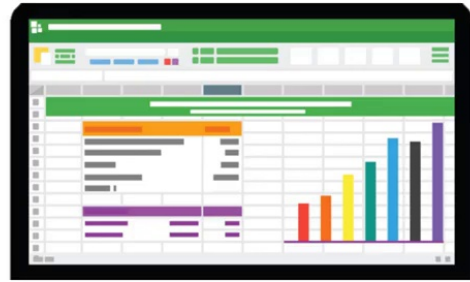




## 스프레드시트(SpreadSheet)

<http://opendatahandbook.org>

- 많은 정부 기관이 마이크로소프트 엑셀과 같은 스프레드시트로 정보 관리
- 스프레드시트는 서로 다른 컬럼의 의미를 올바르게 기술하기 위해 사용 가능
- 스프레드시트 안에 다루기 어려운 매크로와 공식이 있기 때문에, 사용자가 읽기 쉽도록 해당 스프레드시트와 함께 계산 공식을 제공하는 것이 바람직



## CSV(comma-separated values)

<http://opendatahandbook.org>

- CSV 파일은 간결하고 동일한 구조로 대용량 데이터 집합을 전송하기 위해 적합하기 때문에 매우 유용한 형식
- CSV로 표현된 데이터는 별도의 문서가 없을 경우 서로 다른 컬럼의 중요성을 추측하기 어려움 (유용성이 떨어질 수 있음)
- 콤마로 구분된 형식에 따라 개별 필드에 대한 정확한 문서화를 만드는 것이 중요
- 파일 구조가 침해받지 않아야 됨
- CSV 파일에서 하나의 필드가 삭제되면 나머지 데이터를 해석하기 위한 방법이 없기 때문에 파일 수정 없이는 남은 데이터의 해석에 문제가 있음



## HTML(HyperText Markup Language)

<http://opendatahandbook.org>

- 웹 페이지를 위한 마크업 언어
- 제목, 단락, 목록 등과 같은 본문을 위한 구조적 의미를 나타내는 것뿐만 아니라 링크, 인용과 그 밖의 항목으로 구조적 문서를 만들 수 있는 방법 제공
- 이미지와 객체를 내장하고 대화형 양식을 생성하는 데 사용
- HTML은 웹 페이지 콘텐츠 안의 <> 괄호에 둘러싸인 "태그"로 되어있는 HTML 요소 형태로 작성
- HTML은 웹 브라우저와 같은 HTML 처리 장치의 행동에 영향을 주는 자바스크립트와 본문과 그 밖의 항목의 외관과 배치를 정의하는 CSS 같은 스크립트를 포함하거나 불러올 수 있음



## XML(Extensible Markup Language)

<https://ko.wikipedia.org/wiki/XML>

- W3C(World Wide Web Consortium)에서 개발
- 다른 특수한 목적을 갖는 마크업 언어를 만드는데 사용하도록 권장하는 다목적 마크업 언어
- XML은 SGML의 단순화된 부분집합
- 다른 많은 종류의 데이터를 기술하는 데 사용할 수 있음
- XML은 주로 다른 종류의 시스템, 특히 인터넷에 연결된 시스템끼리 데이터를 쉽게 주고 받을 수 있게 하여 HTML의 한계를 극복할 목적으로 만들어짐
- 데이터 상호교환을 위해 폭넓게 사용되는 포맷
- 데이터 구조를 유지하는데 유리하며 개발자는 파일을 읽는 간섭없이 데이터에 문서의 일부 작성 가능



## JSON(JavaScript Object Notation)

<https://ko.wikipedia.org/wiki/JSON>

- 속성-값 쌍 또는 키-값 쌍으로 이루어진 데이터 오브젝트를 전달하기 위해 인간이 읽을 수 있는 텍스트를 사용하는 개방형 표준 포맷
- 비동기 브라우저/서버 통신(AJAX)을 위해, 넓게는 XML(AJAX 사용)을 대체하는 주요 데이터 포맷
- 인터넷에서 자료를 주고 받을 때 그 자료를 표현하는 방법으로 알려져 있음
- 자료의 종류에 큰 제한은 없으며, 특히 컴퓨터 프로그램의 변수값을 표현하는 데 적합
- 어떤 프로그래밍 언어로도 읽을 수 있는 단순한 파일 포맷
- XML 같은 포맷에 비해 컴퓨터가 처리하기 쉬움



## RDF(Resource Description Framework)

<http://opendatahandbook.org>

- W3C가 권고하는 포맷
- 다수의 자료원으로부터 데이터를 쉽게 조합할 수 있는 형식으로 데이터 표현 가능
- RDF 데이터는 여러 가지 직렬화 방법이 있으며, XML과 JSON으로 저장 가능
- RDF는 웹에 있는 오픈 데이터를 서로 연결하는데 편리한 방법을 제공(식별자로 URL 사용용을 장려)



## 문서(Documents)

<http://opendatahandbook.org>

- Word, ODF, OOXML, PDF와 같은 포맷의 표준적인 문서로 일정한 종류의 데이터를 보여주는데 활용
- 이 포맷은 일관적으로 구조를 유지하지 않으며, 자동적으로 데이터를 입력하기 어려움
- 데이터를 재사용할 수 있도록 문서 기준의 템플릿을 사용한다면, 최소한 정보를 문서 밖으로 추출 가능
- 가능한 많은 활자 마크업을 이용하여 기계가 콘텐츠로부터 표제를 구분하기 쉽게 되어있는 것은 데이터의 추가 사용을 이끌어낼 수 있음
- 일반적으로 다른 포맷으로 데이터가 있다면, 워드프로세스 포맷으로 표현하는 것을 추천하지 않음





## 일반 텍스트(Plain Text)

<http://opendatahandbook.org>

- 일반 텍스트 문서 (.txt) 는 컴퓨터가 읽기 매우 쉬움
- 문서 내에 구조화된 메타데이터를 갖지 않음
- 개발자는 개별 문서를 처리할 수 있는 파서를 개발할 필요가 있음
- 운영체제 사이에서 일반 텍스트 문서를 전환할 때 문제가 생길 수 있음  
(MS 윈도우, Mac OSX, Unix 시스템은 라인 끝을 처리하는 서로 다른 방식을 갖고 있음)



## 스캔 이미지(Scanned Image)

<http://opendatahandbook.org>

- 아마도 대부분의 데이터에서 가장 적합한 형식은 문서의 전체 텍스트와 문서의 이미지를 함께 표시하는 것
- TIFF와 JPEG-2000 포맷은 전자적으로 만들어지지 않은 데이터를 이미지로 표현하는데 적합
- 오래된 기록과 사진은 자료가 없는 것보다 나음



## 전용 형식

<http://opendatahandbook.org>

- 전용 시스템은 데이터를 저장하고 추출할 수 있는 독립적인 데이터 포맷을 갖고 있음
- 이와 같은 포맷으로 데이터를 노출시키는 경우가 종종 있음
- 데이터를 유사한 시스템에서 사용할 것으로 기대된다면, 전용 포맷에서만 찾을 수 있는 정보는 항상 표시해야 함

