

## II. 디지털 텍스트

디지털 텍스트란 지식을 전달하는 문자 언어를 디지털 미디어에 전자적인 신호로 기록한 것을 말한다. 워드프로세서로 작성하여 컴퓨터 속에 저장한 한편의 글도 넓은 의미의 디지털 텍스트라고 할 수 있다. 워드프로세서로 만든 문서는 종이에 출력하여 사람이 읽게 하는 것을 목표로 하지만, 우리가 관심을 갖는 디지털 텍스트는 기계(컴퓨터)가 읽고 해석할 수 있게 함으로써 종이에 쓰일 때에 얻을 수 없었던 부가가치를 낼 수 있게 하는 것이다.

### II-1. 디지털 텍스트와 하이퍼링크

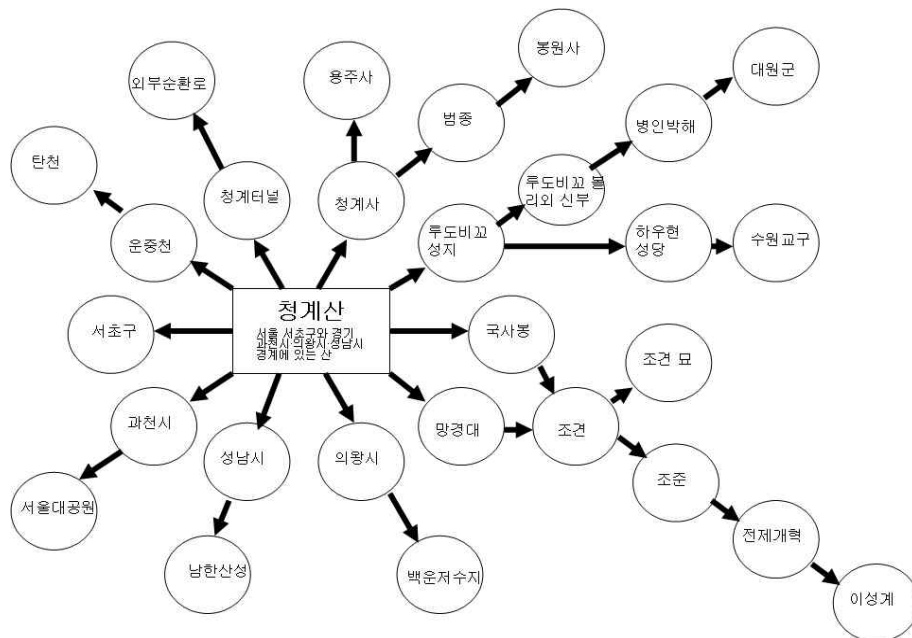
컴퓨터와 정보통신 네트워크, 모바일 컴퓨터라고 할 수 있는 스마트폰까지 대중화된 오늘날의 사회에서 우리는 이미 디지털 텍스트의 영향력을 실감하고 있다. 매일매일 접하는 인터넷상의 웹(web) 문서들이 모두 다 디지털 텍스트이다. 그러한 문서들이 종이책의 텍스트와 다른 점은 무엇인가? 그것이 어느 나라, 어느 기관의 서버에 있든, 인터넷 포털이 제공하는 서비스를 통해 검색과 열람이 가능하다. 그리고 적지 않은 양의 웹 문서에는 그 문서상에서 내용적 관련성을 가진 다른 문서를 찾아갈 수 있는 연결고리가 포함되어 있다.

컴퓨터가 숫자를 다루던 기계에서 벗어나 문자와 영상, 음향 등을 저장하고 전송할 수 있게 되면서, '책'이라고 하는 매체에 담겨 있는 다양한 종류의 지식들이 전자적인 매체로 옮겨 가게 되었다. 과거 종이에 기록되었던 정보들을 컴퓨터를 통해 처음 접할 수 있게 되었을 때, 그 새로운 책의 장점으로 주목받았던 것은 저장 공간의 획기적 절감, 데이터 접근의 신속성, 텍스트와 멀티미디어 데이터의 복합적 활용 가능성, 정보 검색의 효율성 등이었다. 전자책의 이러한 장점들은 지식의 전달과 활용의 측면에서 종이책의 한계를 극복한 커다란 변화로 인식되었다. 그러나 전자매체가 가져 온 더 큰 변화는 단순히 지식 유통의 편의성을 증진시킨 데 있는 것이 아니라, 지식의 기술 형태를 변화시키고 나아가 그 내용마저도 과거와는 다른 것을 지향하게 만든 데 있다고 할 수 있다.

종이책은 그 내용을 이루는 지식 요소가 고정되어 있을 뿐 아니라, 그 요소들의 나열 순서가 저자의 기획에 따라 일정하게 확립되어 있어서, 독자는 그 정해진 순서에 따라 책 속의 지식과 정보를 전수하게 된다. 설사 독자가 저자

의 의도와는 달리 그 책의 이 부분 저 부분을 임의로 읽어 간다 해도 그것은 저자가 미리 일정한 틀 속에 담아 놓은 지식을 부분적으로 탐색하는 것에 지나지 않는다. 반면 우리가 인터넷상의 디지털 문서를 접할 때에는 이와는 다른 방식의 ‘읽기’가 이루어진다. 한 편의 기사를 읽다가도 그 텍스트의 여러 곳에 파랗게 표시되는 부분을 클릭함으로써 다른 기사로 건너뛰기를 할 수 있는 것이다. 전자문서의 텍스트 중에 그 문서의 다른 부분 또는 인터넷상의 다른 곳에 있는 문서로 갈 수 있도록 하는 연결 요소를 ‘하이퍼링크’(hyperlink)라고 한다. 나의 관심의 방향에 따라 내가 선택하는 하이퍼링크가 달라지고, 그에 따라 내가 읽는 콘텐츠의 문맥과 내용이 달라질 수도 있다.

문서 안에 다른 문서에 대한 연결고리, 즉 하이퍼링크를 가지고 있는 전자 문서를 하이퍼텍스트 문서(Hypertext Document)라고 한다. 하이퍼텍스트의 특성을 이해하기 위해 다음의 예시를 보기로 하자.



관련 정보의 연결 고리

위에서 보인 도표는 ‘청계산’에 관한 기사에 담긴 내용 요소들과 그 각각의 요소와 관련이 있는 정보들의 관계를 도시한 것이다. 만일 이 기사의 저자가 위에 나열한 모든 요소들을 자기 글 속에 포함시키고자 한다면 그 결과물은 원고지 수십 매에 달하는 방대한 분량이 될 것이다. 하지만 위에 나열된 유관 정보들의 기사가 이미 어디엔가 존재하고 있고, 그것들을 쉽게 연결해서 볼 수 있다면, ‘청계산’에 대한 기사는 원고지 1~2매 분량으로도 충분할 수 있다.

또한 후자의 경우, 독자들은 그의 개인적 관심이 모아지는 방향에 따라 상이한 줄거리의 지식을 접할 수도 있다. 똑같이 ‘청계산’을 출발점으로 삼았다 하더라도, 어떤 경우에는 ‘청계산의 주봉인 망경대’ → ‘망경대라는 이름을 낳게 한 조건’ → ‘조건의 형 조준’ → ‘조준이 주도한 고려 말의 전제 개혁’ → ‘이성계의 여성혁명’의 경로를 따라 여말선초(麗末鮮初)의 역사를 탐구할 수 있고, 또 다른 경우에는 ‘루도비꼬 유적지’ → ‘루도비꼬 불리외 신부’ → ‘병인박해’ → ‘대원군’으로 이어지는 연결 고리를 좇아 조선 말기의 동서교섭사(東西交涉史)를 공부할 수도 있다.

지식과 정보는 유통 목적에 따라 그 내용 요소의 조직을 일정하게 유지할 필요도 있고, 자유로운 조합을 허용할 필요도 있다. 종래의 ‘종이책’은 전자에 적합한 매체였다. 우리가 새롭게 ‘하이퍼텍스트’에 주목하는 이유는 그것이 과거의 정보 매체에서는 구현하기 어려웠던 후자의 기능을 열어 주고 있기 때문이다.

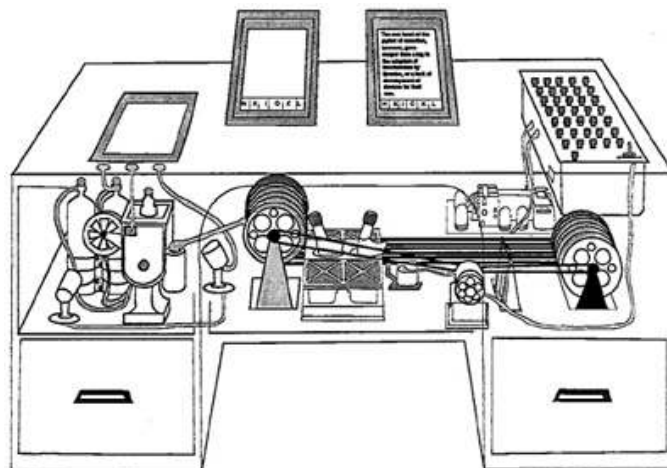
## II -2. 하이퍼텍스트 기술의 발전

‘관계가 있으면 연결한다’고 하는 하이퍼텍스트의 취지는 컴퓨터를 통해서만 현실화할 수 있다. 물론 종이매체를 이용한 백과사전이나 도서관 목록 카드가 ‘참조:…’ 또는 ‘…도 보라’라는 식으로 유관한 정보를 안내해 주는 기능을 포함하기도 하였다. 그러나 종이매체 상에서는 참조 지시를 받을 때마다 페이지를 넘겨 유관 자료를 찾아가는 것이 결코 쉬운 일이 아니며, 그 관련성의 범위도 지극히 제한적일 수밖에 없다. 하이퍼텍스트는 ‘관련 정보로의 순간적인 이동’이 가능한 경우에만 효용성을 발휘한다. 특정 정보에 대한 기록을 빠르고 정확하게 찾아내고자 하는 과제는 전자매체를 사용하는 컴퓨터 시스템을 통해 그 해결책을 찾게 되었고, 그것이 오늘날의 ‘하이퍼텍스트로 짜인 거대한 정보의 세계’- 월드와이드웹(World Wide Web)을 출현시킨 배경이 되었다. 이러한 점에서 진정한 하이퍼텍스트의 역사는 컴퓨터 발전의 역사와 궤적을 같이 한다고 할 수 있다.

### 1) 배니버 부쉬의 메멕스

하이퍼텍스트의 개념을 처음으로 제안한 사람은 제2차 세계대전 당시 미국의 대통령 과학자문위원이었던 배니버 부쉬(Vannevar Bush, 1890~1974)였다.

당시 미국은 전쟁에서 이기기 위해 그와 관련된 제 분야의 생산성을 고도화 하는 데 총력을 기울여야 하는 상황이었으며, 부쉬는 그러한 전시체제 하에서 미국의 과학기술 연구 역량을 집결시키는 구심점 역할을 하는 인물이었다. 전쟁이 막바지로 치달던 1945년 부쉬는 「마치 우리가 생각하는 것처럼 ...」(As we may think)이라는 제목의 글을 통해 ‘기억 확장기’(Memory Extender, 줄여서 Memex)라고 이름붙인 문서관리 장치를 제안하였다. 그는 그 시대에 이미 지식의 양이 폭발적으로 증가한 사실에 주목하였고, 종래의 문서관리 체계 - 수많은 자료를 한두 가지 체계에 의해 분류하고 색인을 부여한 후, 그 색인을 이용하여 필요한 자료를 찾게 하는 방법 - 로는 그 방대한 규모의 지식을 제대로 활용할 수 없다고 하는 사실을 직시하였다. 그는 인간의 사고방식이, 머릿속에 일정한 체계의 색인을 만들어 놓고 매번 그 색인을 참조하면서 생각을 전개시켜 나아가는 것이 아니며, 한 가지 생각 속에서 순간적으로 다른 생각으로 이어지는 실마리를 찾아가는 것이라고 보았다. 인간의 두뇌 밖에 있는 자료를 이용할 때에도 ‘마치 인간이 생각하는 것처럼’ 즉각적인 연상의 실마리를 좇아 지식과 정보를 찾아갈 수 있다면 얼마나 효과적일까? 부쉬는 어떠한 기계적 장치가 자신의 꿈을 현실화할 수 있을 것으로 기대하였다.

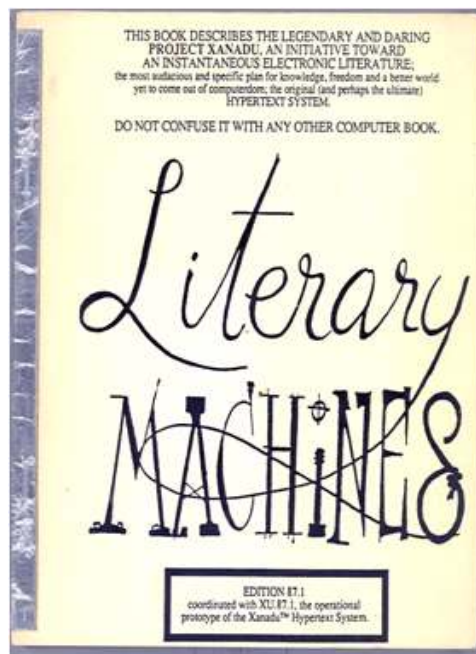


배니버 부쉬가 구상한 메멕스의 모습

## 2) 테드 넬슨의 하이퍼텍스트

부쉬가 ‘메멕스’라는 이름으로 제안한 유연한 문서관리 체계는 그로부터 약 20년이 지나 테드 넬슨(Ted Nelson)에 의해서 ‘하이퍼텍스트’라는 이름을 얻게 되었다. 넬슨은 1965년 그의 저서 『리터러리 머신』(Literary Machines)에서 재나두(Xanadu)라는 이름의 거대한 프로젝트를 제안하였다.

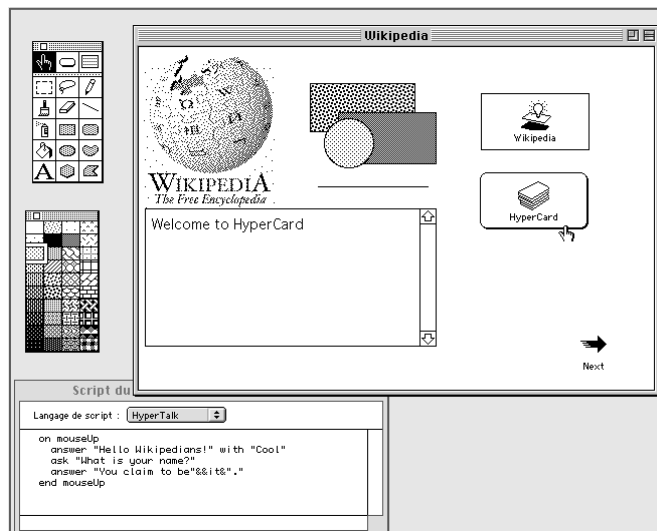
재나두는 전세계의 수많은 사람들이 남긴 기록들이 망라적으로 등재되어 있는 거대한 문서의 세계이다. 이곳에 있는 각종 문서들은 그 내부에 다른 유관 문서로 연결될 수 있는 고리를 갖는다. 재나두를 방문한 독자들은 그 연결 고리를 따라 자유롭게 다양한 기록들을 탐색할 수 있고, 또 그 스스로 또 다른 기록의 저자가 되어 지식과 정보의 새로운 경로를 만들어 낼 수도 있다. 오늘날의 월드와이드웹(World Wide Web)을 더욱 이상적으로 묘사한 듯한 이 가상세계의 중심 개념이 바로 하이퍼텍스트이다. 넬슨은 하이퍼텍스트를 ‘비순차적인 글쓰기’(Non-sequential Writing)라고 정의하였다. 그 자체로는 순서 없이 존재하는 여러 개의 텍스트 조각(Text Chunk)이지만, 그 안에는 다른 조각으로 이어지는 다양한 연줄(link)이 있어서 독자에 선택에 따라 이렇게 이어지기도 하고 저렇게 이어지기도 하는 것을 하이퍼텍스트라고 이름한 것이다.



테드 넬슨의 『리터러리 머신』 표지

### 3) 애플사의 하이퍼카드

배니버 부쉬에 의해 창안되고, 테드 넬슨에 의해 이름을 얻게 된 ‘하이퍼텍스트’는 1960년대 후반부터 실제로 컴퓨터상에서 구현되기 시작하였지만, 연구개발의 차원을 넘어서서 실용화되기 시작한 것은 1980년대 중반을 지나면서 부터였다. 1987년 미국의 애플사(Apple Computer, Inc.)는 자사의 매킨토시 시스템에서 사용되는 하이퍼텍스트 응용상품 하이퍼카드(HyperCard™)를 출시했는데, 이는 사용자들의 폭발적인 호응을 얻었다. 하이퍼카드의 성공 요인은 텍스트뿐 아니라 소리와 이미지로의 링크를 제공하여 컴퓨터의 멀티미디어 적 편리성을 부각시켰다는 점이다. 하이퍼카드처럼 멀티미디어 기능을 가진 하이퍼텍스트 시스템을 ‘하이퍼미디어’(hypermedia) 시스템이라고 부르기도 한다.



애플사의 하이퍼카드

### 4) 월드와이드웹의 탄생

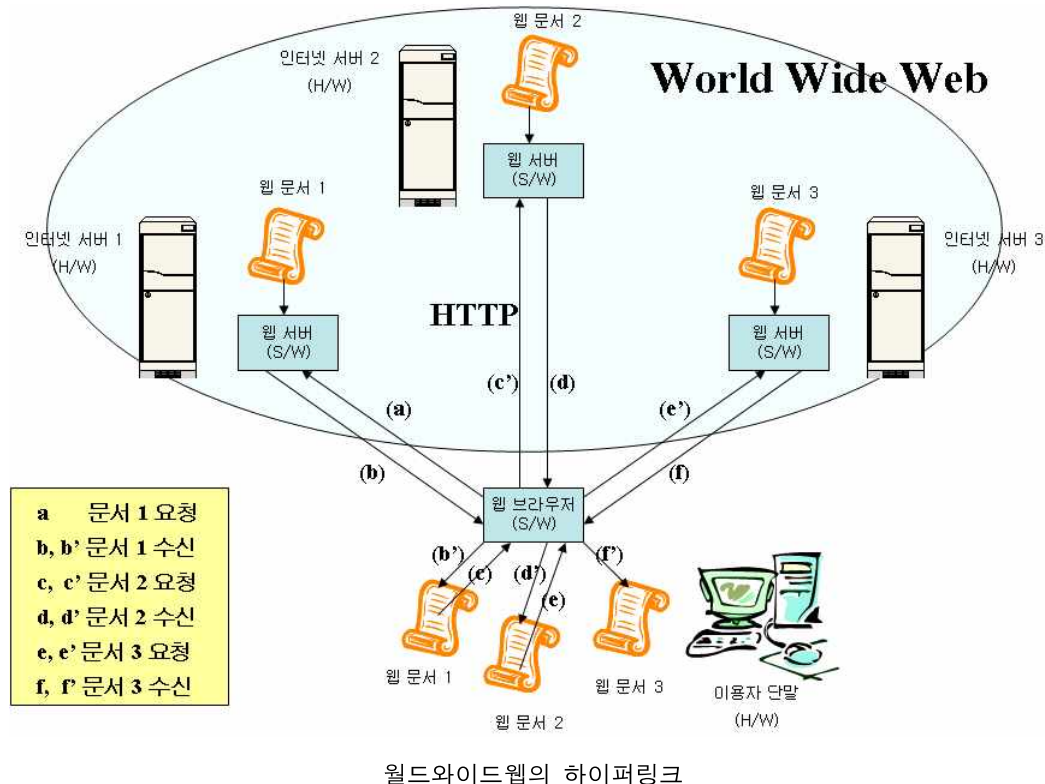
하이퍼텍스트를 모든 곳의 자료가 자유롭게 연결되는 길로 열어 놓은 것은 출발 단계에서부터 비영리적인 목적을 취한 작은 연구 프로젝트에서 비롯되었다. 1980년대 초부터 개인적으로 하이퍼텍스트를 연구하던 팀 버너스리(Tim Berners-Lee)는 스위스 제네바에 있는 유럽 소립자물리학연구소(Conseil Européen pour la Recherche Nucléaire, CERN)에 취직한 후, 그 연구소에 쌓인 엄청난 양의 자료들을 세계 각처의 관련 연구자들에게 효과적으로 제공할 수 있는 방법에 대해 고민하기 시작하였다. 이 때 그가 주목한 것은 애플사

의 하이퍼카드가 입증한 ‘하이퍼텍스트’의 편리성, 그리고 전 세계의 컴퓨터를 하나의 네트워크로 연결하고 있는 ‘인터넷’의 개방성이었다. 1989년 버너즈리는 하이퍼텍스트와 인터넷을 결합시키는 프로젝트를 제안했고, 네트워크 전문가 카이유(Robert Cailliau)의 협조로 온라인 서비스 상에서의 하이퍼텍스트 개념을 정립하였다. 이로써 이용자의 관심이 이끄는 대로 읽어 갈 수 있는 정보 조각들의 거대한 거미줄 망(web)이 탄생하게 되었으며, 그 이름은 월드와이드웹(World Wide Web, WWW)으로 지어졌다. 1991년 12월 버너즈리는 미국의 산안토니오에서 열린 Hypertext’91 학술회의에서 월드와이드웹을 세계에 소개하였다.

## II-3. 하이퍼텍스트 구현 기술

우리가 '월드와이드웹'(World Wide Web)이라고 부르는 것의 정의는 '하이퍼텍스트 문서를 지원하는 인터넷 서버들의 범세계적인 집합'이다. 그리고 여기서 말하는 '하이퍼텍스트 문서'의 기술적 의미는 'HTML로 쓰이고 HTTP를 통해 전송되는 전자 텍스트'를 말한다. HTML은 '하이퍼텍스트 표기 언어'(Hypertext Markup Language), HTTP는 '하이퍼텍스트 전송 규약'(Hypertext Transfer Protocol)의 약어이다. HTML로 작성되어 인터넷 서버에 저장된 전자문서를 우리는 통상적으로 '웹 문서' 또는 '웹 페이지'라고 부른다. 하이퍼텍스트의 구현을 위해 우리가 먼저 알아야 할 것은 이러한 맥락을 가진 웹 문서의 제작 방법이다. '월드와이드웹'이 '인터넷'과 동일시될 만큼 큰 영향력을 발휘하는 오늘날, 웹에서 운영될 수 있는 전자문서를 제작하는 것은 곧 세계의 모든 자원과 소통할 수 있는 하이퍼텍스트 콘텐츠를 제작한다는 의미를 지니게 된다.

하이퍼텍스트를 이루는 텍스트 조각들 사이의 연결을 '하이퍼링크'(hyperlink)라고 한다. 다음 그림은 월드와이드웹 환경에서 개별적인 웹 문서들 사이의 하이퍼링크가 구현되는 과정을 개념적으로 도시한 것이다.





월드와이드웹을 구성하는 인터넷 서버 머신들은 다른 웹 문서로의 연결 고리를 포함하는 여러 개의 웹 문서를 저장하고 있다. 인터넷 세계의 정보를 탐색하고자 하는 이용자는 ‘웹 브라우저’(Web Browser)라고 하는 소프트웨어를 이용하여 첫 번째 웹 문서를 호출한다. ‘인터넷 익스플로러’(Internet Explorer, IE™), 크롬(Chrome™), 파이어폭스(Firefox™), 사파리(Safari™), 오페라(Opera™) 등이 모두 웹 브라우저의 기능을 하는 소프트웨어들이다. PC나 스마트폰 등 이용자 단말기에서 운영되는 웹 브라우저는 인터넷 서버 머신에서 운영되는 ‘웹 서버’ 프로그램과 메시지를 주고받으면서 서버 머신에 저장된 웹 문서를 이용자 단말기로 불러오는 기능을 수행한다. 이 때 웹 브라우저와 웹 서버, 두 소프트웨어가 주고받는 신호에 대한 약속이 이른바 ‘하이퍼텍스트 전송 규약’, 즉 HTTP이다. 이 약속이 있음으로서 이용자 쪽의 웹 브라우저와 서버 쪽의 웹 서버가 웹 문서의 요청과 전송을 정확하게 수행할 수 있는 것이다.

첫 번째 문서를 성공적으로 불러오게 되면, 이용자는 그 문서에 포함되어 있는 다른 문서로의 연결 고리(Hypertext Link Node)를 선택함으로써 내용적 연관성이 있는 다른 문서를 호출할 수 있다. 웹 브라우저는 또 다시 HTTP에 따른 요청 신호를 해당 웹 서버로 보내고, 그 신호를 받은 웹 서버는 자기 쪽 컴퓨터에 수록된 두 번째 웹 문서를 요청자에게 전송한다.

## II -4. HTML

월드와이드웹의 플랫폼은 이미 확립되어 있고, 또 그것은 누구나 참여할 수 있도록 개방되어 있으므로, 새로운 콘텐츠의 제작을 위해서 우리가 해야 할 일은 월드와이드웹에서 통용될 수 있는 형식의 ‘디지털 텍스트’를 만드는 일이다. 이 말은 바꿔 말해, ‘웹 브라우저’와 ‘웹 서버’가 읽을 수 있는 형태의 문서를 만드는 것이라고 할 수 있으며, 더 구체적으로는 HTML로 쓰인 문서를 만드는 것이라고 할 수 있다.

HTML은 문서의 내용을 다양한 형태로 이용자에게 보여줄 수 있는 기능을 지원하는데, 정보 요소의 시작과 끝 부분에 ‘태그’(tag)라고 불리는 표시를 부가하는 형식을 취한다. 다음은 HTML 문서의 가장 기본적인 형태를 보이는 예시이다.

[예시] HTML 문서

```
<html>
  <head>
    <title>웹 사이트 목록</title>
  </head>
  <body>
    <a href="http://www.humancontent.or.kr">인문콘텐츠학회</a><br/>
    <a href="http://www.aks.ac.kr">한국학중앙연구원</a><br/>
    <a href="http://www.digerati.kr">인문정보학교실</a><br/>
  </body>
</html>
```

순수하게 문자나 삽도만으로 이루어진 기본 텍스트(Plain Text)에 이와 같은 태그를 기입하는 것을 마크업(markup)이라고 한다. 순수한 텍스트는 사람에게 읽히기 위한 것이지만, 여기에 추가된 태그들은 기계(컴퓨터)가 읽고 해석하며, 지시된 기능을 수행하게 한다. 이를 테면, 문서의 처음과 끝에 반드시 들어가야 하는 <html>..... </html>이라는 태그는 그 사이의 내용이 하나의 HTML 문서임을 알리는 역할을 한다. <head>..... </head>는 그 사이에 이 문서에 대한 정보(메타데이터: 문서의 제목, 문서의 스타일에 관한 지정 등)가 있다는 뜻이고, <body>..... </body>는 이곳이 문서의 내용을 담고 있는 부분임을 알리는 것이며, <title>..... </title>이라는 태그는 표시된 부분이 그 문서의 제목이니 제목으로 취급하라는 의미이다.

<a href="url">..... </a>라는 태그는 그곳을 클릭하면 URL(Uniform Resource Locater)로 지시되는 곳(월드와이드웹 상의 주소)의 문서로 이동하라는 의미이다. ‘닷’(anchor)이라는 이름을 가진 이 태그는 HTML 문서가 하이퍼텍스트 기능을 수행할 수 있게 해 주는 핵심 장치이다.<sup>1)</sup>

HTML의 ‘앵커’(<a>) 요소는 반드시 ‘href’라는 이름의 속성을 갖는다. ‘href’는 ‘하이퍼텍스트 참조’(Hypertext Reference)를 의미하며, 그 속성값으로 하이퍼링크 목적지의 주소를 기입하게 되어 있다.

하이퍼링크 목적지의 주소, 즉 웹 어드레스는 ‘하이퍼텍스트 전송 규약’(HTTP)에 의해 월드와이드웹 상에서 통용되는 웹 문서의 소재 정보이다. 이러한 주소 체계를 ‘URL’이라고 부르는데, 그 구성 요소는 다음과 같이 나눌 수 있다.

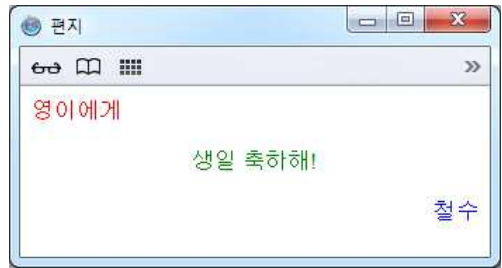
1) HTML에서 하이퍼링크를 위한 연결 고리를 ‘앵커’(anchor), 즉 ‘닷’이라고 부르는 이유는 ‘월드와이드 웹’의 세계에서 이 문서를 읽다가 저 문서로 옮겨 가는 행위를 마치 대양을 향해하는 배가 이 항구에 들렀다가 다시 저 항구에 정박하는 것에 비유한 데서 비롯된다. ‘앵커’는 웹의 세계를 향해하는 (navigate) 배가 앞으로 ‘닷’을 내려야 할 곳을 지목한다. 하나의 웹 문서 속에 여러 개의 ‘앵커’가 포함되어 있다면, 그 항해의 다음 기착지를 여러 곳으로 제안하는 것이다.



이 약화시켰던 SGML의 중요한 목적을 XML이 다시 회복시킨 것이다. SGML은 문서에 담긴 정보 요소와 그 요소들 간의 관계를 컴퓨터가 이해할 수 있도록 만든 마크업 언어인데, 1969년 미국 IBM사의 찰스 골드파브(Charles Goldfarb)와 그 동료들에 의해 GML(Generalized Markup Language)이라는 이름으로 처음 만들어졌고, 그 후 지속적으로 발전하여 1986년 국제표준기구에 의해 구조적 데이터 표현의 표준안(ISO 8876)으로 인증되었다.

XML 문서의 특징은 문서를 ‘문서의 구조’, ‘문서의 내용’, ‘문서의 모양’ 3가지로 분리한다는 것이다. 다음에 예시한 간단한 편지를 HTML로 만들었을 때와 XML로 만들었을 때를 비교하여 보자.

[예시 1] HTML 문서와 XML 문서



[예시 1-1] HTML: 문서의 내용과 모양을 한꺼번에 취급

```
<html>
  <head>
<title>편지</title>
  </head>
  <body>
    <p style="color:red text-align:left">영이에게</p>
    <p style="color:green text-align:center">생일 축하해!</p>
    <p style="color:blue text-align:right">철수</p>
  </body>
</html>
```

[예시 1-2] XML: 문서의 내용

```

<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/xsl" href="letter.xsl"?>

<편지 xmlns="letter.xsd">
  <수신>영이에게</수신>
  <본문>생일 축하해!</본문>
  <발신>철수</발신>
</편지>

```

[예시 1-3] XSD: 문서의 구조

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema targetNamespace="letter.xsd"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  <xs:element name="편지">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="수신"/>
        <xs:element ref="본문"/>
        <xs:element ref="발신"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="수신" type="xs:string"/>
  <xs:element name="본문" type="xs:string"/>
  <xs:element name="발신" type="xs:string"/>
</xs:schema>

```

[예시 1-4] XSL: 문서의 모양

```

<?xml version="1.0" encoding="utf-8"?>

<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:ltt="letter.xsd">
<xsl:template match="/">
  <html>
    <head>
      <title>편지</title>
    </head>
    <body>
      <xsl:apply-templates/>
    </body>
  </html>
</xsl:template>
<xsl:template match="ltt:편지">
  <xsl:apply-templates/>
</xsl:template>
<xsl:template match="ltt:수신">
  <p style="color:red; text-align:left"><xsl:value-of select="."/></p>
</xsl:template>
<xsl:template match="ltt:본문">
  <p style="color:green; text-align:center"><xsl:value-of select="."/></p>
</xsl:template>
<xsl:template match="ltt:발신">
  <p style="color:blue; text-align:right"><xsl:value-of select="."/></p>
</xsl:template>
</xsl:stylesheet>

```

첫 번째 HTML 문서에서는 문서의 내용과 모양이 한꺼번에 취급되었다. 반면, XML 문서는 XML(XML 텍스트), XSD(Schema Definition), XSL(Stylesheet Language) 세 부분이 모여서 하나의 완성된 문서를 만들어낸다. 위의 예시에서 XML 문서 letter.xml은 내용만 담고 있고, 스키마 정의 문서 letter.xsd는 그 문서의 내용 구조를 정의하며, 스타일 문서 letter.xsl은 문서의 모양새에 관한 정보만 가지고 있다. 자칫 쉽게 할 수 있는 일을 복잡하게 만든 듯이 보이기도 하겠지만, 하나의 문서를 만들어내는 요소를 이렇게 세 가지로 나눈 데에는 그럴만한 이유가 있다. 첫째, XML은 HTML이 표현하지 못하는 문서의 구조와 내용 요소의 성격을 스키마 정의 문서(XSD)를 통하여 명확하게 드러내 준다. 또한 문서의 내용인 텍스트(XML)와 그것을 어떻게 보여 줄 지를 정의하는 스타일시트(XSL)를 구분함으로써, 문서의 내용을 그대로 두고 모양만을 바꾼다거나, 모양에 손대지 않은 채 내용을 확장하고 수정할 수 있는 유연성을 갖게 하였다.

HTML의 태그는 미리 만들어진 것을 쓰는 것인 데 반해, XML의 태그는 문

서 제작자가 새롭게 정의할 수 있고, 그 각각의 태그에 대해 컴퓨터가 어떠한 처리를 할 것인지를 지시할 수 있다. XML의 이러한 특성 때문에 HTML로 구현된 하이퍼텍스트와 XML로 구현된 하이퍼텍스트는 큰 차이를 갖는다. 전자는 ‘텍스트’와 ‘텍스트’의 연결을 가능하게 하지만 그 텍스트 조각이 어떠한 특성을 가지고 있으며, 어떠한 의미맥락 속에 있는 것인지에 대해서는 관심을 두지 않는다. XML에 의한 하이퍼텍스트는 하이퍼링크의 노드 하나하나가 정보 요소로서의 성격을 명시적으로 드러내기 때문에 그 성격에 합당한 방법으로 다양한 형태의 하이퍼링크를 구현할 수 있다.

다음은 XML 문서로 만들어진 디지털 텍스트의 한 예이다.

[예시 2] XML 문서

[2-1] 원시 데이터

	<p>教旨 吳性鎰爲鬱陵島島監者. 光緒十六年九月 日</p>
	<p>교지 오성일을 울릉도 도감으로 임명함. 광서 16년 9월 일</p>

[2-2] XML 문서: 고문서 정보

```

<기록물 ID="R0001" 유형="고문서">
  <명칭>울릉도 도감 교지</명칭>
  <메타데이터>
    <분류>교지</분류>
    <발급자>고종(高宗)</발급자>
    <수급자>오성일(吳性鎰)</수급자>
    <발급일>1890.10.14. ~ 11. 11.</발급일>
    <소장처 ID="S0001">독도박물관</소장처>
    <원문이미지 ID="D0001"/>
  </메타데이터>
  <원문>
    <표제>教旨</표제>
    <본문><인명>吳性鎰</인명>爲<지명>鬱陵島</지명>島監者</본문>
    <일자 서력="1890. 10.14.~11. 11."><연호>光緒</연호>十六年九月 日</일자>
  </원문>
  <번역>
    <표제>교지</표제>
    <본문><인명>오성일</인명>을 <지명>울릉도</지명> 도감으로 임명함.
    <일자 서력="1890. 10.14.~11. 11."><연호>광서</연호> 16년 9월 일</일자>
  </번역>
</기록물>

```

#### [2-4] XML 문서: 위치 정보

```

<공간정보 ID="S0001" 유형='기관'>
  <명칭>독도박물관</명칭>
  <주소>경상북도 울릉군 울릉읍 도동리 581-1</주소>
  <위치>
    <위도>37.48332</위도>
    <경도>130.9008</경도>
  </위치>
</공간정보>

```

우리는 [2-1] 문서에 쓰인 오성일(吳性鎰)이 사람 이름이고, 울릉도(鬱陵島)가 지역 이름이라는 것을 알지만, 컴퓨터는 그것을 인식할 수 없다. 하지만 [2-2]에서처럼 ‘吳性鎰’에 <인명> 태그를 달아주고, ‘鬱陵島’에 <지명> 태그를 달아주면 컴퓨터도 텍스트 속에서 그것을 식별해 낼 수 있다. 이러한 문서가 수만 건 정도 있다고 가정해 보자. 컴퓨터는 그 모든 자료에 대해 완벽한 인명 색인과 지명 색인을 만들어낼 수 있다. 그 뿐 아니다. [2-4]에는 이 문서를 소장하고 있는 독도박물관이라고 하는 곳이 지구상의 어느 곳에 위치하는지를 알리는(Global Positioning) 정보가 포함되어 있다. XML 마크업을 통해 <경



도> 요소와 <위도> 요소를 컴퓨터가 인식할 수 있게 하고, XML 문서의 시각화를 담당하는 XSL(eXtensible Stylesheet Language) 문서에 그것의 처리를 지시하면,<sup>2)</sup> 컴퓨터는 전자지도 상에서 그곳의 위치를 정확하게 나타내 준다. [2-2]와 [2-3]은 독립적인 문서로 존재할 수도 있지만, ‘독도박물관’이라는 키워드를 매개로 하이퍼링크를 맺을 수 있다. 조선시대의 고문서를 3차원 위성 영상 지도를 통해 볼 수 있게 하는 서비스가 가능해지는 것이다.



고문서와 위치 정보의 연계

위의 예시들은 소략한 정보만을 담고 있지만, 이보다 훨씬 크고 복잡한 데이터도 XML이라는 디지털 텍스트의 형식을 통해 체계적, 명시적으로 기술하여, 컴퓨터에게 그 처리를 지시할 수 있다. 낱장의 고문서나 지도뿐 아니라 수십 권 분량의 백과사전이나 수백 년에 걸쳐 만들어진 역사 기록이 모두 디지털 텍스트로 편찬됨으로써, 그 속에 담긴 작은 사실 하나까지도 사이버 공간에서 지식 정보로 활용될 수 있는 것이다.

2) XSL 문서에서 전자지도 콘텐츠 공급자가 제공하는 Open API(Application Programming Interface)를 구동하는 방법을 사용한다.