

Web



HTML/CSS/Javascript

- HTML은 마크업 언어(markup language)
 - 구조와 내용
- CSS는 스타일 언어(Style Language)
 - 모양, 디자인
- Javascript는 웹 개발 언어
 - 동적 기능 및 응용프로그램 작성

HTML

- HTML (Hypertext Markup Language)
- 웹 페이지를 위한 마크업 언어(구조를 설계할 때 사용되는 언어)라고 할 수 있습니다.
 - CSS 가 비주얼 디자인(Visual Design) 이라면 HTML 은 구조적 설계(Structure Design) 이라 할 수 있습니다.
- HTML5
 - 비표준 기술, 비호환성의 해결
 - 범용 웹 표준, 적용기기의 다양화

HTML

- HTML은 대소문자를 구분하지 않습니다.
 - <title>, <TITLE>, <Title>, <TiTlE> 같이 사용할 수 있습니다.
 - 하지만 가독성 등의 이유로 소문자로 작성합니다.
- XHTML(EXtensible HTML)
 - XML문서로 문법적으로 보다 엄격하게 적용

HTML 문서의 구조

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My test page</title>
  </head>
  <body>
    <p>This is my page</p>
  </body>
</html>
```



- <!DOCTYPE html>: 문서 형식
- <html> </html>: 전체 페이지의 콘텐츠를 포함하며, 기본 요소

HTML 버전

버전	년도	DOCTYPE 및 설명
HTML 1.0	1991	팀 버나스리(Tim Berners-Lee)가 발표한 최초의 HTML
HTML 2.0	1995	국제 표준으로 제정된 최초의 HTML
HTML 3.2	1997	W3C에 의해 제정된 최초의 HTML
HTML 4.01	1999	<code><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd"></code>
XHTML 1.0	2000	<code><!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"></code>
HTML 5	2014	<code><!DOCTYPE html></code>

HTML Browser Support

- 브라우저별로 지원하는 차이점이 존재
- 버전별 선언(HTML5, HTML4, XHTML)에 따라서 HTML 은 지원하는 태그가 조금씩 다릅니다.

					
<u><select></u>	Yes	Yes	Yes	Yes	Yes
<u>autofocus</u>	Yes	10.0	No	Yes	Yes
<u>disabled</u>	Yes	9.0	Yes	Yes	Yes
<u>form</u>	Yes	Yes	Yes	Yes	Yes
<u>multiple</u>	Yes	Yes	Yes	Yes	Yes
<u>name</u>	Yes	Yes	Yes	Yes	Yes
<u>required</u>	Yes	10.0	4.0	Yes	Yes
<u>size</u>	Yes	Yes	Yes	Yes	Yes

기본 구조

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML문서의 제목입니다.</title>
  </head>
  <body>
    <h1>제목 크기1입니다.</h1>
    <h2>제목 크기2입니다.</h2>
    <p>이 부분은 단락입니다.</p>
  </body>
</html>
```

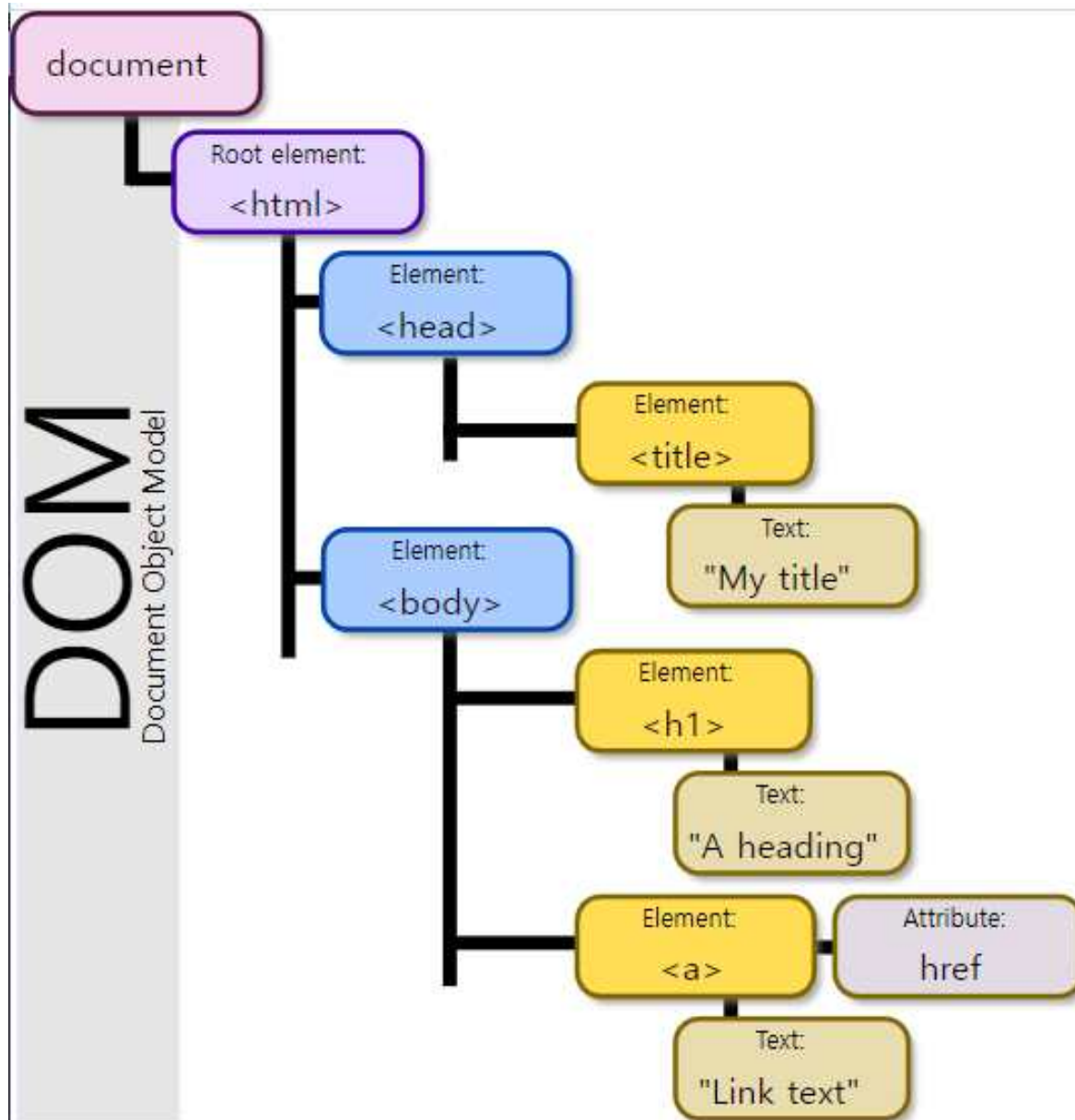

기본 구조

- <!DOCTYPE html>
 - 현재 문서가 HTML5 문서임을 명시합니다.
- <html>
 - HTML 문서의 루트(root) 요소를 정의합니다.
- <head>
 - HTML 문서의 메타데이터(metadata)를 정의합니다.
 - 메타데이터(metadata)
 - HTML 문서에 대한 정보(data)
 - 웹 브라우저에는 직접적으로 표현되지 않는 정보
 - <style>, <meta>, <link>, <script>, <base> 태그 등을 이용하여 표현할 수 있습니다.

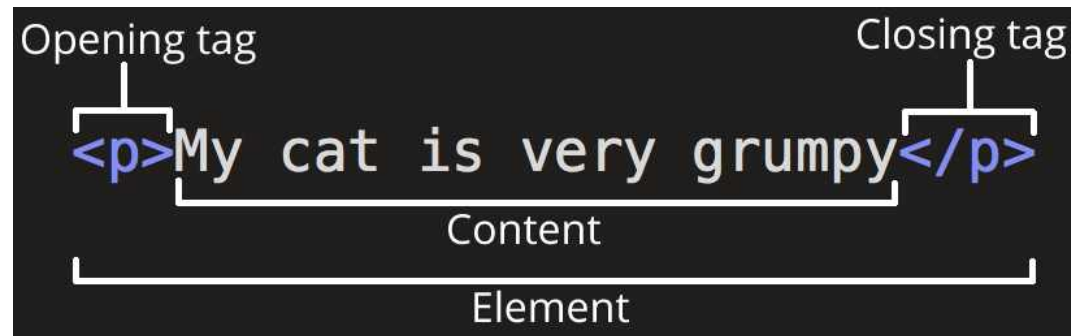
기본 구조

- <title> : HTML 문서의 제목(title)을 정의
 - 웹 브라우저의 툴바(toolbar)에 표시
- <body> : 웹 브라우저를 통해 보이는 내용(content) 부분입니다.
- <h1> ~ <h6> : 제목(heading)을 나타냅니다.
- <p> : 단락(paragraph)을 나타냅니다.

DOM



element



- 여는 태그(Opening tag)
 - 요소의 이름과(이 경우 p), 열고 닫는 꺾쇠 괄호로 구성
 - 요소가 시작(이 경우 단락의 시작 부분)부터 효과가 적용
- 닫는 태그(Closing tag)
 - 요소의 이름 앞에 슬래시(/)를 제외하면 여는 태그(opening tag)와 동일
 - 이것은 요소의 끝(이 경우 단락의 끝 부분)에 위치합니다.
 - 닫는 태그를 적어주어야 정상적으로 표시됩니다.

element

- 빈 태그(empty tag)
 - 닫는 태그(close tag)가 없는 element
 - contents를 감싸지 않아 비어있다는 의미
 - `<meta charset="utf-8">`
- 애트리뷰트(attribute) – 속성



A diagram illustrating an attribute in an HTML tag. The text `<p class="editor-note">My cat is very grumpy</p>` is shown on a dark background. A white bracket is drawn above the `class="editor-note"` part, with the word "Attribute" written above the bracket.

- 벨류(value) – 값
 - `<tagname attribute="value"> 내용 </tagname>`

빈 요소(Empty elements)

- 주로 문서에 무언가를 첨부하기 위해 단일 태그(Single tag)를 사용하는 요소도 있습니다.
 - 예를 들어 `` 요소는 해당 위치에 이미지를 삽입하기 위한 요소입니다.

```

```



Nesting elements

- 포함된 요소

```
<p>My cat is <strong>very</strong> grumpy.</p>
```

```
<p>My cat is <strong>very grumpy.</p> </strong>
```

- 중첩(overlap 또는 stacking)된 문장은 잘못된 문장
- 잘못 된 내포에도 불구하고 어떤 브라우저에서 이상 없이 보일 수도 있습니다. 그 것은 해당 브라우저가 문법에 흐물흐물(permittable)해서 웬만한 오류는 자체 수정해서 브라우저 창에 rendering해주기 때문일 뿐 입니다. '웬만한 오류는 알아서 처리해 준다.' 는 부분에서 특히 주의

속성(Attributes)

Attribute



```
<p class="editor-note">My cat is very grumpy</p>
```

- 속성은 요소에서 표시되지 않는 내용
 - 스타일에 관련된 내용이나 기타 내용을 위해 해당 목표를 구분할 수 있는 class 속성이 있으며 이외에 id, name 등이 있습니다.
 - 요소 이름 다음에 바로 오는 속성은 요소 이름과 속성 사이에 공백
 - 하나 이상의 속성들이 있는 경우엔 속성 사이에 공백
 - 속성 이름 다음엔 등호(=)가 붙습니다.
 - 속성 값은 열고 닫는 따옴표로 감싸야 합니다.
 - 소문자 작성 권장

참과 거짓 속성(Boolean attributes)

- 때때로 값이 없는 속성을 볼 수 있는데 이를 불 속성이라고 하며, 일반적으로 그 속성의 이름과 동일한 하나의 값만을 가질 수 있습니다.
 - 예를 들어 disabled 속성을 양식 입력 요소에 할당하면 사용자가 데이터를 입력할 수 없도록 비활성화(회색으로 표시) 할 수 있습니다.

```
<input type="text" disabled="disabled">
```

```
<input type="text" disabled>
```

```
<input type="text">
```



The image shows two text input fields side-by-side. The first field is disabled, indicated by its gray background and lack of focus. The second field is active, with a white background and a thin border.

작은 따옴표, 큰 따옴표

- 둘 다 사용 가능

```
<a href="http://www.example.com">A link to my example.</a>
```

```
<a href='http://www.example.com'>A link to my example.</a>
```

- 섞어 사용 불가

```
<a href="http://www.example.com">A link to my example.</a>
```

```
<a href='http://www.example.com'>A link to my example.</a>
```

- 따옴표 안에서 따옴표 혹은 특수문자의 경우
 - 별도의 코드를 사용하여 표시합니다.

HTML 문자코드

엔티티(entity) : 키보드 입력이 불가능하거나 HTML 예약어를 기존에 사용하던 의미 그대로 사용하기 위해 별도로 만든 문자셋

[HTML 특수문자 리스트](#) [표준안](#)

기호	숫자코드	문자코드	설명
"	"	"	큰따옴표
'	'	'	작은 따옴표
<	<	<	이하
>	>	>	이상
@	@		골뱅이 / 엣사인
[[왼쪽 대괄호
₩	\		백슬러시
]]		오른쪽 대괄호
&	&	&	앰퍼센트
•	•	•	점

HTML Tags

- <!-->, <!DOCTYPE>, <a>, <abbr>, <acronym>, <address>, <applet>, <area>, <article>, <aside>, <audio>, , <base>, <basefont>, <bdi>, <bdo>, <big>, <blockquote>, <body>,
, <button>, <canvas>, <caption>, <center>, <cite>, <code>, <col>, <colgroup>, <data>, <datalist>, <dd>, , <details>, <dfn>, <dialog>, <dir>, <div>, <dl>, <dt>, , <embed>, <fieldset>, <figcaption>, <figure>, , <footer>, <form>, <frame>, <frameset>, <h1> - <h6>, <head>, <header>, <hr>, <html>, <i>, <iframe>, , <input>, <ins>, <kbd>, <label>, <legend>, , <link>, <main>, <map>, <mark>, <meta>, <meter>, <nav>, <noframes>, <noscript>, <object>, , <optgroup>, <option>, <output>, <p>, <param>, <picture>, <pre>, <progress>, <q>, <rp>, <rt>, <ruby>, <s>, <samp>, <script>, <section>, <select>, <small>, <source>, , <strike>, , <style>, <sub>, <summary>, <sup>, <svg>, <table>, <tbody>, <td>, <template>, <textarea>, <tfoot>, <th>, <thead>, <time>, <title>, <tr>, <track>, <tt>, <u>, , <var>, <video>, <wbr>.....

제목(Heading)

- `<h1>~<h6></h1>~<h6>`
- title 속성 > tool tip

단락(Paragraph)

- `<p> </p>`
- `
`
 - break line 줄바꿈, 내려쓰기
- `<pre> </pre>`
 - preformatted text
 - 텍스트 그대로
- `<hr>`
 - horizontal rule 수평가로선

서식

- 강조
 - `` ``, `` `` 중요
 - `` ``, `<i>` `</i>`
- 크기
 - `<small>` `</small>`
- 취소 추가
 - `` `` 취소줄표시
 - `<ins>` `</ins>` 밑줄
- 위 아래 첨자
 - `^{` `}`, `_{` `}`
- 하이라이팅
 - `<mark>` `</mark>`
- 추가적인 CSS를 통해 특성 추가 가능

인용구

- 짧은 인용구
 - `<q> </q>`
 - 앞뒤 큰 따옴표
- 블록인용구
 - `<blockquote> </blockquote>`
 - 별도의 단락 처리

주석(Comment)

- `<!-- 주석내용 -->`
- 개발자가 작성한 해당 코드에 대한 이해를 돕는 설명이나 디버깅을 위해 작성한 구문
- 브라우저는 주석을 무시
- 주석의 목적은 코드에 메모를 포함시켜 논리 또는 코딩을 설명 할 수 있도록 하는 것입니다.
- 코드베이스로 다시 분석할 때 매우 유용합니다.

```
<p>I'm not inside a comment</p>
```

```
<!-- <p>I am!</p> -->
```

요소의 타입

- 블록 레벨 요소(Block-level elements)
 - 앞뒤 요소 사이에 새로운 줄(Line)을 생성
 - 전과 이후 요소 사이의 줄바꿈
 - 일반적으로 페이지의 구조적 요소를 표현
 - 단락(Paragraphs), 목록(lists), 네비게이션 메뉴(Navigation Menus), 꼬리말(Footers) 등
 - 블록 레벨 요소는 인라인 요소(Inline elements)에 중첩(Nested inside)불가
 - 블록 레벨 요소는 다른 블록 레벨 요소에 중첩가능
- 인라인 요소(Inline elements)
 - 항상 블록 레벨 요소내에 포함
 - 새로운 줄(Line)을 생성하지 않음
 - 하이퍼링크를 정의하는 요소인 <a> , 텍스트(Text)를 강조하는 요소인 , 등

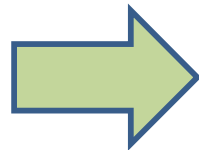
요소의 타입

- <div> elements
 - 다른 요소를 묶는데 자주 사용되는 대표적인 블록 요소
 - 주로 요소들의 스타일을 한 번에 적용하기 위해 사용
- element
 - 텍스트의 특정 부분을 묶는데 자주 사용되는 인라인 요소
 - 특정부분에 따로 스타일을 적용하는데 사용

구조적인 markup tags

- 콘텐츠를 보다 읽기 쉽게 보다 편리하도록 구조화

My short story I am a policewoman and my name is Trish. My legs are made of cardboard and I am married to a fish.



My short story

I am a policewoman and my name is Trish.

My legs are made of cardboard and I am married to a fish.


- `<h1>~<h6>, <p>`
- CSS 등 attribute를 통해 구조를 형성할 수도 있다.
``
Is this a top level heading?
``

메타 데이터

- 데이터를 설명하는 데이터
- <meta>
- <title>
- <style>
- <script>
- <link>
- <base>

<meta>

- HTML document 의 메타데이터 표시
- 항상 <head> 요소 안에
- 화면에 보이지는 않음
- 브라우저, 검색엔진, 다른 웹서비스에서 사용

속성명	속성값	설명
charset	 문자셋	해당 문서의 문자 인코딩 방식을 명시함.
content	텍스트	name 속성이나 http-equiv 속성과 관련된 값(value)을 명시함.
http-equiv	content-type default-style refresh	content 속성에 명시된 값에 대한 HTTP 헤더를 제공함.
name	application-name author description generator keywords viewport	메타데이터를 위한 이름을 명시함.
scheme	포맷 또는 URI	content 속성의 속성값을 해석하는데 사용할 스키마(scheme)를 명시함. HTML5에서는 더 이상 지원하지 않음.

참고

– HTML4에서 UTF-8의 경우

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

– HTML5에서 UTF-8의 경우

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

<style>

- HTML 문서의 스타일 정보를 정의
- 해당 요소가 포함된 HTML 문서의 콘텐츠에 적용되는 CSS를 명시
- HTML 문서는 여러 개의 <style> 요소를 포함
- 속성에 값을 주어 해당 태그의 보여지는 형태를 결정한다.
 - 테이블의 경우 표의 선굵기, 형태, 크기, 간격 등
 - 각 개체간의 간격, 위치, 표시여부 등

CSS 적용 방법

- CSS(Cascading Style Sheets)
 - HTML 요소들이 각종 미디어에서 어떻게 보이는가를 정의하는 데 사용되는 스타일 시트 언어
- Inline style
 - 요소 내부에 style 속성을 사용하여 CSS 스타일을 적용
 - 해당 요소에만 스타일 적용
- Internal style sheet
 - HTML 문서의 <head>태그 내에 <style>태그를 사용하여 스타일을 지정
- External style sheet
 - 웹 사이트 전체의 스타일을 하나의 파일에서 변경
 - <head>태그 내에 <link>태그를 사용하여 외부 스타일 시트를 포함
- 적용 우선순위
 1. 인라인
 2. 내부/외부 스타일 시트
 3. 웹 브라우저 기본 스타일

CSS 적용 방법

인라인 스타일

```
<p style="color:green; text-decoration:underline">  
인라인 스타일을 이용하여 스타일을 적용하였습니다.  
</p>
```

내부 스타일 시트

```
<style>  
  body { background-color: lightyellow; }  
  p { color: red; text-decoration: underline; }  
</style>
```

외부 스타일 시트

```
<link rel="stylesheet" href="/examples/media/style.css">
```

style.css


```
body { background-color: lightyellow; }  
p { color: red; text-decoration: underline; }
```

CSS

- CSS > > > >

<script>

- 자바스크립트와 같은 클라이언트 사이드 스크립트(client-side scripts)를 정의
- 스크립트 코드를 요소 내부에 직접 명시하거나, src 속성을 사용하여 외부 스크립트 파일을 참조 가능
- src 속성이 명시된 <script> 요소에는 스크립트 코드를 직접 명시 불가

속성명	속성값	설명
async	 async	스크립트가 비동기적으로 실행됨을 명시함. (단, <script> 요소가 외부 스크립트를 참조하는 경우에만 사용할 수 있음)
charset	문자셋	외부 스크립트 파일에서 사용되는 문자 인코딩 방식을 명시함. (단, <script> 요소가 외부 스크립트를 참조하는 경우에만 사용할 수 있음)
defer	defer	페이지의 파싱이 모두 끝나면 스크립트가 실행됨을 명시함. (단, <script> 요소가 외부 스크립트를 참조하는 경우에만 사용할 수 있음)
src	URL	외부 스크립트 파일의 URL을 명시함.
type	미디어 타입	스크립트의 미디어 타입을 명시함.

스크립트 실행

- <script>태그가 있는 웹사이트 로드
 1. HTML 페이지 가져오기
 2. HTML 구문 분석 시작
 3. 파서는 외부 스크립트 파일을 참조하는 <script> 태그를 발견
 4. 브라우저는 스크립트 파일을 요청
 1. 파서는 페이지의 다른 HTML 구문 분석을 차단하고 중지
 5. 잠시 후 스크립트가 다운로드되어 실행
 6. 파서는 HTML 문서의 나머지 부분을 계속 파싱
- 스크립트 실행 전까지 로드 중지 때문에 다른 로딩 지연 혹은 화면 기능 미실행의 문제

스크립트 실행

- async
 - 비동기
 - 즉시 JavaScript가 다운로드되고 다운로드 후에는 HTML 파싱과 함께 비동기(병렬)로 실행
 - `<script src="s1.js" async> </script> <script src="s2.js" async> </script>`
 - s1이 s2보다 먼저 다운로드 실행 가능
- defer
 - 연기하다
 - JavaScript는 HTML 파싱과 병렬로 다운로드되지만 전체 HTML 파싱이 완료된 후에만 실행
 - `<script src="s1.js" defer> </script> <script src="2.js" defer> </script>`
- 주의
 - `<script src="jquery.js" async> </script> <script>jQuery(something);</script>`
 - *<!-- * might throw "jQuery is not defined" error * defer will not work either -->*

Javascript

- 자바스크립트 >>>>

<base>

- 해당 문서의 모든 상대주소(relative URL)에 대한 기본 URL(base URL)과 target 속성값을 정의
- <head>요소 내에 위치
- 언제나 href 속성이나 target 속성 중 하나 또는 둘 모두를 명시

속성명	속성값	설명
href	URL	페이지 내의 모든 상대 주소(relative URL)들의 기준 URL을 명시함.
target	_blank _parent _self _top 프레임 이름	페이지 내의 모든 하이퍼링크와 폼(form)을 위한 기본 대상(default target)을 명시함.

<link>

- 해당 문서와 외부 소스(external resource) 사이의 관계를 정의할 때 사용
- 빈 태그, 속성만을 포함
- <head> 요소 내부에만 위치
- 개수에는 제한이 없음
- 주로 외부 스타일 시트(external style sheet)를 연결할 때 사용

```
<link rel="stylesheet" type="text/css" href="/example/media/style.css">
```

HTML 링크

- hyperlink
- `HTML 링크`
 - href 링크 클릭시 연결할 페이지나 사이트의 URL주소를 명시

target 속성값	설명
_blank	링크로 연결된 문서를 새 창이나 새 탭에서 오픈.
_self	링크로 연결된 문서를 현재 프레임(frame)에서 오픈. (기본 설정)
_parent	링크로 연결된 문서를 부모 프레임(frame)에서 오픈.
_top	링크로 연결된 문서를 현재 창의 가장 상위 프레임(frame)에서 오픈.
프레임(frame) 이름	링크로 연결된 문서를 지정된 프레임(frame)에서 오픈.

하이퍼링크(hyperlink)

- 웹의 특성으로 문서간의 연결, 다른 사이트로의 이동 혹은 활성화

```
<p>I'm creating a link to  
<a href="https://www.mozilla.org/en-US/"  
  title="The best place to find more information about Mozilla's  
    mission and how to contribute">the Mozilla homepage</a>.  
</p>
```

- 다양한 요소들을 모두 link로 적용 가능

```
<a href="https://www.mozilla.org/en-US/">  
    
</a>
```

절대 URL과 상대 URL

- **절대 URL**

- 웹에서 정의된 상대적인 위치를 가리킵니다. (protocol , domain name 포함)
- 절대 URL은 어디에 사용되든 항상 같은 장소를 가리킵니다.

- **상대 URL**

- 당신이 link하고 있는 파일로부터 상대적인 위치를 가리킵니다.
- 상대 URL은 파일의 실제 위치가 어디냐에 따라 다른 장소를 가리킬 것입니다.

■ index.html 파일을 projects 디렉토리에서 웹사이트의 root로 옮긴다면

pdfs/project-brief.pdf 의 상대 URL은

<http://www.example.com/pdfs/project-brief.pdf>

<http://www.example.com/projects/pdfs/project-brief.pdf>를 가리키지 않습니다.

■ index.html 파일을 옮겼다고 해서 project-brief.pdf 파일과 pdfs 폴더의 위치가 갑자기 바뀌지는 않을 것입니다- 이것은 당신의 링크가 잘못된 곳을 **주의!**

링크의 상태(state)

링크의 상태	설명
link	아직 한 번도 방문한 적이 없는 상태 (기본설정)
visited	한 번이라도 방문한 적이 있는 상태
hover	링크 위에 마우스를 올려놓은 상태
active	링크를 마우스로 누르고 있는 상태

HTML 이미지(image)

- 이미지 삽입

- src : 저장된 주소의 URL 주소 명시

- alt : 로딩되지 않는 경우 대신 표시할 문자열

- style 속성

- width/height : 너비와 높이를 픽셀 단위로 설정

HTML 목록(List)

- 순서없이

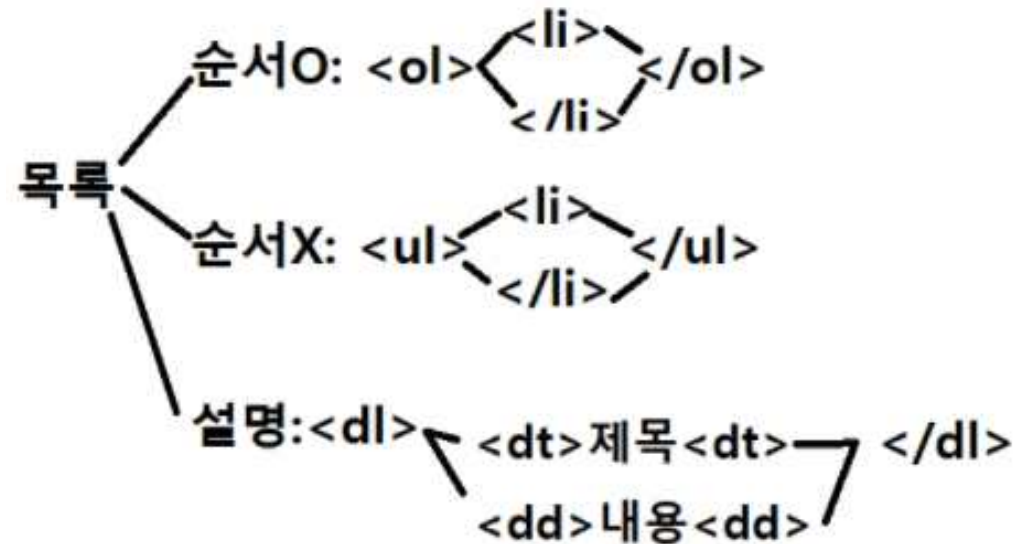
```
<ul>
  <li>milk</li>
  <li>eggs</li>
  <li>bread</li>
  <li>hummus</li>
</ul>
```

- milk
- eggs
- bread
- hummus

- 순서대로

```
<ol>
  <li>Drive to the end of the road</li>
  <li>Turn right</li>
  <li>Go straight across the first two roundabouts</li>
  <li>Turn left at the third roundabout</li>
  <li>The school is on your right, 300 meters up the road</li>
</ol>
```

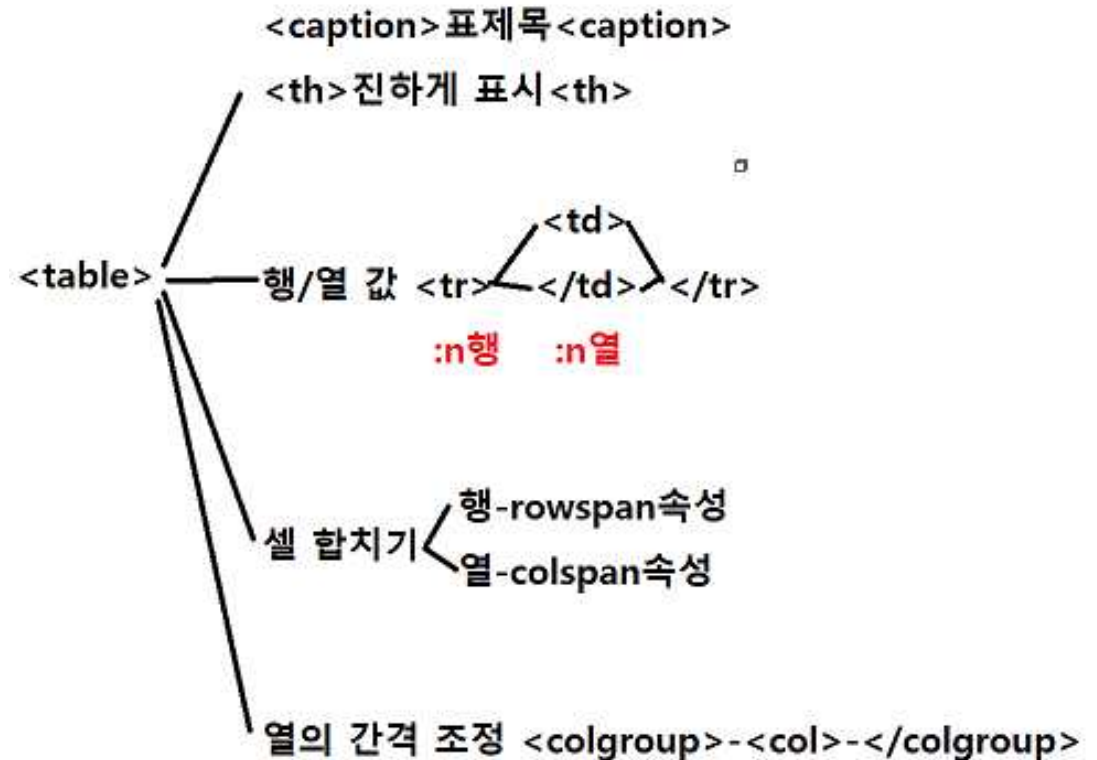
1. Drive to the end of the road
2. Turn right
3. Go straight across the first two roundabouts
4. Turn left at the third roundabout
5. The school is on your right, 300 meters up the road



HTML 표(table)

```
<table>
  <thead>
    <tr>
      <th>Month</th>
      <th>Savings</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>January</td>
      <td>$100</td>
    </tr>
    <tr>
      <td>February</td>
      <td>$80</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td>Sum</td>
      <td>$180</td>
    </tr>
  </tfoot>
</table>
```

Month	Savings
January	\$100
February	\$80
Sum	\$180



- `<table> </table>`
- `<tr> </tr> <td> </td>`
- `<thead> </thead>`
- `<tbody> </tbody>`
- `<tfoot> </tfoot>`

표의 열과 행 합치기, 캡션

- colspan
- rowspan
- caption

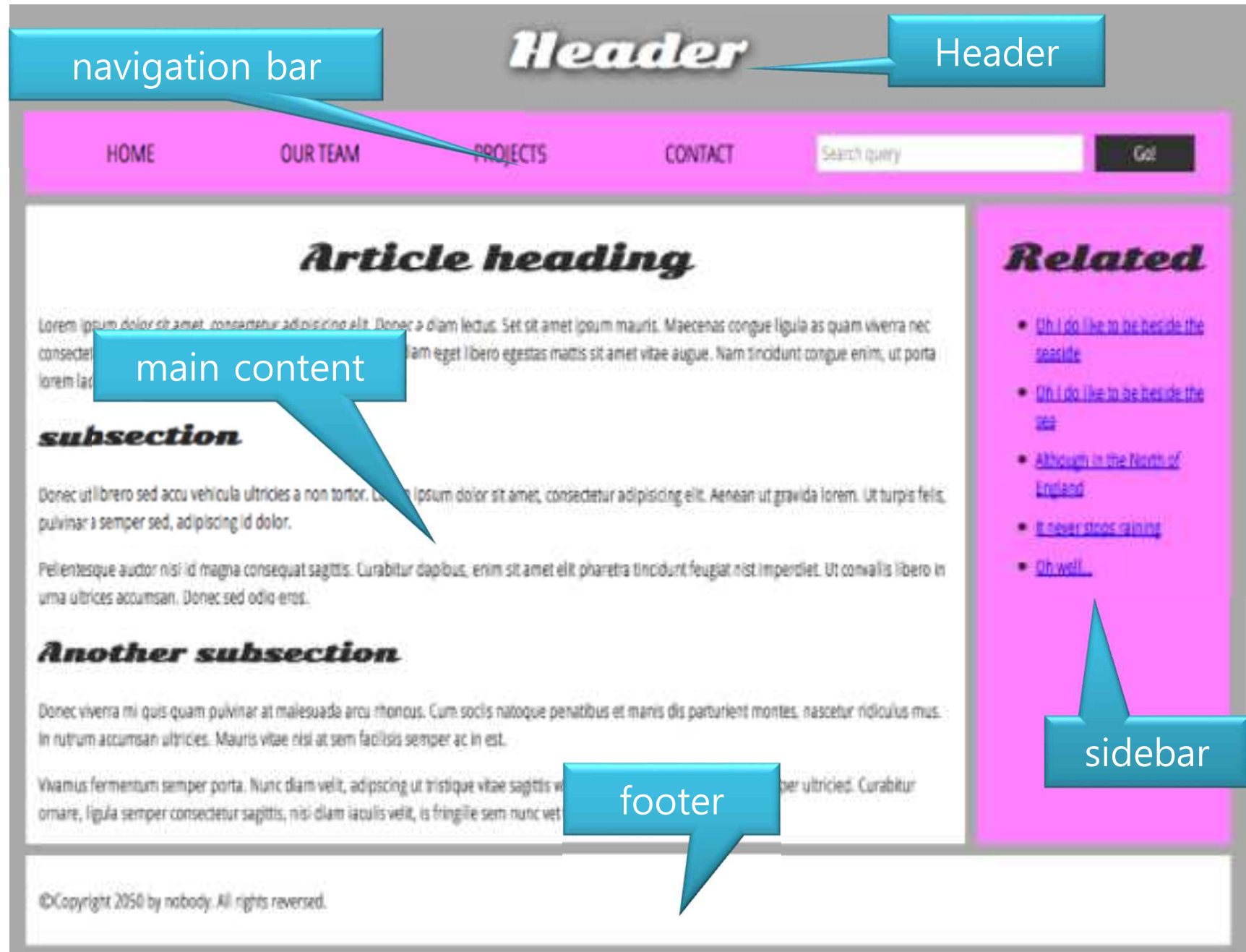
해양 생물

참치	고래	날치
상어	문어	꽁치
오징어	고등어	돌고래
참치	고래	
	구래	고래

HTML 공간분할

- 블록/인라인
- iframe
- layout

웹사이트의 구조 예



웹사이트의 구조 기본 섹션

- header
 - 일반적으로 큰 제목과 로고 등이 있는 큰 띠
- navigation bar
 - 홈 페이지의 메인 섹션으로 연결
 - 대부분 메뉴 버튼이나 링크, 탭으로 표현
 - 헤더와 같이, 이 항목은 페이지 이동 후에도 유지
- main content
 - 웹 페이지에서 당신이 원하는 메인 콘텐츠를 포함하고 있는 부분
- sidebar
 - 주변의 정보, 링크, 인용 부호, 광고 등
 - 일반적으로 메인콘텐츠와 연동된 정보 표시
 - 보조 navigation system으로 사용
- footer
 - 페이지 바닥의 줄로 일반적으로 작은 정보, 저작권 정보, 또는 연락처 등을 포함
 - (header와 같이) 일반적인 정보를 담고 있는 부분이지만 보통 중요하지 않거나 웹 사이트 자체에 부차적인 정보

웹사이트의 구조화 HTML

- header
 - <header>
- navigation bar
 - <nav>
- main content
 - <main>
 - <article>, <section>, and <div> elements.
- sidebar
 - <aside> (자주 <main> 안에 만들어진다.)
- footer
 - <footer>

HTML layout

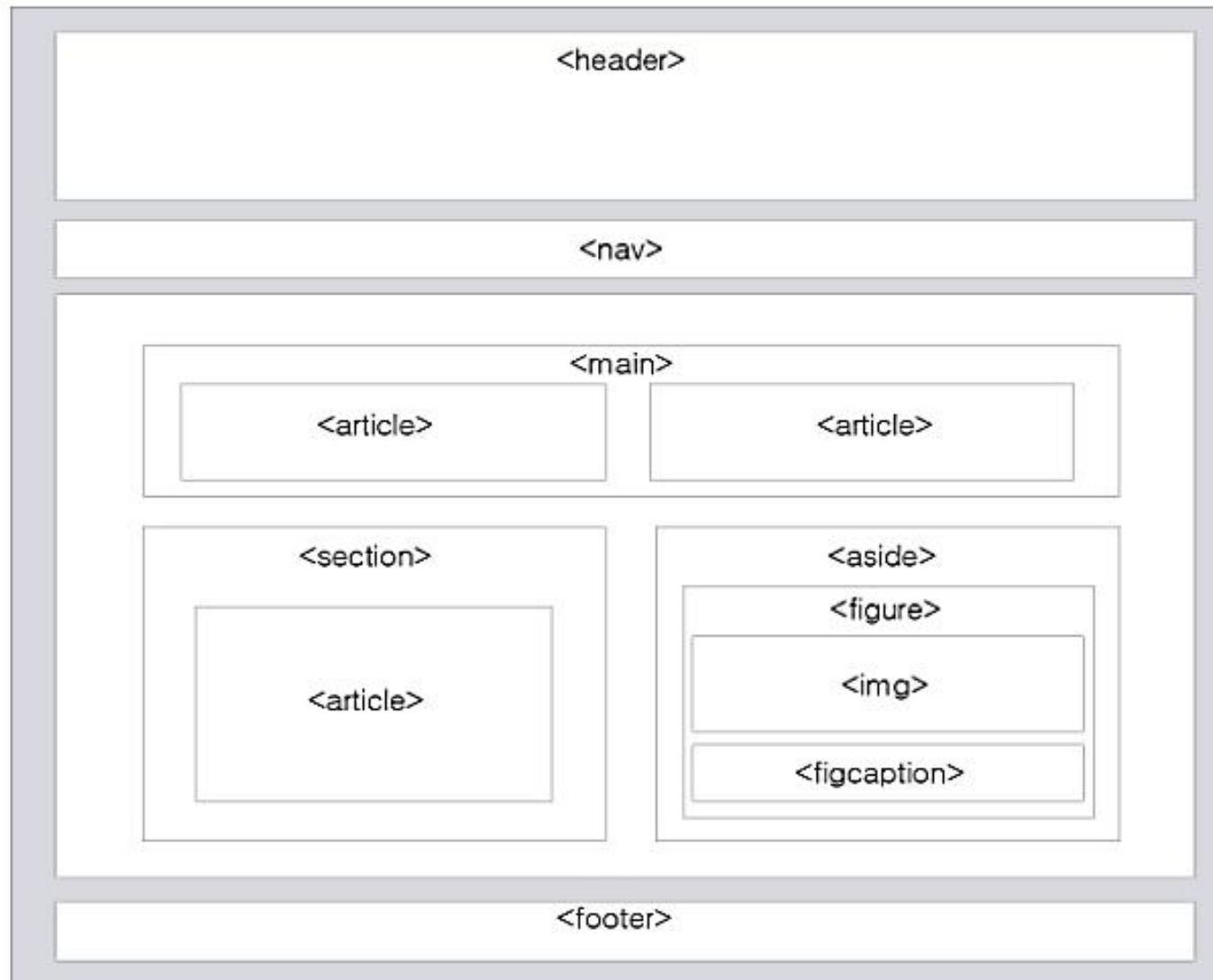
- 레이아웃
 - 특정 공간에 여러 구성 요소를 보기 좋게 효과적으로 배치하는 작업
- 1. div 요소를 이용한 레이아웃
- 2. HTML5 레이아웃
- 3. table 요소를 이용한 레이아웃

div 요소를 이용한 레이아웃

- 각 구조화 대상들을 div로 묶어 float 속성을 사용하여 원하는 형태를 구성
- float 속성 >>> CSS

HTML5 레이아웃

- HTML5에서는 웹 페이지의 레이아웃만을 위한 의미 (semantic) 요소 제공



의미(semantic) 요소

- non-semantic elements
 - <div>,
 - 내용물에 대한 정보 전달 없음
- semantic elements
 - 웹사이트의 정보를 명확히 하여 검색에 대응하는 인덱싱을 용이하게 함
 - <article> <aside> <detail> <figcaption>
<figure> <footer> <header> <main>
<mark> <nav> <section> <summary>
<time>

HTML5의 Semantic Elements

태그	설명
<article>	글을 의미합니다.
<aside>	페이지의 내용 주변에 있는 내용을 의미합니다.
<details>	사용자가 볼 수 있거나 숨겨진 추가적인 세부사항을 의미합니다.
<figcaption>	<figure>의 설명을 의미합니다.
<figure>	일러스트, 도형, 사진, 코드같은 그 자체로 의미를 갖는 내용을 의미합니다.
<footer>	문서나 섹션의 꼬릿말을 의미합니다.
<header>	문서나 섹션의 머릿말을 의미합니다.
<main>	문서의 주된 내용을 의미합니다.
<mark>	마크되거나 하이라이트된 글을 의미합니다
<nav>	네비게이션 링크를 의미합니다.
<section>	문서의 섹션을 의미합니다.
<summary>	<details>를 위한 가시적인 제목을 의미합니다.
<time>	날짜나 시간을 의미합니다.

table 요소를 이용한 레이아웃

- 테이블의 형태를 통해 원하는 영역의 위치를 테이블의 칸으로 구현
- HTML layout 적용 기술
 - CSS framework
 - CSS float property
 - CSS flexbox
 - CSS grid

HTML 입력

- form element
- input element

form element

- `<form action="주소" method="get|post"> </form>`
 - 웹 페이지에서의 입력 양식
- action 속성
 - 입력 데이터를 처리할 서버상의 파일의 주소 명시
 - 폼 핸들러(form-handler)
- method 속성
 - GET 방식
 - url뒤에 쿼리스트링(query string) 형태로 form data의 name과 value가 주소창에 노출
 - 한번 전송 시 데이터 양은 (url+쿼리 스트링의 글자 수) ≤ 255 자 이내로 제한
 - 검색 엔진의 쿼리(query)와 같이 크기가 작고 중요도가 낮은 정보를 보낼 때 주로 사용
 - POST 방식
 - form data가 request body안에 들어간 채 전송
 - 전송할 수 있는 데이터의 크기 또한 제한이 없음
 - 보안과는 관계 없음(별도의 보안솔루션 필요)
- name 속성
 - 스크립트나 서버에서 폼을 식별하기 위해 사용하는 폼의 이름
- `enctype='multipart/form-data'`
 - form안에 파일을 첨부

form element

역할	태그
요소들을 묶어주는 태그	<fieldset> </fieldset>
<fieldset>으로 묶은 태그의 제목	<legend>
입력 폼 이름	<label> </label>
입력 칸	<input>

input element

- <input>
 - type: 입력 양식
 - name: 입력한 값의 key값
 - 폼 태그에 들어가는 <input>이나 <label> 같은 태그들은 꼭 **<form></form> 태그 안에 위치**
 - 입력 폼을 필수 선택지로 만들고 싶다면 checked옵션을 주면 됩니다.
 - 추가로 placeholder="힌트" 속성으로는 입력 폼에 열은 글씨로 010-xxxx-xxxx 같은 형식에 대한 힌트를 줄 수 있습니다.

HTML id, class, name

id	class	name
하나의 요소만 가능 (페이지에서 유일)	여러 요소에 적용가능	여러 요소에 적용가능
CSS에서 식별자 사용 #아이디	CSS에서 식별자 사용	CSS에서 식별자 사용불가

- id와 class는 숫자로 시작 불가
- class는 중복 사용 가능, 여러 개의 값을 사용할 경우 띄어쓰기로 구분 사용
- name은 form 태그에서 submit 전송시 사용

input의 타입

- 텍스트 입력(text)
 - `<input type="text">`
 - 사용자에게 한 줄의 텍스트를 입력받음
- 비밀번호 입력(password)
 - `<input type="password" >`
 - 사용자에게 입력 받을 때 입력받은 문자나 숫자 대신 별표나 작은 원모양이 표시됨
- 문장 입력(textarea)
 - `<textarea name="message" rows="5" cols="30">여기에 적으세요</textarea>`
 - 사용자에게 여러 줄의 텍스트 입력받음
 - row속성/cols속성 : textarea의 크기를 지정

input의 타입

- 라디오 버튼(radio)
 - `<input type="radio" name="lecture" value="css">CSS
`
`<input type="radio" name="lecture" value="html" checked>HTML`
 - 여러 개의 옵션중에서 단 하나의 옵션만 입력 가능
 - 모든 input 요소의 name 속성이 동일해야 함
 - checked 속성 : 미리 선택되는 옵션 지정
- 체크박스(checkbox)
 - `<input type="checkbox" name="lecture" value="html" checked> HTML
`
`<input type="checkbox" name="lecture" value="css">`
 - 여러 개의 옵션 중에서 다수의 옵션을 입력 가능
 - 모든 input 요소의 name속성이 동일해야 함
 - checked 속성: 처음에 미리 선택되는 옵션 지정
 - disabled 속성: 해당 옵션을 선택할 수 없게 설정
- 선택 입력(select)
 - `<select name="fruit">`
`<option value="apple"> 사과`
`<option value="orange" selected> 귤`
`<option value="peach"> 복숭아`
`</select>`
 - 여러 개의 옵션이 드롭다운 리스트(drop-down list)로 표시
 - 하나의 옵션만 입력 가능

input의 타입

- 버튼 입력(button)
 - `<button type="button" onclick="alert('버튼을 클릭하셨습니다!')">버튼을 눌러주세요.</button>`
 - 사용자에게 버튼을 표시
- 전송 버튼(submit)
 - `<form action="#"> <input type="text"...>
 <input type="submit" value="전송"> </form>`
 - 입력받은 데이터를 서버의 폼 핸들러로 제출하는 버튼
- 파일 선택(file)
 - `<input type="file" name="upload_file" accept="image/*">`
 - 사용자에게 파일을 전송받음
 - accept 속성: 입력받을 수 있는 파일의 확장자 및 종류 명시

HTML5에서 추가된 타입

- datalist

```
<input list="lectures" name="lecture">  
  <datalist id="lectures">  
    <option value="HTML">  
    <option value="CSS">  
    <option value="JAVA">  
    <option value="C++">  
  </datalist>
```

- input 요소에 미리 정의된 옵션 리스트를 명시해 주는 요소
- 텍스트를 입력 가능, 드롭다운 메뉴에서 선택도 가능

HTML5에서 추가된 타입

- keygen
 - 사용자가 인증할 수 있는 안전한 방법을 제공
 - keygen 요소는 사용자가 입력한 데이터를 암호화하여 서버로 전송
 - 생성된 키(key)를 가지고 서버는 해당 사용자의 인증을 수행합니다.
 - 해당 기능을 제공하지 않는 브라우저 및 별도의 보안 요소로 인해 잘 사용되지 않음.

```
<form action="...">  
  사용자 : <br>  
  <input type="text" name="username"> <br>  
  암호화방법 : <br>  
  <keygen name="security"> <br>  
  <input type="submit" value="전송">  
</form>
```

HTML5에서 추가된 타입

- output

- 스크립트 등에 의해 수행된 계산의 결과나 사용자의 액션에 의한 결과를 나타낼 때 사용

```
<form action="#"
  oninput="result.value=parseInt(a.value)+parseInt(b.value)">
  <input type="number" id="a" name="a" value="20">
    +
    ( 0 <input type="range" id="b" name="b" value="0"> 100 )
    =
  <output name="result" for="a b"> </output> <br>
  <input type="submit">
</form>
```

input 일반적인 제한 속성

Attribute	Description
disabled	입력필드를 비활성화
max	입력의 최대값 설정
maxlength	입력필드의 최대 문자 개수 설정
min	입력의 최소값 설정
readonly	입력 필드의 값을 읽는 것만 가능 (변경할 수 없음)
required	비워둘 수 없음(반드시 입력)
size	<input> 요소의 너비를 문자수(in characters) 단위로 명시함. 기본값은 20
step	<input> 요소에 입력할 수 있는 숫자들 사이의 간격을 명시함. 기본값은 1
value	<input> 요소의 초기값(value)을 명시 - "button", "reset", "submit" : 버튼 내의 텍스트 - "hidden", "password", "text" : 입력 필드의 초기값 - "checkbox", "image", "radio" : 전송할 값을 정의

HTML5에서 추가된 form/input 요소의 속성

- form
 - autocomplete
 - novalidate
- input
 - autocomplete
 - autofocus
 - form
 - formaction
 - formenctype
 - formmethod
 - formnovalidate
 - formtarget
 - height and width
 - list
 - min and max
 - multiple
 - pattern (정규식)
 - placeholder
 - required
 - step

추가된 속성

- autocomplete 속성
 - form 요소나 input 요소에 입력된 정보의 저장여부
 - 속성값이 on으로 설정되면, 브라우저는 사용자가 입력하는 정보를 자동으로 저장
 - 이 후에 입력되는 입력값을 저장된 정보를 바탕으로 자동 완성
 - 다음과 같은 input 타입에서만 사용 가능
 - text, password, range, color, date, datetime-local, month, week, email, url, tel, search 타입
- novalidate 속성
 - input 요소의 속성이 아닌 form 요소의 속성
 - url 타입이나 email 타입과 같이 자동으로 유효성 검사하는 input 타입에 이 속성을 사용하면 유효성 검사 실시하지 않음

추가된 속성

- multiple 속성
 - 사용자가 input 요소에 값을 두 개 이상 입력하는 것을 허용
 - email 타입과 file 타입에서만 사용
- pattern 속성
 - input 요소에 입력된 값을 검사하기 위한 정규표현식(regular expression)을 명시
 - 정규 표현식이란 문자열에서 특정한 규칙을 가지는 문자열의 집합을 찾아내기 위한 검색 패턴
 - <https://regexr.com/>

HTML 5

- HTML 언어의 최신 표준 권고안
- [HTML5](#)
- 변경사항
 - 의미(semantic) 요소
 - <header>, <nav>, <main>, <section>, <aside>, <article>, <footer>, <figure>
 - 멀티미디어 요소
 - <video>, <audio>
 - 그래픽 요소
 - <canvas>, <svg>
 - input 요소의 타입
 - number, date, time, calendar, range

HTML5에서 추가된 자바스크립트 API

- Geolocation
- Drag and Drop
- Web Storage
- Web Worker

멀티미디어

- HTML5 이전까지는 웹 브라우저마다 어떤 종류의 멀티미디어 파일을 지원할지 각자 다른 방식으로 처리
- HTML5에서는 플래시와 같은 외부 플러그인의 도움 없이도 멀티미디어 파일을 간단히 사용 가능
- 웹 브라우저는 파일의 타입(type)을 파일의 확장자로 판단
- 비디오(video)나 사운드(sound)와 같은 멀티미디어 요소들은 멀티미디어 파일에 저장

비디오(video) 파일 형식

- HTML5 표준이 공식적으로 지원하는 비디오 파일 형식은 MP4, WebM, OGV 뿐

파일 형식	파일 확장자	설명
MPEG	.mpg .mpeg	Moving Picture Experts Group에 의해 개발되었으며, 변환 코덱을 이용하는 손실 압축 방식을 사용함.
MP4	.mp4	Moving Picture Experts Group에 의해 개발되었으며, 적은 용량으로도 고품질의 영상 및 음성을 구현할 수 있어서 인터넷을 통한 스트리밍에 자주 활용됨.
OGV	.ogg	Xiph 재단에 의해 개발되었으며, MP3의 대안으로 개발된 특허권으로 보호되지 않는 개방형 공개 멀티미디어 파일 형식임.
WebM	.webm	구글의 지원으로 개발된 개방형 공개 멀티미디어 파일 형식으로, 비디오 코덱으로는 VP8, 오디오 코덱으로는 Vorbis를 사용하는 멀티미디어 파일 형식임.
AVI	.avi	Microsoft에 의해 개발되었으며, PC에서 동영상 구현하기 위한 파일 형식임.
WMV	.wmv	Microsoft에 의해 개발되었으며, Microsoft windows media player의 주 스트리밍 파일 형식임.
QuickTime	.mov	Apple에 의해 개발되었으며, 매킨토시 컴퓨터에 동영상을 지원하기 위하여 개발된 파일 형식임.
RealVideo	.rm .ram	Real Networks에 의해 개발되었으며, 스트리밍 기술을 이용한 동영상용 플러그인 파일 형식임.
Flash	.swf .flv	Macromedia에 의해 개발되었으며, 벡터 도형 처리 기반의 애니메이션 제작용 소프트웨어 파일 형식임.

오디오(audio) 파일 형식

- HTML5 표준이 공식적으로 지원하는 오디오 파일 형식은 MP3, WAV, Ogg 뿐

파일 형식	파일 확장자	설명
WAV	.wav	IBM과 Microsoft에 의해 개발되었으며, 개인용 PC에서 오디오를 재생하기 위한 IBM과 Microsoft의 표준 오디오 파일 형식임.
Ogg	.ogg	xiph 재단에 의해 개발되었으며, MP3의 대안으로 개발된 특허권으로 보호되지 않는 개방형 공개 멀티미디어 파일 형식임.
MP3	.mp3	Moving Picture Experts Group에 의해 개발되었으며, MPEG-1의 오디오 규격으로 개발된 손실 압축 파일 형식임.
MP4	.mp4	Moving Picture Experts Group에 의해 개발되었으며, MPEG-4의 일부로 규정된 멀티미디어 컨테이너 파일 형식임. MP4는 비디오 파일 형식이기도 하지만 오디오에서도 사용할 수 있음.
MIDI	.mid .midi	모든 전자 음악기기의 연주 정보를 상호 전달하기 위해 정해진 데이터 전송 규격임.
RealAudio	.rm .ram	Real Networks에 의해 개발되었으며, 인터넷에서 실시간으로 음악을 들을 수 있는 스트리밍 사운드 기술임.
WMA	.wma	Microsoft에 의해 개발되었으며, Microsoft windows media 기술에서 음악 정보(data)만을 압축하는 기술임.
AAC	.aac	Apple에 의해 개발되었으며, iPhone, iPod, iTunes의 기본 오디오 파일 형식임. 표준적인 손실 압축 방식을 사용함.

비디오(video) 요소

- HTML5 이전 비디오(video)를 보여주기 위한 표준 안 부재
 - 비디오를 삽입하기 위해서는 플래시(flash)와 같은 외부 플러그인(plug-in)에 의존
- HTML5에서는 <video>태그를 이용 웹 페이지에 비디오를 삽입하는 표준화된 방식을 제공

<video>

<source src="대상" type="video/형식">

<track kind="subtitles" src="en.vtt" srclang="en" label="English">

</video>

- 웹 브라우저는 여러 개의 <source>태그 중 위쪽에서부터 순서대로 가장 먼저 인식되는 파일의 타입과 주소를 사용
- <track>태그는 비디오가 재생될 때 보일 자막이나 캡션 파일

MP4	video/mp4
WebM	video/webm
OGV	video/ogg

비디오(video) 요소 속성

- control 속성
 - 재생, 정지 및 소리의 조절 등 비디오의 기본적인 동작을 조절할 수 있는 패널을 생성
- height와 width
 - 웹 브라우저에 삽입되는 비디오의 크기를 명시
- autoplay 속성
 - 웹 페이지가 로드(load) 될 때 비디오를 자동으로 재생시켜 줄지 않을지를 설정
- loop 속성
 - 비디오의 재생이 끝나도 계속적으로 반복해서 비디오를 재생
- <video>태그 사이에 존재하는 텍스트는 해당 웹 브라우저가 <video>태그를 지원하지 않을 때만 화면에 표시

오디오(audio) 요소

- HTML5 이전 오디오(audio)를 들려주기 위한 표준안 부재
 - HTML5에서는 <audio>태그를 이용 오디오를 삽입하는 표준화된 방식을 제공

```
<audio controls>
```

```
<source src="/examples/media/sample_audio_ogg.ogg" type="audio/ogg">
```

```
<source src="/examples/media/sample_audio_mp3.mp3" type="audio/mp3">
```

```
</audio>
```

- <source>태그 중 위쪽에서부터 순서대로 가장 먼저 인식되는 파일의 타입과 주소를 사용

파일 형식	미디어 타입
MP3	audio/mp3
WAV	audio/wav
Ogg	audio/ogg

<audio> 요소 속성

- control 속성
 - 재생, 정지 및 소리의 조절 등 오디오의 기본적인 동작을 조절할 수 있는 패널을 생성
- autoplay 속성
 - 웹 페이지가 로드(load) 될 때 음악을 자동으로 재생시켜 줄지 않을지를 설정
- loop 속성
 - 재생이 끝나도 계속적으로 반복해서 오디오를 재생
- preload 속성
 - 오디오 재생 전에 파일의 내용을 모두 불러올지를 설정

첨부파일

일 종류	사용법
이미지	<code></code>
오디오	<code><audio src="경로"> </audio></code>
비디오	<code><video src="경로"> </video></code>
멀티미디어	<code><object data="파일"> </object></code> <code><embed src="경로"></code>
하이퍼링크	<code>txt or img</code>

- 파일 태그 중 **object**는 문서 파일뿐만 아니라 이미지 비디오 등 다양한 파일을 삽입가능
- 추가로 경로를 지정할 때 파일이 같은 경로 내에 있는 것이 아니라면 절대 경로를 사용해야 한다.

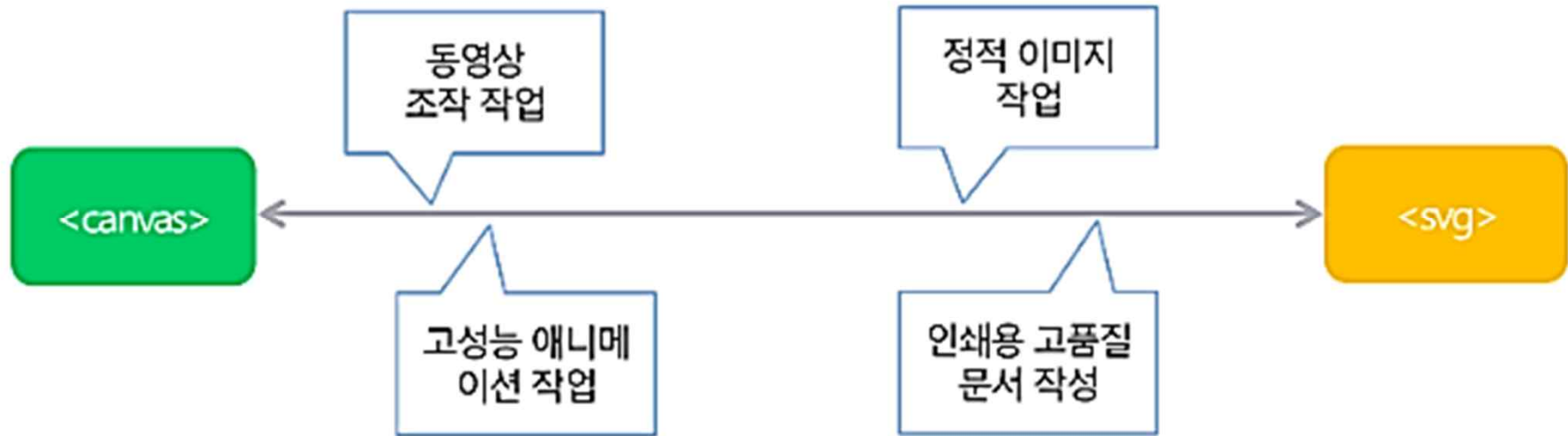
다른 멀티미디어 삽입

- object 요소
- embed 요소
 - HTML5 이전까지는 HTML 표준이 아니었지만 embed 요소는 전부터 사용
- HTML 문서에 삽입할 객체(object) 사용
- 객체뿐만 아니라 HTML 문서를 삽입 가능

그래픽 요소

- Canvas 요소
 - 웹 페이지에 그래픽을 그려주는 쉽고 강력한 방법을 제공
 - 그래픽을 위한 컨테이너(container) 역할만을 수행
 - 실제로 그래픽을 그리기 위해서는 자바스크립트 (JavaScript) 등의 스크립트 언어를 이용
- svg 요소
 - Scalable Vector Graphics(XML 기반의 W3C 그래픽 표준 권고안)
 - canvas 요소로는 벡터(vector) 이미지를 표현 불가
 - 픽셀 기반인 웹 페이지에서 픽셀의 영향을 받지 않는 벡터 이미지를 표현 가능
 - 도표나 그래프 등 벡터 기반의 다이어그램(diagram)을 표현하는 데 매우 효과적

canvas / svg



Canvas	SVG
픽셀(pixel) 기반	모양(shape) 기반
단일 HTML 요소	DOM의 일부분이 되는 다중 그래픽 요소
스크립트(script)를 통해서만 수정할 수 있음.	스크립트(script) 및 CSS를 통해서도 수정할 수 있음.
그래픽이 주작업인 게임에 적합함.	렌더링 영역이 넓은 응용 프로그램(application)에 적합함.

HTML5 API

- HTML5에서 새롭게 추가된 API는 모두 자바스크립트로 구현
- JavaScript >>>

geolocation API

- 사용자의 현재 위치 정보를 가져올 때 사용하는 자바스크립트 API
- 사용자의 위도 및 경도에 관한 정보는 자바스크립트를 이용해 웹 서버로 전송
- 위치기반 서비스 사용
 - 사용자의 사생활을 침해할 가능성이 높으므로, 사용자의 동의 없이는 사용 불가

드래그 앤 드롭 API

- 웹 페이지 내의 요소를 사용자가 자유롭게 드래그할 수 있도록 설정

이벤트	설명
dragstart	사용자가 객체(object)를 드래그하려고 시작할 때 발생함.
dragenter	마우스가 대상 객체의 위로 처음 진입할 때 발생함.
dragover	드래그하면서 마우스가 대상 객체의 위에 자리 잡고 있을 때 발생함.
drag	대상 객체를 드래그하면서 마우스를 움직일 때 발생함.
drop	드래그가 끝나서 드래그하던 객체를 놓는 장소에 위치한 객체에서 발생함.
dragleave	드래그가 끝나서 마우스가 대상 객체의 위에서 벗어날 때 발생함.
dragend	대상 객체를 드래그하다가 마우스 버튼을 놓는 순간 발생함.

DataTransfer 객체

드래그 앤 드롭 이벤트를 위한 모든 이벤트 리스너 메소드(event listener method)는 드래그 앤 드롭 동작에 관한 정보를 가지고 있는 DataTransfer 객체를 반환

draggable 속성

드래그될 수 있는 객체(draggable object)로 변환가능한 속성

웹 스토리지(web storage) API

- 웹 브라우저가 직접 데이터(data)를 저장
 - HTML5 이전에는 응용 프로그램이 데이터를 서버에게 요청할 때마다 매번 쿠키(cookie)라는 곳에 그 정보를 저장
- 사용자 측에서 좀 더 많은 양의 정보를 안전하게 저장
 - (최소 5MB 이상, 서버로 전송되지 않음)
- 오리진(origin)마다 하나씩만 존재
 - 오리진(origin)이란 도메인(domain)과 프로토콜(protocol)의 한 쌍으로 이루어진 식별자
- 하나의 오리진에 속하는 모든 웹 페이지는 같은 데이터(data)를 저장/접근 가능
- sessionStorage 객체
 - 하나의 세션(session)만을 위한 데이터를 저장하는 객체
- localStorage 객체
 - 보관 기한이 없는 데이터를 저장할 수 있는 객체

web worker API

- 스크립트가 웹 페이지의 성능에 영향을 미치지 않도록 백그라운드에서 동작하게 해주는 자바스크립트입
 - 스크립트가 실행되면, 해당 웹 페이지는 실행 중인 스크립트가 종료될 때까지 응답 불가 상태
 - 스크립트의 다중 스레드(multi-thread)를 지원

마무리