

HTML5

김용만

1. HTML 개요

1-1 웹의 역사

월드 와이드 웹(**World Wide Web, www**)은 인터넷에 연결된 컴퓨터를 통해 사람들이 정보를 공유할 수 있는 정보 공간을 말한다. 인터넷은 전 세계를 연결하고 있는 국제 정보 통신망이고 웹은 그 위에서 작동하는 서비스를 의미한다.

- 미국의 ARPA(Advanced Research Projects Agency)는 1969년 현재 웹의 모태가 되는 ARPANET을 개발

- 1989년 팀 버너스리는 인터넷 공간 안에서 문서가 서로 이동할 수 있는 새로운 개념의 방법을 제안. 바로 하이퍼링크라는 특수한 기능을 사용하여 문서와 문서 사이를 이동할 수 있는 간단한 문서 개념.

- 1994년 팀 버너스리는 **W3C(World Wide Web Consortium)** 재단 창설. W3C는 웹 표준을 지정하는 기관이며 **HTML** 표준을 제정하는 단체

1-2. HTML 정의

HTML : HyperText Markup Language의 약자

HyperText : 사용자의 선택에 따라 원하는 페이지로 이동

Markup Language : 태그의 형식과 규칙을 정의한 언어

1-3. 태그의 구성 요소

1) 복합 태그

<태그명> 내용 </태그명>

<div>안녕</div>

2) 속성을 가지는 태그

<태그명 속성명=값> 내용 </태그명>

<p align="center">안녕</p>

3) 단독 태그

<태그명>

1-4. 태그 사용 규칙

- 1) <태그명></태그명>과 같이 여는 태그와 닫는 태그가 쌍으로 사용된다.
<div></div>
- 2) 태그명과 속성명, 속성명과 속성명 사이는 한 칸씩 띄운다. 그 외 공백을 두어서는 안 된다.
<div align="center"></div>
- 4) 속성명과 값은 = 부호를 사용하여 표시하고 값은 ""로 감싼다.

1-5 주석

프로그램의 실행에 영향을 미치지 않고 설명을 위한 목적으로 사용하는 코드

```
<!-- HTML 주석 -->
```

2. HTML의 기본 구조

```
<!DOCTYPE html>
<html>
  <head>
    <title></title>
  </head>
  <body>
    화면에 보여질 내용 명시
  </body>
</html>
```

<!DOCTYPE html> 태그는 웹 브라우저가 현재 웹 페이지가 **HTML5** 문서임을 인식하게 만든다.

<html> 태그는 모든 HTML 페이지의 루트 요소. Html태그에는 lang 속성을 입력할 수 있다.

<html lang="ko">

2-1 태그의 사용

1) <head> 태그 사이에 표시할 태그

<title>	문서의 제목을 지정
<meta>	HTML문서의 필요한 다양한 설정
<script>	자바스크립트 코드 명시
<style>	문서 전체의 레이아웃을 결정하여 문서에 통일감을 줄 수 있는 스타일 지정
<link>	웹 페이지에 다른 파일을 추가

2) <body> 태그

<body> 태그에는 화면의 표시할 내용을 명시

2-2 글자 태그

1) 제목

제목을 입력할 때 사용하는 제목 글자 태그

<h1> ~ <h6>

<h1>

<h2>

<h3>

<h4>

<h5>

<h6>

2) 단락

<p> 본문 글자 태그. Paragraph의 줄임말로 단락을 의미함.

<p>단락</p>

3) 줄바꿈

 줄바꿈 태그

4) 수평선

<hr> 수평선 태그

<hr size="1" width="90%" noshade>

5) 앵커 태그

<a> 앵커 태그는 서로 다른 웹 페이지 사이를 이동하거나 웹페이지 내부에서 특정한 위치로 이동할 때 사용하는 태그

네이버

6) 글자 형태

태그 이름	설명
-------	----

b	굵은 글자 태그
i	기울어진 글자 태그
small	작은 글자 태그
sub	아래에 달라 붙은 글자 태그
sup	위에 달라 붙은 글자 태그
ins	밑줄 글자 태그
del	가운데 줄이 그어진 글자 태그

7) 자주 사용하는 특수 문자

문자 엔티티	결과	설명
 		non-breaking space의 약어. 공백문자 하나를 추가한다.
<	<	Less than 의 약어. <를 표시한다.
>	>	Greater than의 약어. >를 표시한다.
&	&	Ampersand의 약어. &부호를 표시한다.
"	“	quotation maker의 약어. “를 표시한다.
©	©	Copyright의 약어. 저작권,판권을 표시한다.
™	™	Trademark의 약어. 상표를 표시한다.

2-3 공간 분할 태그

<div>

<div>공간 분할</div>

2-4 목록 태그

태그 이름	설명
ul	순서 없는 목록 태그
ol	순서가 있는 목록 태그

2-5 정의 목록

정의 목록은 특정 용어와 그 정의를 표현할 때 사용하는 태그 단어를 써놓고 그 정의를 풀이한 목록을 정의 목록이라고 한다.

태그 이름	설명
dl	정의 목록 태그
dt	정의 용어 태그
dd	정의 설명 태그

2-6 테이블 태그

```
<table>
<caption>제목</caption>
<tr>
  <td>첫 번째 셀</td>
  <td>두 번째 셀</td>
</tr>
</table>
```

2-7 이미지 태그

```

```

2-8 입력 양식 태그

```
<form action="signup.jsp" method="post">
  ID <input type="text" name="name"><br>
  비밀번호 <input type="password" name="password"><br>
  좋아하는 계절<br>
    <input type="checkbox" name="season" value="봄">봄
    <input type="checkbox" name="season" value="여름">여름
    <input type="checkbox" name="season" value="가을">가을
    <input type="checkbox" name="season" value="겨울">겨울<br>
  거주지<br>
    <input type="radio" name="city" value="서울">서울
    <input type="radio" name="city" value="부산">부산
    <input type="radio" name="city" value="대구">대구
  학력<br>
  <select name="school">
    <option value="초등학교">초등학교</option>
```

```

<option value="중학교">중학교</option>
<option value="고등학교">고등학교</option>
<option value="대학교">대학교</option>
</select>
자기소개<br>
<textarea rows="10" cols="30" name="content"></textarea>
<input type="submit" value="전송"><br>
</form>

```

3. HTML5 특징

- 1) 웹 개발의 플랫폼 : 의미가 더 분명한 태그들, 사이트나 창 사이의 더 나은 데이터 교환, 애니메이션, 더 나은 멀티미디어 지원 등, 더 좋은 사용자 인터페이스 구현 가능
- 2) 더 분명한 마크업 : 내용을 설명하는데 직접 연관된 요소 추가 (제목, 푸터, 내비게이션 절, 사이드바, 글을 정의하는 요소들)
- 3) 플러그인에 덜 의존하는 멀티미디어 : 오디오, 비디오 기능 제공
- 4) 웹 소켓 : **HTML5**는 서버와의 연결을 유지시키는 웹 소켓을 지원
- 5) 클라이언트 측 스토리지 : 웹 스토리지와 웹 **SQL** 데이터베이스 **API**를 이용하면 브라우저에서 동작하긴 하지만 모든 데이터를 사용자의 컴퓨터에 보관하는 애플리케이션을 만들 수 있음.
- 6) 더 좋은 인터페이스 : 테이블에 스타일을 적용하거나, 둥근 모서리를 만드는 등 사용자 인터페이스를 좀 더 풍요롭게 사용 가능.
- 7) 더 좋은 폼 : 더 나은 사용자 인터페이스 컨트롤 제공
- 8) 향상된 접근성 : 스크린 리더 같은 프로그램을 사용해서 보조 기술에 의존하는 사람에게 더 좋은 웹 페이지 접근성을 제공.

3-1 HTML5 지원 API

API	내용
웹 스토리지(Web Storage)	로컬 스토리지(Local Storage). 웹의 내용을 사용자 컴퓨터에 임시저장하는 기능
드래그앤드롭(Drag & Drop)	웹 페이지상에서 원하는 항목을 드래그할 수 있게 해주는 기능
지오로케이션(Geolocation)	공개지도 API와 함께 사용하면 자신의 현재 위치를 알아낼 수 있는 기능
캔버스(Canvas) 2D	Canvas 요소와 그리기 API를 제공
웹 워커(Web Workers)	CPU에 부하를 많이 주는 작업을 여러 워커(Worker)로 나누어 작업

웹 SQL 데이터베이스 (Web SQL Database)	웹 브라우저에 내장 데이터베이스를 만들고 SQL 질의를 사용할 수 있는 기능
------------------------------------	---

3-2 웹 표준, 웹 접근성, 웹 호환성

웹 표준

웹 기술을 표준화하기 위한 일련의 단계와 요구사항이다. 플러그인 기술(예: 액티브X 등)과 같은 비표준 기술을 배제하고, W3C(World Wide Web Consortium)에서 나온 권고안을 사용하는 것을 말하며, 웹 문서의 구조와 표현, 동작을 구분해서 사용하는 것을 뜻한다

웹 접근성

장애를 가진 사람과 장애를 갖지 않은 사람 모두가 웹사이트를 이용할 수 있게 하는 방식을 가리킨다. 사이트가 올바르게 설계되어 개발되고 편집되어 있을 때 모든 사용자는 정보와 기능에 동등하게 접근할 수 있다. 한국형 웹 콘텐츠 접근성 지침 2.1 문서는 웹 콘텐츠의 접근성을 향상시키기 위한 기술적 규격을 포함하고 있다.

웹 호환성

웹사이트가 작동하는 서비스 이용자 단말기(PC, 모바일기기 등)의 하드웨어 및 소프트웨어 환경이 다른 경우에도 동등한 서비스를 제공하는 것을 말한다. 전자정부서비스 웹 호환성 지침은 행정기관 및 공공기관이 전자정부서비스의 호환성 확보를 위해 지켜야 할 사항을 규정함을 목적으로 한다.

3-3 HTML5 독타입

독타입은 간단히 말하면 브라우저 혹은 다른 파서에게 지금 보고 있는 문서의 종류를 알려주는 수단.

[HTML5이전 독타입]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional/EN"
http://www.w3.org/TR/xhtml1/DTD/xhtml1 - transitional.dtd>
```

[HTML5 독타입]

```
<!DOCTYPE html>
```

3-4 head 요소

[HTML5이전 문자 인코딩 표시]

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

[HTML5 문자 인코딩 표시]

```
<meta charset="utf-8">
```

4. HTML5의 기본적인 문서 만들기

4-1 HTML5에서 사라진 태그

태그	속성
<acronym>	영문 축약형 사용시, 예를 들면 NATO, NASA, GUI 등
<applet>	임베디드 애플릿 사용 시
<basefont>	글꼴의 형태 정의
<big>	글꼴의 크기 지정
<center>	텍스트 위치 중앙 정렬
<dir>	디렉토리 리스트
	글꼴
<frame>	프레임
<frameset>	프레임 부속 태그
<noframes>	프레임 부속 태그
<strike>	글꼴 효과(글 취소 효과)
<tt>	글꼴 효과(타자기 효과)
<u>	글꼴 효과 언더라인
<xmp>	pre 태그와 동일 효과

4-2 새로운 태그

페이지의 구조를 더 잘 나타내기 위해 새로운 태그 추가

header	페이지나 절의 헤더 영역을 정의
footer	페이지나 절의 푸터 영역을 정의
nav	페이지나 절의 내비게이션 영역을 정의

section	페이지의 논리적 영역 또는 내용의 그룹을 정의
article	하나의 글 또는 완결된 내용을 정의
aside	부차적인 또는 관련 있는 내용을 정의
meter	특정 범위 내의 측정 값을 나타냄
progress	어떤 목표까지의 진척도를 실시간으로 보여줌
hgroup	제목과 그에 관련된 부제목을 하나로 묶음
figure,figcaption	그림이나 비디오 같은 멀티미디어 콘텐츠에 캡션을 붙일 때
audio,video	멀티미디어 콘텐츠를 표시
embed	플러그인 콘텐츠 표시
mark	텍스트에 형광펜으로 칠한 것과 같은 강조 효과
time	날짜와 시간을 표시
ruby,rt,rp	루비 문자 표시
canvas	웹상에 그래픽을 표시
command	사용자 실행 명령어 표시
details	추가적인 정보나 사용자가 요청하는 정보 표시
datalist	사용자가 텍스트 필드에 내용을 입력할 때 선택할 수 있는 값들을 리스트
keygen	양식(form)을 서버로 전송할 때 한 쌍의 키를 만들어 private key (프라이빗 키)는 로컬에 저장하고 public key (퍼블릭 키)는 서버에 저장
output	수학적인 계산의 결과값을 표시

4-3 시맨틱 마크업 사용하기

의미 없는 <div> 태그의 사용보다 문서의 내용을 쉽게 이해할 수 있도록 의미를 가지는 새로운 태그를 도입. 시맨틱 요소(**Semantic Element**) 또는 시맨틱 태그라 지칭.

header	사이트에 대한 소개 정보나 메인 메뉴, 사이트 로고 등이 포함
hgroup	제목과 부제목을 묶어주는 역할
nav	사이트의 메뉴나 링크 같은 내비게이션 요소 포함
section	실제 문서 내용
article	문서 내용이 많을 경우 여러 개의 <article> 요소로 분할
aside	문서의 주요 내용 외의 내용들. 문서의 주영역 주변에 배치
footer	작성자 정보나 저작권 정보, 관련 문서 링크 등 부가 정보 배치

5. HTML5 품

5-1 필드

이메일 필드	<label for="email">이메일 연락처</label> <input type="email" name="email" id="email">
URL 필드	<label for="url">테스트 URL</label> <input type="url" name="url" id="url">
전화번호 필드	<input type="tel">
검색 필드	<input type="search">
슬라이더(범위) 필드	<label for="priority">중요도</label> <input type="range" min="0" max="10" name="priority" value="0" id="priority">
숫자 필드	<label for="estimated_hours">예상 시간</label> <input type="number" name="estimated_hours" min="0" max="1000" id="estimated_hours">
날짜 필드	<label for="start_date">시작 날짜</label> <input type="date" name="start_date" value="2011-02-01" id="start_date">
날짜와 시간 필드	<input type="datetime">
색상 필드	<label for="project_color">프로젝트 색상</label> <input type="color" name="project_color" id="project_color">

Search	검색 필드(하나 혹은 그 이상의 검색어를 입력하기 위한 한 줄 텍스트 입력창)을 제공
--------	---

5-2 속성

자동 포커스	<code><input type="text" autofocus></code>
required	해당 필드가 제대로 채워졌을 때만 폼을 전송
플레이스홀더	<code><input type="email" placeholder="me@example.com"></code>
pattern	사용자가 입력한 내용이 유효한 값이 되기 위해 반드시 일치해야 하는 정규 표현식을 지정
disabled	이 속성은 모든 필드가 정확하게 채워지기 전까지 전송 버튼이 비활성화되어 있게 할 때 자주 사용됨. Disabled 속성을 지닌 폼 컨트롤은 폼과 함께 전송되지 않음
readonly	사용가 해당 값을 수정할 수 없게 만듦. disabled 와 달리 이 필드는 포커스를 받을 수 있고, 해당 값이 폼과 함께 전송됨
multiple	폼 컨트롤에 값을 여러 개 넣을 수 있음을 나타냄. select , email , file 에 적용가능
form	HTML5에서 form 속성을 이용하면 폼 요소를 그것들을 감싸고 있지 않은 다른 폼과 연관시킬 수 있음. form 속성은 해당 필드셋이나 컨트롤이 연결돼야 하는 form 요소의 id 를 값으로 취한다.
autocomplete	폼이나 폼의 컨트롤이 자동완성 기능을 사용해야 할지 말지를 지정

6. HTML5 오디오와 비디오

6-1 지원 비디오 파일 형식 및 코덱

비디오 코덱 : 멀티미디어 데이터 스트림을 인코딩, 디코딩하는 알고리즘을 정의.

비디오	파일형식	코덱
파이어폭스 3.5	ogg	테오라, Vorbis
오페라 10.5	ogg	테오라, Vorbis

사파리	mp4	H.264, AAC
크롬5	ogg mp4	테오라, Vorbis H.264, AAC
인터넷 익스플로러9	mp4	H.264, AAC
인터넷 익스플로러8	지원안함	지원안함

6-2 video 태그 사용

```
<video src="http://www.videotest.com/test.ogv"></video>
```

video 태그 속성

속성	설명
src	동영상 파일의 경로를 지정
poster	동영상이 화면에 나타나지 않을 때 대신 표시할 그림을 지정
preload	동영상이 백그라운드에서 다운로드되어 재생 단추를 눌렀을 때 재생되게 함
autoplay	동영상을 자동 재생
loop	반복 재생 횟수를 지정
controls	동영상 화면에 컨트롤 막대를 추가
width	동영상 화면에 너비를 지정
height	동영상 화면의 높이를 지정

6-3 다중 비디오 포맷 지원

다중 비디오 포맷을 포함할 수 있게 **video** 요소에 **source** 요소를 정의할 수 있으며, 이로써 모든 유저 에이전트는 자신에게 적합한 포맷을 이용해 동영상을 재생할 수 있다.

```
<source src="example.mp4" type="video/mp4">
<source src="example.webm" type="video/webm">
<source src="example.ogv" type="video/ogg">
```

```
<source src="example.mp4" type='video/mp4; codecs="avc1.42E01E, mp4a.40.2"'>
<source src="example.webm" type='video/webm; codecs="vp8, vorbis"'>
<source src="example.ogv" type='video/ogg; codecs="theora, vorbis"'>
```

6-4 오디오 지원 파일

브라우저	파일 형식
파이어폭스 3.5	ogg, wav
오페라 10.5	wav
사파리	mp3, wav
크롬5	ogg, mp3
인터넷 익스플로러8	지원 안함

6-5 audio 태그 사용

```
<audio src="http://www.audiotest.com/test.wav"></audio>

<audio controls autoplay>
  <source src="sample.ogg">
</audio>
```

7. 웹스토리지

웹 스토리지 **API**는 간단한 데이터를 사용자의 컴퓨터나 기기에 로컬로 저장하는 방법에 대한 표준을 정의한다.

Web Storage 지원 이전부터 사용된 **Cookie**의 한계

1. 크기 제한 : **Cookie**는 하나의 사이트에 저장할 수 있는 크기가 제한되어 있음(4KB). **Web Storage**는 5MB(크롬, 사파리, 파이어폭스, 오페라, 인터넷 익스플로러(10MB))
2. 서버로 자료 전송 : **Cookie**는 **HTTP** 요청에 자동으로 포함되어 서버로 전송. **Web Storage**는 서버로 데이터 전송하지 않음
3. 유효 기간 : **Cookie**는 유효 시간이 경과하면 자동으로 삭제되지만 **Web Storage**는 삭제되지 않음.

4. 객체 저장 : Cookie와는 다르게 Web Storage에는 javascript 객체를 저장할 수 있음

7-1 세션 스토리지

하나의 창이나 탭에 한정된 데이터를 보관할 수 있다. 사용자가 같은 사이트를 두 개의 창에서 접속하고 있더라도, 각 창은 자체적인 개별 세션 스토리지 객체를 가지며, 따라서 데이터를 별도로 유지하게 된다. 특정 사이트 접속해서 사용자 세션이 지속되는 동안만 세션 스토리지도 지속된다.

데이터 저장하기

```
sessionStorage.setItem('name',first_value); 또는  
sessionStorage.name = first_value;
```

저장된 데이터 읽기

```
var name = sessionStorage.getItem('name'); 또는  
var name = sessionStorage.name;
```

7-2 로컬 스토리지

세션 스토리지와 달리 로컬 스토리지를 이용하면 브라우저를 통해 사용자의 컴퓨터에 데이터를 지속적으로 저장할 수 있다. 나중에 사용자가 다시 사이트를 방문했을 때 로컬 스토리지에 저장된 모든 데이터를 다시 이용할 수 있다.

데이터 저장하기

```
localStorage.setItem('name',first_value); 또는  
localStorage.name = first_value;
```

저장된 데이터 읽기

```
var name = localStorage.getItem('name'); 또는  
var name = localStorage.name;
```

7-3. 스토리지 제한

인터넷 익스플로러에서는 거의 10MB에 가까운 사용자 데이터를 웹 애플리케이션에서 저장할 수 있다. 크롬, 사파리, 파이어폭스, 오페라에서는 모두 W3C규격에서 제안하는 양인 5MB까지 사용자 데이터를 저장할 수 있다.

브라우저의 스토리지 크기에 따라 스토리지 할당량을 초과했는지 체크하기

```
try{
  sessionStorage["name"] = "test";
}catch(exception){
  if(exception == QUOTA_EXCEEDED_ERR){
    alert("스토리지 할당량이 초과되었습니다.");
  }
}
```

8. File API

8-1. <input type="file">의 프로퍼티 목록

프로퍼티명	설명
accept	선택을 허가하는 파일의 종류를 MIME Type 로 지정함. 여러 개의 MIME Type 을 지정하는 경우에는 콤마 구분자로 지정함 예) image/*,video/* 또는 image/png,image/gif
multiple	복수의 파일을 선택 가능하게 함
files	선택된 파일의 File 객체를 포함한 배열
onchange	파일을 선택할 때 실행되는 이벤트 핸들러

8-2. 파일 정보 얻기

파일 정보를 얻는 속성 및 메서드

size	파일 크기(바이트)를 반환
type	파일의 MIME 형식을 소문자로 변환된 값으로 반환. MIME 형식이 불분명하면 빈 문자열을 반환.
name	파일 이름을 반환. 파일 경로는 포함되지 않음
lastModifiedDate	파일의 업데이트 시간을 나타내는 문자열 반환. 파일의 업데이트 날짜가 불분명하면 빈 문자열을 반환
slice(start,length) slice(start,length,contentType)	파일 데이터 중 start 위치에서 length 만큼의 범위의 데이터를 나타내는 Blob 객체를 반환. 선택적으로 세 번째 매개변수에 HTTP 응답 헤더의 Content-Type 값을 지정할 수 있음. 이것은 파일 데이터를 HTTP 를 통해 얻었을 때 데이터를 제한하는 데 사용할 수 있음.

8-3. FileReader 객체를 이용한 파일 데이터 읽기

파일의 데이터를 읽기 위한 객체 메서드

<code>FileReader = new FileReader()</code>	FileReader 객체 반환
<code>readAsBinaryString(blob)</code>	매개변수에 지정된 Blob 객체(또는 File 객체)가 나타내는 파일의 데이터를 읽고 그 데이터를 바이너리로 보존.
<code>readAsText(blob)</code> <code>readAsText(blob,encoding)</code>	매개변수에 지정된 Blob 객체(또는 File 객체)가 나타내는 파일의 데이터를 읽고 그 데이터를 UTF-8 텍스트로 변환. 읽어들인 텍스트가 EUC-KR 이여도 UTF-8 로 변환. 두 번째 매개변수에 문자 인코딩을 나타내는 문자열을 지정할 수 있음. 두 번째 매개변수가 지정되면 읽은 데이터는 지정된 문자 인코딩(EUC-KR 지정 가능)으로 변환.
<code>readAsDataURL(blob)</code>	매개변수에 지정된 Blob 객체(또는 File 객체)가 나타내는 파일의 데이터를 읽고 그 데이터를 Data URL 형식의 문자열로 변환.
<code>readAsArrayBuffer(blob)</code>	매개변수에 지정된 Blob 객체(또는 File 객체)가 나타내는 파일의 데이터를 읽고 그 데이터를 ArrayBuffer 객체로 변환.
<code>abort()</code>	파일 읽기를 중지
<code>result</code>	읽어들인 데이터를 반환. 데이터의 형식은 읽어들인 방식에 따라 다름.

8-4. 이벤트

파일을 읽는 과정에서 발생하는 이벤트

이벤트명	이벤트 발생 시점
<code>loadstart</code>	파일을 읽기 시작할 때 발생
<code>progress</code>	파일을 읽는 중에 연속적으로 발생
<code>abort</code>	<code>abort()</code> 메서드 호출 등의 이유로 파일 읽기가 중지되면 발생
<code>error</code>	파일을 읽다가 오류가 발생하면 발생
<code>load</code>	파일 읽기가 정상적으로 종료되면 발생
<code>loadend</code>	파일 읽기에 대한 처리가 종료되면 발생. 파일을 성공적으로 읽었는지 실패하였는지 여부에 상관없이 발생

각 이벤트를 처리하는 이벤트 핸들러

onloadstart	loadstart 이벤트가 발생했을 때 실행되는 핸들러를 정의할 수 있음
onprogress	progress 이벤트가 발생했을 때 실행되는 핸들러를 정의할 수 있음
onload	load 이벤트가 발생했을 때 실행되는 핸들러를 정의할 수 있음
onabort	abort 이벤트가 발생했을 때 실행되는 핸들러를 정의할 수 있음
onerror	error 이벤트가 발생했을 때 실행되는 핸들러를 정의할 수 있음
onloadend	loadend 이벤트가 발생했을 때 실행되는 핸들러를 정의할 수 있음

8-5. 로드 상태

파일의 읽기 상태를 파악하기 위한 속성

readyState	로드 상태를 나타내는 수치를 반환. 1 : File 객체가 생성되었지만 아직 파일 데이터를 읽지 않은 상태(EMPTY) 2 : 파일을 읽는 도중(LOADING) 3 : 파일의 모든 데이터 읽기가 끝나고 메모리에 저장된 상태, 또는 어떠한 오류가 발생하여 파일의 읽기가 중단된 상태(DONE)
EMPTY	항상 0을 반환
LOADING	항상 1을 반환
DONE	항상 2를 반환

사용 예)

<pre>if(reader.readyState == 1){ ... } 대기중 if(reader.readyState == reader.LOADING) { ... } 로드중 if(reader.readyState == 1){ ... } 완료</pre>

8-6. 오류 처리

오류의 원인을 파악하기 위한 속성

error.code	오류의 종류를 나타내는 숫자를 반환 1:파일을 찾을 수 없을 때(NOT_FOUND_ERR) 2:웹 애플리케이션에서 사용하기엔 보안상 좋지 않다고 판단했을 때. File에 너무 많은 읽기 호출이 있을 때. 사용자가 파일을 선택한 후 파일의 내용이 바뀌었을 때(SEcurity_ERR) 3:abort() 메서드의 호출 등의 이유로 파일 읽기 처리가 중지되었을 때(ABORT_ERR)
------------	---

	4:파일 권한 및 기타 응용 프로그램에서 잠금 처리 등의 이유로 파일을 읽을 수 없었을 때 (NOT_READABLE_ERR) 5:Data URL 길이가 URL 길이의 상한을 초과할 때, readAsText() 메서드로 불러올 때는 제외(ENCODING_ERR)
error.NOT_FOUND_ERR	항상 1을 반환
error.SECURITY_ERR	항상 2을 반환
error.ABORT_ERR	항상 3을 반환
error.NOT_READABLE_ERR	항상 4를 반환
error.ENCODING_ERR	항상 5를 반환

9. 캔버스

9-1 <canvas>태그

```
<canvas id="rect1" width="300" height="200"></canvas>
```

9-2 다양한 도형 그리기

1) 사각형 그리기

fillRect(x,y,width,height)	색이 칠해진 사각형
strokeRect(x,y,width,height)	테두리만 있는 사각형
clearRect(x,y,width,height)	특정 영역을 지우고 완전히 투명하게 만듦

2) 경로 그리기

beginPath()	경로 시작
moveTo(x,y)	지정된 좌료부터 시작하는 서브패스를 새롭게 만듦
lineTo(x,y)	서브패스의 마지막 위치에서 지정된 자표로 직선을 추가함
closePath()	경로 종료
stroke()	경로 테두리 그리기

fill()	도형 채우기
--------	--------

3) 원, 호 그리기

<p>arc(x, y, r, startAngle, endAngle, anticlockwise)</p> <p>x,y,반지름, 호의 시작점, 끝점을 각도로 표시, 호를 그릴 때 반시계방향으로 그릴지 여부</p>

9-3 스타일 적용하기

1) 색상

<p>fillStyle=color</p> <p>strokeStyle=color</p>

2) 투명도

globalAlpha=value

3) 선 스타일

lineWidth=value	선의 두께. 기본값 1.0
lineCap=type	<p>선의 끝모양 기본값 : butt</p> <p>butt : 기본 값으로 아무런 효과 없음</p> <p>round : 선 너비의 1/2을 반지름으로 하는 반원이 선 양쪽 끝에 그려짐</p> <p>square : 선 양쪽 끝에 사각형이 그려지는데 그 높이가 선 너비의 1/2</p>
lineJoin=type	<p>선과 선의 만남. 선의 교차점에 대한 처리. 기본값 : miter</p> <p>round : 선과 선이 만나는 부분이 둥글게 처리</p> <p>bevel : 두 선 연결 부분에 단면을 만듬</p> <p>miter : 연결한 흔적이 남지 않고 마치 처음부터 하나의 선이었던 것처럼 연결</p>
miterLimit=value	<p>lineJoin 프로퍼티에서 유형을 'miter'로 선택했을 경우 연결 흔적을 없애줌. 이때 연결된 부분에는 뾰족한 꼭지점이 생기는데 이부분을 얼마나 잘라낼 것인지 결정하는 속성이 miterLimit. 값은 두 개의 선이 만나는 안쪽 점에서 바깥쪽 점까지의 거리를 가리킴</p>

4) 그림자 효과

<code>shadowOffsetX=float</code>	객체로부터 그림자가 x 축 방향으로 얼마나 떨어져 있는가를 나타냄. 기본값은 0 이며, 음수이면 왼쪽에 그림자가 생김
<code>shadowOffsetY=float</code>	객체로부터 그림자가 y 축 방향으로 얼마나 떨어져 있는가를 나타냄. 기본값은 0 이며, 음수이면 위쪽에 그림자가 생김
<code>shadowBlur=float</code>	그림자가 얼마나 흐릿한가를 나타냄. 기본값은 0
<code>shadowColor=color</code>	그림자 색상을 지정하는데 기본값은 완전히 투명한 검정색

9-4 이미지 사용하기

<code>drawImage(image,x,y)</code> <code>drawImage(image,x,y,width,height)</code> 이미지 자르기 <code>drawImage(image,sx,sy,sWidth,sHeight,dx,dy,dWidth,dHeight)</code>

9-5 변형하기

위치 옮기기 <code>translate(x,y)</code> 회전시키기 <code>rotate(각도)</code> 크기 조절하기 <code>scale(x,y)</code>

10. 지오로케이션

지오로케이션을 이용하면 방문자가 자신의 현재 위치를 공유할 수 있다.
방문자의 위치는 다음과 같은 사항에 의해 결정

- IP주소
- 무선 네트워크 연결
- 셀 타워(cell tower)
- 기기의 GPS 하드웨어

10-1 현재 위치 얻기

getCurrentPosition() 함수

```
window.navigator.geolocation.getCurrentPosition(  
    successCallback, //위치 정보를 얻었을 때 실행되는 함수  
    errorCallback, //위치 정보를 얻지 못 했을 때 실행되는 함수  
    options //위치 정보에 대한 각종 매개 변수를 포함하는 자바스크립트 객체 지정  
);
```

위치 정보를 얻었을 때 실행되는 함수의 첫번째 매개변수에는 이벤트 객체가 자동으로 할당되고 이벤트 객체로 부터 위치 정보를 나타내는 다양한 값을 얻을 수 있다.

위치 정보를 나타내는 값

event.coords.latitude	위도를 수치로 반환.단위는 도(°)
event.coords.longitude	경도를 수치로 반환.단위는 도(°)
event.coords.accuracy	위도 및 경도의 정밀도를 반환. 단위는 미터
event.coords.altitude	GPS 고도를 반환. 단위는 미터. 값을 얻을 수 없으면 null 반환
event.coords.altitudeAccuracy	GPS 고도의 정밀도를 반환. 단위는 미터. 값을 얻을 수 없으면 null를 반환
event.coords.heading	진행 방향을 0부터 360사이의 수치로 반환.이 값은 각도를 나타내며, 단위는 도(°)
event.coords.speed	이동 속도를 수치로 반환. 단위는 m/s. 이 값을 얻을 수 없을 때는 null를 반환
event.coords.timestamp	위치 정보를 얻었을 때의 시간을 나타내는 Date 객체를 반환

10-2. 위치 정보를 연속해서 얻기

연속해서 위치 정보 얻기

```
watchId = window.navigator.geolocation.watchPosition(  
    successCallback,  
    errorCallback,  
    options  
);
```

watchPosition 메서드에 설정하는 옵션값

옵션값	설명
enableHighAccuracy	정확도가 높은 위치 정보를 요청
timeout	위치 정보 확인에 시간제한을 설정(밀리 세컨드). 시간제한을 초과하면 에러 발생
maximumAge	위치 정보의 유효 기간을 설정함(밀리 세컨드)0을 지정하면 항상 새로운 위치 정보를 확인함.

10-3. 실시간 감시의 정지

실시간 감시의 정지

```
window.navigator.geolocation.clearWatch(watchId);
```

11. 웹 SQL 데이터베이스

11-1 데이터베이스 연결하기

```
openDatabase(name ,version,displayName, estimatedSize,createCallback);  
name : 데이터 베이스의 식별자를 문자열로 지정 데이터 베이스 이름은 대소문자 구분.  
지정한 이름의 데이터베이스가 없을 때는 새로 만듦  
version:대기 중인 데이터베이스의 버전을 문자열로 지정  
displayName : 사용자에 표시될 데이터베이스 이름을 지정  
estimatedSize : 데이터베이스 용량, 단위는 바이트  
createCallback : 처리가 완료되었을 때 호출할 콜백 함수를 지정
```

ex)

```
var db = openDatabase("workdb","1.0","메모",1024*1024);
```

```
var db = openDatabase("workdb","1.0","메모",1024*1024,  
    function(database){  
  
    });
```

11-2 트랜잭션

트랜잭션이란 데이터베이스에 대한 여러 개의 조작을 하나로 묶어 처리하는 기술

```
transaction(callback, errorCallback, successCallback);
```

- 첫번째 인수 외에는 생략할 수 있음
- **callback** : 트랜잭션 안의 처리를 지정하기 위해 이용됨
- **errorCallback** : 트랜잭션이 에러로 종료되었을 때 호출
- **successCallback** : 트랜잭션이 정상 종료될 때 호출

ex)

```
db.transaction(function(tx){  
    //인수 tx를 사용하여 SQL을 실행  
});
```

11-3 테이블 만들기

```
createNotesTable = function()  
{  
    db.transaction(function(tx){  
        tx.executeSql(  
            "CREATE TABLE notes (id INTEGER \\  
            PRIMARY KEY, title TEXT, note TEXT)", [],  
            function(){ alert('노트 테이블이 잘 만들어졌습니다!'); },  
            function(tx, error){ alert(error.message); } );  
    });  
};
```

11-4 CRUD 수행

```
fetchNotes = function(){  
    db.transaction(function(tx) {  
        tx.executeSql('SELECT id, title, note FROM notes', [],  
            function(SQLTransaction, data){  
                for (var i = 0; i < data.rows.length; ++i) {  
                    var row = data.rows.item(i);  
                    var id = row['id'];  
                    var title = row['title'];  
  
                    addToNotesList(id, title);  
                }  
            });  
    });  
};
```

12. 드래그 앤 드롭

12-1. draggable 속성

img요소와 href 속성이 마크업된 a 요소는 원래 드래그할 수 있는 요소로 규정되어 있다. img요소와 href속성이 마크업된 a 요소를 외에 모든 요소를 HTML5에서는 draggable 속성을 true로 지정함으로써 드래그할 수 있도록 규정하고 있다.

HTML 5 에 새로이 추가된 속성 중에 draggable 속성이 있다. true, false 값을 가질 수 있다. draggable 속성이 true 이면 drag 할 수 있는 것이고, false 면 drag 못한다.
예)

```
<div draggable="true">드래그 가능</div>  
<div draggable="false">드래그 불가능</div>
```

12-2. 드래그 앤 드롭의 이벤트

드래그 측에서 발생하는 이벤트

이벤트	설명
dragstart	드래그 요소를 마우스 버튼을 눌러 드래그하기 시작했을 때 드래그 요소에서 발생
drag	드래그 요소를 마우스 버튼을 누른 상태에서 드래그하는 중에 드래그 요소에서 발생
dragend	드래그를 종료했을 때 발생

드롭 측에서 발생하는 이벤트

dragenter	드래그 요소가 드롭 요소에 들어왔을 때 드롭 요소에서 발생
dragleave	드래그 요소가 드롭 요소에서 나갔을 때 드롭 요소에서 발생
dragover	드래그 요소가 드롭 요소 위에서 드래그되고 있을 때 드롭 요소에서 연속해서 발생
drop	드래그 요소가 드롭 요소에 드롭되었을 때 드롭 요소에서 발생

12-3. DataTransfer

DataTransfer 객체는 Drag & Drop API 를 사용하는데 있어서 반드시 있어야 하는 객체다. 이는

Drag & Drop 되는 대상 **Data** 를 담는 역할을 한다.

데이터 전송 기능에 관한 메서드

<code>event.dataTransfer.setData(format,data)</code>	첫번째 매개변수로 포맷 문자열을 지정. 이곳에 지정한 포맷과 일치하는 값을 두번째 매개변수로 지정. 두번째 매개변수로 문자열만 지정 가능
<code>event.dataTransfer.getData(format)</code>	첫번째 매개변수에 지정한 포맷의 전송 데이터를 반환. 지정된 포맷의 데이터가 지정되어 있지 않으면 공백 문자열을 반환
<code>event.dataTransfer.clearData()</code> <code>event.dataTransfer.clearData(format)</code>	데이터 전송용으로 지정된 데이터를 모두 제거. 첫번째 매개변수로 포맷 문자열을 지정하면 해당 형식과 일치하는 데이터만을 제거.
<code>event.dataTransfer.types</code>	dragstart 이벤트 발생시 DOM 목록에 있는 data format 을 설정하며 setData 함수를 호출할때 지정되는 format 문자열을 배열형식으로 얻을 수 있음.

12-4. 선택 텍스트의 드래그 포인터

드래그 중인 마우스 포인터를 변경하는 속성

<code>event.dataTransfer.effectAllowed</code>	드래그 앤 드롭으로 할 수 있는 조작을 나타내는 문자열을 반환. 값을 지정해서 허용 작업을 변경할 수 있음. none,copy,copyLink,copyMove,link,linkMove,move,all,uninitialized 중 하나 사용
<code>event.dataTransfer.dropEffect</code>	드롭될 때 처리 가능한 조작을 나타내는 문자열을 반환. 값을 지정하여 처리 가능한 조작을 변경할 수 있음. 지정 가능한 값은 none, copy, link, move. 값을 지정하면 해당 조작에 맞는 드래그 포인터가 표시

effectAllowed속성

값	의미
copy	선택 영역이 복사됨을 의미
link	선택 영역이 드롭 영역에 링크됨을 의미
move	선택 영역이 드롭 영역에 이동함을 의미
copyLink	선택 영역이 복사 또는 링크됨을 의미

copyMove	선택 영역이 복사 또는 이동함을 의미
linkMove	선택 영역이 링크 또는 이동함을 의미
all	모든 드롭 조작이 유효해짐
none	모든 드롭 조작이 무효화됨. 드롭 중에는 드롭 불가를 나타내는 커서가 표시
uninitialized	effectAllowed 속성에 아무것도 정의되지 않은 상태. 기본값

dropEffect 속성

값	의미
copy	복사를 나타내는 포인터가 표시
link	링크를 나타내는 포인터가 표시
move	이동을 나타내는 포인터가 표시
none	드롭 불가를 나타내는 포인터가 표시. 기본값

일반적으로 **effectAllowed** 속성은 드래그되는 요소에서 발생하는 **dragstart** 이벤트 리스너에서 지정한다. 그리고 **dropEffect** 속성은 드롭되는 요소에서 발생하는 **dragenter** 이벤트와 **dragover** 이벤트에서 사용.

effectAllowed 속성에서 허용하는 동작만 드롭 측에서 사용 가능. 따라서 **dropEffect** 속성에는 **effectAllowed** 속성에서 허용하는 작업을 나타내는 값을 지정.

이 값이 일치하지 않으면 **none**에 해당하는 드래그 포인터가 표시되므로 주의

13. 웹 워커

13-1. 웹 워커란?

웹 워커(Web Worker)는 자바스크립트의 처리를 백그라운드에서 실행하는 기능을 제공한다.

13-2. 워커의 사용

Worker 객체 만들기

<code>new Worker(scriptURL)</code>	매개변수로 워커용 스크립트의 URL을 지정하면 해당 스크립트의 Worker 객체를 반환하고 그 스크립트를 워커로 실행 예) <code>var worker = new Worker('worker.js');</code>
------------------------------------	---

워커에 메시지 전송

```
//워커에 메시지 보내기  
worker.postMessage('message');
```

부모 페이지에서 보내온 메시지 수신

```
//부모로부터 메시지가 왔을 때의 처리  
self.onmessage = function(event){  
  
//내용 처리  
  
//처리 결과를 부모에 메시지로 보내기  
self.postMessage(n);  
};
```

부모 페이지에서 워커에서 보낸 메시지 수신

```
worker.onmessage = function(event){  
  
};
```

Worker 객체의 메서드

메서드	설명
onerror	에러를 포착하기 위한 이벤트 핸들러
terminate	워커를 강제로 종료함
postMessage(message)	워커에 메시지를 보냄
onmessage	워커로부터의 메시지를 포착하기 위한 이벤트 핸들러

14. 애플리케이션 캐시

14-1. 오프라인 웹 애플리케이션

오프라인 웹 애플리케이션은 애플리케이션 캐시(application cache)를 이용해 동작하며 모든 자바스크립트, HTML, CSS뿐만 아니라 전체 웹사이트를 오프라인으로 저장할 수 있다. 저장된 데이터를 이용해 오프라인 상태에서도 웹 애플리케이션을 이용할 수 있다.

14-2. 오프라인에서 작동하도록 사이트 설정하기

- 1) `cache.manifest` 파일을 생성
- 2) 매니페이스(**manifest**)파일이 정확한 콘텐츠 타입으로 제공되는지 확인
- 3) 모든 **HTML** 파일이 캐시 매니페스트를 가리키게 함

cache.manifest 파일

CACHE MANIFEST

CACHE: ← 방문자의 하드디스크에 저장하고 싶은 파일 리스트

#버전 1.1 ← 주석

index.html

photo.jpg

main.js

NETWORK: ← 사용자가 온라인이 되면 다시 로드하게 되고 오프라인에서는 이용할 수 없음
service.jsp

오프라인에 저장하고 싶은 모든 파일은 일일이 열거해야 하지만 온라인시 사용할 수 있는 파일을 명시할 때는 와일드카드 *를 이용할 수 있음. 애플리케이션 캐시에 저장되지 않는 모든 파일이나 URL을 서버에서 불러와야 한다는 사실을 브라우저에 알려줌

NETWORK:

*

서버에 콘텐츠 타입 설정하기

아파치 웹 서버 사용시 `.htaccess` 파일에 다음과 같은 내용 추가

`AddType text/cache-manifest .manifest`

톰캣 사용시 `web.xml`에 다음과 같은 내용 추가

```
<mime-mapping>
  <extension>.manifest</extension>
  <mime-type>text/cache-manifest</mime-type>
</mime-mapping>
```

HTML에서 매니페스트 파일 가리키기

```
<!doctype html>
<html manifest="cache.manifest">
```

14-3. 오프라인 웹 애플리케이션 저장 공간의 제약

오프라인 웹 애플리케이션의 규격에는 애플리케이션 캐시에 대한 저장 공간의 제약이 정의되어 있지 않고 브라우저가 저장 공간의 제약을 만들고 그것을 적용해야 한다고 기술되어 있으나 일반적으로 이용 가능한 공간이 **5MB**를 넘지 않도록 한다.

14-4. 대체 수단 지정하기

리소드 로드 실패했을 때 사용자가 보게 될 내용을 정의할 수 있다.

himg/는 **himg** 폴더에 위치한 모든 파일을 의미. 두 번째 항목은 지정된 파일이 로드되는데 실패하는 경우 표시하고 싶은 것을 지정 (**test.html** 이 로드되지 않으면 **replace.html** 표시)

FALLBACK:

himg/ images/cinema.jpg (또는 **test.html replace.html**)

해당 사이트의 모든 페이지와 매치하는 **/**를 지정할 수 있다. 어떤 페이지가 로드되는 데 실패하거나 애플리케이션 캐시에 존재하지 않는다면 **offline.html** 페이지로 대체되어 표시

FALLBACK:

/ /offline.html