내부에는 피드를 나타내는 channel 요소가 있음
- channel 요소의 앞부분에는 피드의 메타 정보를 나타내는 title 요소와 link 요소 등이 있음

데이터 저장하기

CSV 형식으로 저장하기

- CSV(Common Seperated Values)는 하나의 레코드를 한 줄에 나타내고, 각 줄의 값을 쉼표로 구분하 는 텍스트 형식
- 행과 열로 구성되는 2차원 데이터를 저장할 때 사용
- CSV 형식을 만드 는 가장 쉬운 방법은 str.join() 메서드를 사용
- csv.writer를 사용하면 간단하게 CSV 형식으로 출력
- 한 줄을 줄력할 때 는 writerow()메서드를 사용하며 , 매개변수로 list 또는 tuple과 같은 반복 가능한 객체

데이터 저장하기

JSON 형식으로 저장하기

- JSON(JavaScript Object Notation)은 자바스크립트에서 객체를 표현하는 방법을 사용하는 텍스트형식
- JSON을 사용하면 list 또는 dict를 조합 한 복잡한 데이터 구조를 쉽게 다룸
- 파이썬은 JSON 형식을 쉽게 다룰 수 있게 json 모듈을 제공
- json.dumps()함수 를 사용하면 list와 dict 등의 객체를 JSON 형식 문자열로 변환

데이터 저장하기

데이터베이스 (SQLite3) 에 저장하기

- SQLite3는 파일기반의 간단한 관계형 데이터베이스
- 구문을 사용해 데이터를 읽고 쓸 수 있음
- SQLite는 가볍게 사용할 수 있는 관계형 데이터베이스지만 파일을 쓰는 데 시간이 꽤 걸린다는 것이 단점
- 적은 데이터를 다룰 때는 문제 없지만 크롤링한 대량의 데이터를 계속해서 올리면 SQLite를 사용할 경우 파일을 쓰는 부분이 병목지점으로 작용
- 어떤 프로그램 이 파일을 열고 내용을 쓰고 있을 때는 다른 프로그램에서 해당 파일을 사용할 수 없으므로 동시 처리도 불가능

파이썬으로 스크레이핑하는 흐름

파이썬으로 스크레이핑하는 흐름

- fetch(url)
  - 매개변수로 url을 받고 지정한 URL의 웹 페이지를 추출
- scrape(html)
  - 매개변수로 html을 받고 정규 표현식을 사용해 HTML에서 도서 정보를 추출
- save(db_path, books)
  - 매개변수로 books라는 도서 목록을 받고, SQLite 데이터베이스에 저장

urllib 사용법 및 기본 스크래핑

urllib.request 기초 사용법

- 네이버 이미지 다운로드 대상
- 구글 HTML 정보 다운로드 대상
- Header 정보 확인
- 다운로드 정보 파일 저장

urllib 사용법 및 기본 스크래핑

urllib.request 예외 처리

- 기존 소스 코드 변경
- 예외 처리 추가
- 기타 리팩토링

lxml.html 사용

네이버 뉴스 스탠드 스크랩핑(1)

- 네이버 메인 뉴스 정보 스크랩핑
- 신문사 정보 리스트 출력
- CSS 선택자 활용

lxml.html 사용

네이버 뉴스 스탠드 스크랩핑(2)

- 네이버 메인 뉴스 정보 스크랩핑
- 신문사 정보 딕셔너리 출력
- Session 사용
- Xpath 활용

Get 방식 데이터 통신

urlopen 함수의 다양한 함수

- 사이트 요청 정보 확인
- encar(엔카)사이트 정보 수신
- Get 파라미터 요청
- 수신 데이터 디코딩(Decoding)
- 요청 URL 정보 분석

Get 방식 데이터 통신

RSS 데이터 스크랩핑

- 행정안전부 사이트 RSS 데이터 수신
- RSS란?
- 반복문을 활용한 연속 요청
- 요청 URL 정보 분석
- 수신 XML 데이터 확인

requsts 사용 스크랩핑

Session 및 Cookie 사용

- Requests 요청 정보 Payload
- 세션 활용성화 및 비활성화
- 쿠키 정보 전송
- User-Agent 정보 전송
- 수신 상태 코드 확인

requsts 사용 스크랩핑

JSON 수신 데이터 처리

- Httpbin 사이트를 이용한 JSON
- 수신 데이터 처리
- 수신데이터 -> JSON 변환 출력
- Response 다양한 정보 출력

requsts 사용 스크랩핑

Rest API

- 개발자 도구 송수신 분석
- Rest API 란?
- POST, PUT
- DELETE
- Requests

BeautifulSoup 사용 스크랩핑

Beautiful Soup 사용법

- Beautiful Soup Selector
- HTML 태크 선택자 이해
- FIND, FIND_ALL
- SELECT, SELECT_ONE
- 다양한 DOM 접근 방법

BeautifulSoup 사용 스크랩핑

네이버 이미지 다운로드

- Beautiful Soup 이미지 다운로드
- Naver 이미지 검색 송수신 분석
- Select, Find_all
- 이미지 변환 및 저장
- 예외 처리

BeautifulSoup 사용 스크랩핑

로그인 처리

- Session 사용 로그인, 데이터 수집
- 대상 사이트 로그인 과정 분석
- 로그인 후 페이지 이동
- 필요 데이터 추출

Selenium 사용

Selenium – 웹 자동화

- Selenium 설명 및 기본 설정
- Driver 설치
- 웹 자동화의 이해
- Selenium 기초
- 다음 사이트 기반

Selenium 사용

웹 크롤링

- 데이터 수집 프로젝트
- 대상 사이트 선정 및 분석
- Explicitly wait
- Implicitly wait
- 필요 정보 추출

Selenium 사용

웹 크롤링

- 데이터 수집 프로젝트
- 페이지 전환 추가
- Selenium 성능 개선
- 전체 프로세스 확인

Selenium 사용

웹 크롤링

- 데이터 수집 프로젝트
- 이미지 수집
- 엑셀 데이터 작성
- 전체 프로젝트 소스 코드 리뷰
- 기능 개선 및 공부 내용 추천

정리

정리

- 크롤링과 스크레이핑
- HTML과 CSS
- BeautifulSoup
- selenium