

03

HTML

PNU Mini Bootcamp – Front End

HTML이란

Hyper Text Markup Language

- Markup Language 는 문서를 작성하기 위한 언어이며 이 문서를 해석하는 애플리케이션에게 어떻게 해석하라는 의사소통을 목적으로 한다.
- Hyper link 로 연결된 문서
- HTML은 웹에서 자유롭게 오갈 수 있는 웹 문서를 만드는 언어
- HTML 문서는 이 문서를 만드는 애플리케이션과 이 문서를 해석하는 애플리케이션이 지정되지 않는다.
- 단지 문서를 만드는 규칙만 정의되어 있을 뿐이다.
- 그럼으로 애플리케이션에 비 종속적
- 전 세계의 각종 문서를 연결하는데 용이하다.

HTML이란

Hyper Text **Markup Language**

- 마크업은 일종의 태그이다.
- 인스타그램의 태그와 목적은 동일하다.
- 인스타그램의 태그는 등록하는 이미지에 대한 설명, 혹은 검색을 위한 검색어다.
- 태그는 문서의 글을 감싸서 그 글이 어떤 역할을 하는지를 표현하는 것이다.
-
- 인스타그램등의 태그는 # 규칙으로 등록된다.
- 마크업 태그는 <> 규칙으로 등록된다.

HTML이란

vs XML

- Extensible Markup Language
- Html 이 먼저 나왔다.
- Hyper text markup language 이외에 다양한 문서를 태그로 표현하기 위해 xml 이 정의되었다.
- Xml 이 나오면서 html 이 xml 의 subset 이 되었다.

XML

- HTML 과 약간의 차이가 있다.
- 태그는 종료태그를 가져야 한다.
- 대소문자를 구분한다.
- 속성 값은 꼭 " , ' 로 묶여야 한다.

HTML 문서의 구조



<!DOCTYPE html>

- 웹 브라우저에게 '이제부터 처리할 문서는 HTML 문서'라고 알려주는 것

<html>

- 실제 문서 정보와 내용이 시작되고 끝나는 것을 표시
- lang 속성을 사용해 문서에서 사용할 언어 지정
- HTML 문서를 브라우저에 출력할 때는 의미가 없다.
- 단지 어떤 소프트웨어가 이 문서를 유저들에게 읽어줄 때 언어가 지정되면 이 언어로 읽어 줄 수도 있다.
- 또한 검색엔진의 검색조건 중 언어에 대한 조건에 반응하기 위해서 명시된다.

HTML 문서의 구조

<head>

- 문서 관련 정보 입력, 웹 브라우저 화면에는 보이지 않음.
- 문서에서 사용할 외부 파일 링크

<meta>

- 문서의 다양한 정보 설정
- 대표적 설정이 문서를 해석할 때 적용할 언어 셋
- `html parser` 를 위한 정보 혹은 특정 애플리케이션 만을 위한 meta 정보도 설정
- `<meta name = " " content = " " />`로 설정하는 것이 기본이다.
- `charset` 처럼 웹 문서의 기본 정보는 `charset` 이라는 속성이 따로 정의되어 있는 것 뿐이다.
- 검색엔진에 노출에 중요한 부분을 차지한다.

HTML 문서의 구조

<title>

- 브라우저 탭 제목에 사용되며
- 검색엔진에 문서가 검색될 때 가장 중요한 역할을 한다.
- 대부분의 경우 등록해 준다.
- HEAD 태그에는 이 이외에도 TITLE, SCRIPT, LINK 등의 태그가 들어갈 수 있다.

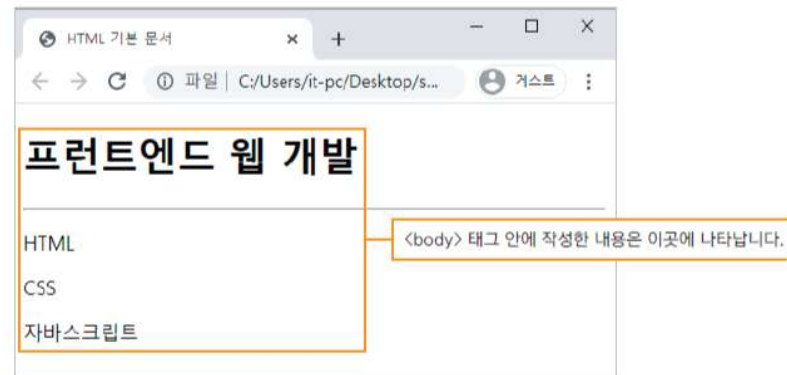


HTML 문서의 구조

<body>

- 문서 내용
- 화면 출력 정보

```
<body>  
  <h1> 프론트엔드 웹 개발 </h1>  
  <hr>  
  <p>HTML</p>  
  <p>CSS</p>  
  <p>자바스크립트</p>  
</body>
```



기초 태그

<h1>....<h6>

- 제목표시
- <h1> 제목 </h1>
- 크기 : h1 > h2 > h3 > h4 > h5 > h6

<p>

- 단락표시
- <p> 텍스트 </p>
- 입력한 내용 앞뒤로 빈 줄이 생기면서 텍스트 단락이 만들어짐

**
**

- 줄 바꿈
-

기초 태그

<blockquote>

- 인용문
- `<blockquote>` 인용 내용 `</blockquote>`
- 다른 텍스트보다 안으로 들여 써짐

**, **

- 굵게 표시
- `` 내용 ``, `` 내용 ``
- `` - 중요한 내용이라서 강조해야 할 때
- `` - 단순히 굵게 표시할 때

<i>

- 이탤릭체로 표시
- `` 내용 ``, `<i>` 내용 `</i>`
- `` - 흐름상 특정 부분을 강조하고 싶을 때
- `<i>` - 단순히 이탤릭체로 표시할 때

기초 태그

`` ``

- 목록
- 각 항목 앞에 숫자가 붙여짐
- type 속성 : 순서 목록의 숫자 조정
(1: 숫자, a: 소문자, A:대문자, i:로마소문자, I:로마대문자)
- start 속성 : 목록의 시작 번호 수정

`` ``

- 순서가 없는 목록
- 각 항목 앞에 불릿이 붙여짐

기초 태그

<dl> <dt> <dd>

- '이름(제목)'과 '값(설명)' 형태로 된 설명 목록
- dl – definition list(정의목록) – 용어를 설명하는 목록
- dt – definition term (정의용어) – 용어의 제목
- dd – definition description (정의 설명)

```
<h2>상품 구성</h2>
<dl>
  <dt>선물용 3kg</dt>
  <dd>소과 13~16과</dd>
  <dd>중과 10~12과</dd>
</dl>
<dl>
  <dt>선물용 5kg</dt>
  <dd>중과 15~19과</dd>
</dl>
```

상품 구성

선물용 3kg
소과 13~16과
중과 10~12과

선물용 5kg
중과 15~19과

기초 태그

<table> <caption> <thead> <tbody> <tfoot> <tr> <th> <td>

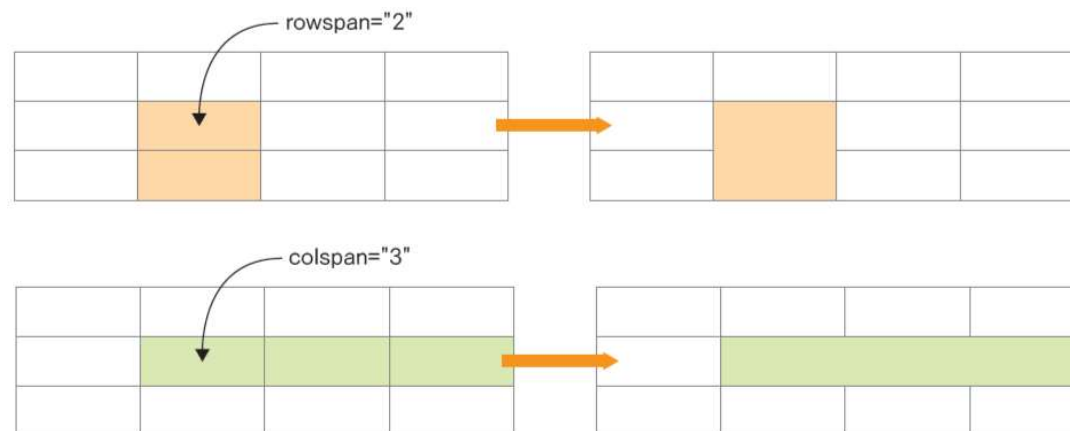
- <table> : 표 전체
- <caption> : 표 제목
- <tr> : 행
- <td> : 셀
- <th> : 제목 셀
- <thead> <tbody> <tfoot> 는 화면과는 관련 없다.
- 화면 낭독기 혹은 자바스크립트의 제어적인 문제

```
<table>
  <caption>선물용과 가정용 상품 구성</caption>
  <thead>
    <tr>
      <th>용도</th>
      <th>중량</th>
      <th>갯수</th>
      <th>가격</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>선물용</td>
      <td>3kg</td>
      <td>11~16과</td>
      <td>35,000원</td>
    </tr>
    ... ..
    <tr>
      <td>가정용</td>
      <td>5kg</td>
      <td>18~26과</td>
      <td>47,000원</td>
    </tr>
  </tbody>
</table>
```

기초 태그

colspan, rowspan 속성

- 행, 열 합치기
- `<td rowspan="합칠 셀의 개수">셀의 내용</td>`
- `<td colspan="합칠 셀의 개수">셀의 내용</td>`



기초 태그

- 웹 문서에 이미지를 삽입할 때 사용
- ``

이미지 포맷

- 색상수
- 투명을 표현할 수 있는지
- 애니메이션을 지원하는지
- 압축이 가능한지
- 압축 손실이 있는지
- 레스터(비트맵) VS 벡터 이미지인지.

기초 태그

Gif(Graphic Interchange Format)

- 그래픽을 압축해 빠르게 전송하기 위해 사용
- 초기 인터넷에서 사용
- 작은 용량으로 8비트 256 색상 표현
- 인터넷에서 움직이는 animated gif 파일

JPEG(Joint Photographic Experts Group)

- 넓은 범위의 색 지원, 카메라, 스캐너에서 흔히 사용
- 압축 효율이 좋음
- 1600만 색상 표현이 가능해 고해상도에 적합
-
- 압축 손실이 있다.
- 확대 시 선명하지 않다.
- 투명성을 표현하지 못한다.
- 애니메이션을 지원하지 않는다.

PNG(Portable Network Graphics)

- W3C 에서 제정
- JPEG 보다 더 고품질 이미지 생성가능
- 파일 크기가 JPEG 보다 크다.
- 투명표현한다.
- 애니메이션 지원하지 않는다.
- 1600만 색상 표현한다.
- 압축 손실이 없다.
- 로그, 데이터가 출력되는 이미지에 활용

WebP

- 구글이 개발
- 파일 압축해 로딩할 때 속도를 높이기 위해 개발
- 압축률이 좋아 웹에서 이상적인 포맷으로 알려져 있다.
- 지원하지 않는 브라우저, 애플리케이션들이 많았다.
- 과거의 포토샵, 과거의 los
- 점점 이용비율이 늘어나고 있다.

기초 태그

<a>

- 다른 문서, 혹은 다른 사이트로 바로 연결해 주는 기능
- `주문서 작성하기`

기초 태그

<object> <embed>

- 음악, 동영상이나 자바 애플릿, PDF 파일 등 다양한 개체를 삽입
 - 웹의 기본인 문자열과 이미지를 제외한 다양한 콘텐츠 혹은 plugin 을 연동시키기 위해 사용
 - object, embed 는 별개로 사용가능
 - 보통의 경우 object 태그를 이용하면 되는데 과거 IE 브라우저에서 object 의 영상등이 출력되지 않는 문제가 있었다.
 - 그래서 object 내에 embed 를 삽입해서 브라우저가 object 를 지원하지 않는다면 embed 에 의해 처리하게 설정 했던 것이다.
 - 결국 embed 는 object 의 polyfill 개념이다.
-
- `<object width="900" height="800" data="product.pdf"></object>`
 - `<embed src="파일 경로" width="너비" height="높이">`

기초 태그

<audio> <video>

- `<audio src="medias/spring.mp3" controls></audio>`
- `<video src="medias/salad.mp4" controls width="700"></video>`

속성	설명
<code>controls</code>	플레이어 화면에 컨트롤 바를 표시합니다.
<code>autoplay</code>	오디오나 비디오를 자동으로 실행합니다.
<code>loop</code>	오디오나 비디오를 반복 재생합니다.
<code>muted</code>	오디오나 비디오의 소리를 제거합니다.
<code>preload</code>	페이지를 불러올 때 오디오나 비디오 파일을 어떻게 로딩할 것인지 지정합니다. 사용할 수 있는 값은 <code>auto</code> , <code>metadata</code> , <code>none</code> 입니다. 기본적으로 <code>preload="auto"</code> 가 사용됩니다.
<code>width, height</code>	비디오 플레이어의 너비와 높이를 지정합니다. <code>width</code> 나 <code>height</code> 의 값 중에서 하나만 지정하면 나머지는 자동으로 계산해서 표시합니다.
<code>poster="파일 이름"</code>	<code><video></code> 태그에서 사용하는 속성으로, 비디오가 재생되기 전까지 화면에 표시될 포스터 이미지를 지정합니다.

Form 관련 태그

HTTP: Stateless

TCP/IP: Stateful

<form>

- 폼을 만드는 기본 태그. <form>과 </form> 사이에 여러 폼 요소 삽입

속성	설명
method	사용자가 입력한 내용을 서버 쪽 프로그램으로 어떻게 넘겨줄 것인지 지정합니다. method에서 사용할 수 있는 속성값은 get과 post입니다. <ul style="list-style-type: none">• get: 데이터를 256~4,096byte까지만 서버로 넘길 수 있습니다. 주소 표시줄에 사용자가 입력한 내용이 그대로 드러나는 단점이 있습니다.• post: 입력한 내용의 길이에 제한받지 않고 사용자가 입력한 내용도 드러나지 않습니다.
name	자바스크립트로 폼을 제어할 때 사용할 폼의 이름을 지정합니다.
action	<form> 태그 안의 내용을 처리해 줄 서버 프로그램을 지정합니다.
target	action 속성에서 지정한 스크립트 파일을 현재 창이 아닌 다른 위치에서 열도록 합니다.

웹에 request를 보내면 무조건 response가 오는데 그걸 어떻게 어디다가 뿌릴건지 = target

Form 관련 태그

<label> <legend>

- <input> 태그의 입력 요소에 레이블을 추가
- <label>Name <input type="text"/> </label>
- <label for="name">Name</label>
 <input type="text" id="name"/>
- for 은 화면과는 상관없다.
- for 는 화면 낭독기를 위해 사용된다.
- <label> 과 <input> 을 따로 선언하고 for 와 id 를 매핑하는 경우가 일반적이다.

Form 관련 태그

<input>

- type 으로 입력 요소 결정

datetime	국제 표준시(UTC)로 설정된 날짜와 시간(연, 월, 일, 시, 분, 초, 분할 초)을 넣습니다.
datetime-local	사용자가 있는 지역을 기준으로 날짜와 시간(연, 월, 일, 시, 분, 초, 분할 초)을 넣습니다.
submit	전송 버튼을 넣습니다.
reset	리셋 버튼을 넣습니다.
image	submit 버튼 대신 사용할 이미지를 넣습니다.
button	일반 버튼을 넣습니다
file	파일을 첨부할 수 있는 버튼을 넣습니다.
hidden	사용자에게는 보이지 않지만 서버로 넘겨주는 값이 있는 필드를 만듭니다.

종류	설명
text	한 줄짜리 텍스트를 입력할 수 있는 텍스트 박스를 넣습니다.
password	비밀번호를 입력할 수 있는 필드를 넣습니다.
search	검색할 때 입력하는 필드를 넣습니다.
url	URL 주소를 입력할 수 있는 필드를 넣습니다.
email	이메일 주소를 입력할 수 있는 필드를 넣습니다.
tel	전화번호를 입력할 수 있는 필드를 넣습니다.
checkbox	주어진 여러 항목에서 2개 이상 선택할 수 있는 체크 박스를 넣습니다.
radio	주어진 여러 항목에서 1개만 선택할 수 있는 라디오 버튼을 넣습니다.
number	숫자를 조절할 수 있는 스핀 박스를 넣습니다.
range	숫자를 조절할 수 있는 슬라이드 막대를 넣습니다
date	사용자 지역을 기준으로 날짜(연, 월, 일)를 넣습니다.
month	사용자 지역을 기준으로 날짜(연, 월)를 넣습니다.
week	사용자 지역을 기준으로 날짜(연, 주)를 넣습니다.
time	사용자 지역을 기준으로 시간(시, 분, 초, 분할 초)을 넣습니다.

Form 관련 태그

type – search, url, email, tel

- 유효성 검증에서 의미가 있다.
- 모바일에서 의미가 있다.

type – checkbox, radio

- value 는 서버에 전송될 데이터
- radio 에서는 name 을 동일하게 해야 그룹핑이 된다.

type – number, range

- 숫자 입력 필드. 브라우저에 따라 스펀 박스로 표시됨
- 숫자 입력 필드. 슬라이드 막대를 이용해 숫자 입력

Form 관련 태그

type – date, month, week

- 모든 브라우저에서 지원하지 않는다.
- caniuse.com 사이트에서 브라우저의 지원사항을 확인



Form 관련 태그

type – submit, reset, image

- 전송(submit) 버튼 : 사용자 입력 내용을 서버로 전송
- 리셋(reset) 버튼 : 사용자 입력 내용 전부 삭제
- value 속성을 이용해 버튼 표시 내용 지정
- `<input type="image" src="이미지경로"/>` 로 submit 대체



아이디: 비밀번호:

type – button

- 기능 없이 버튼 형태만 삽입
- 주로 버튼 클릭해서 자바스크립트 실행할 때 사용

Form 관련 태그

type – file

- '파일 선택' 이나 '찾아보기' 버튼으로 표시됨 파일업로드 post방식으로만 가능

type – hidden

- 화면 상의 폼에는 보이지 않는다.
- 폼을 서버로 전송할 때 서버로 함께 전송되는 요소

Form 관련 태그

autofocus

- 페이지를 불러오자마자 원하는 폼 요소에 마우스 커서 표시

placeholder

- 입력란에 표시하는 힌트로, 필드를 클릭하면 사라짐

readonly, disabled

- 내용을 보기만 하고 입력하지 못하게 함.
- Readonly 는 포커스는 가질 수 있다.
- Disabled 는 서버 전송도 되지 않는다.

required

- 필수 필드 체크
- 필수 필드는 브라우저에서 직접 체크하는 것이므로 오류 메시지 내용은 브라우저들마다 다르게 나타남

Form 관련 태그

<textarea>

- 여러 줄의 텍스트 입력

종류	설명
cols	텍스트 영역의 가로 너비를 문자 단위로 지정합니다.
rows	텍스트 영역의 세로 길이를 줄 단위로 지정합니다. 지정한 숫자보다 줄 개수가 많아지면 스크롤 막대가 생깁니다.

Form 관련 태그

<select> <option>

- 여러 옵션 중에서 선택 – 드롭다운 목록
- <select>

종류	설명
size	화면에 표시할 드롭다운 항목의 개수를 지정합니다.
multiple	드롭다운 목록에서 둘 이상의 항목을 선택할 때 사용합니다.

- <option>

종류	설명
value	해당 항목을 선택할 때 서버로 넘겨줄 값을 지정합니다.
selected	드롭다운 메뉴를 삽입할 때 기본적으로 선택해서 보여 줄 항목을 지정합니다.

Form 관련 태그

<datalist> <option>

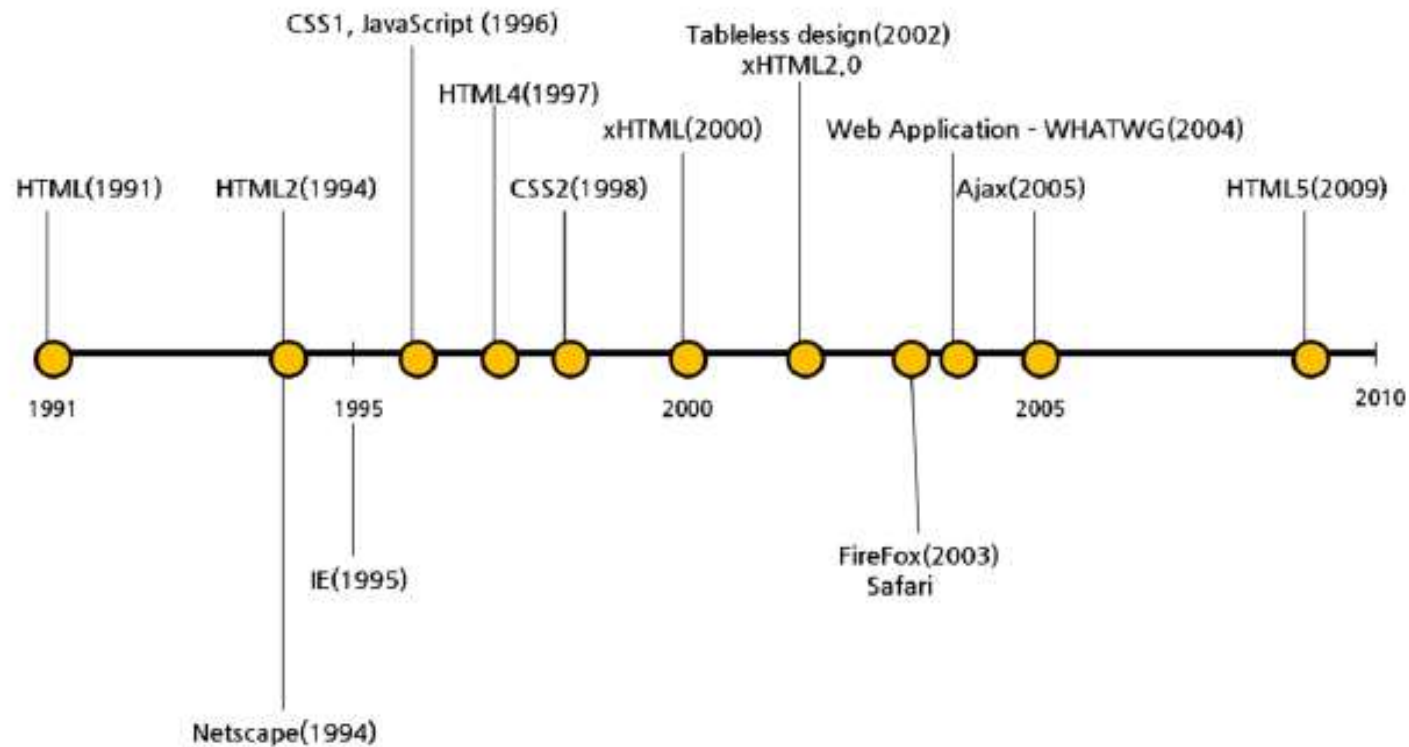
- 데이터 목록에 제시한 값 중에서 선택하면 그 값이 자동으로 텍스트 필드에 입력됨
- 데이터 목록에 id를 이용해 이름을 붙이고,
- <input> 태그의 list 속성에 데이터 목록 id를 지정함.

```
기본형 <input type="text" list="데이터 목록 id">
      <datalist id="데이터 목록 id">
        <option value="서버로 넘길 값1">선택 옵션1</option>
        <option value="서버로 넘길 값2">선택 옵션2</option>
      </datalist>
```

web의 query문의 기본형태: key=value&key=value

Semantic Tag

► HTML Timeline



Semantic Tag

HTML의 한계

•HTML4

- 1990년대 초부터 소개되어 진화를 거듭하던 표준안은 1997년 HTML4를 마지막으로 더 이상의 버전업이 없었다.
- 1990년대의 웹 환경에 초점을 맞추어 지다보니 브라우저 환경에서의 UI 표현에만 치중, 기능적 구현 한계

•Vendor 중심의 기술발전

- Active X Control
- Script Language
- Flash, Flex, SilverLight

• Web2.0 과 Web Application

- Web의 UI 표현 중심에 한계, Application 급의 다양한 기술에 대한 요구
- Ajax의 등장
- Vendor 중심의 기술 등장 – Flex, SilverLight
- 다양한 브라우저의 등장 – Chrome, Safari, Opera, FireFox

Semantic Tag

- **새로운 웹 표준에 대한 필요성**

- WHATWG 주도로 새로운 Web 표준 책정 작업 시작
- WHATWG 의 작업을 W3C 에서 이어받아 HTML5 로 명명
- Mobile 환경의 급격한 변화로 인한 HTML5 필요성 급부상

- **HTML5 의 목적**

- 특정 Vendor 독점 기술에 의존하지 않는 공개되고 표준화된 웹 환경을 만들기 위한 기술
- add-on 또는 plug-in 없이도 웹을 사용 가능
- Application 수준의 다양한 기능구현 기능

Semantic Tag

•Web Application 을 위한 Javascript API

- 비디오/오디오 재생을 위한 요소(video/audio)
- 2D, 3D 그래픽 처리(canvas)
- 오프라인에서도 작동되는 애플리케이션(애플리케이션 캐시)
- 클라이언트 측에 데이터 저장(Web Storage, Web SQL Database)
- 백그라운드 처리 수행(Web Workers)
- 서버로부터의 데이터 푸시나 서버와의 양방향 통신 수행(Server-Sent Events, Web Sockets)
- 로컬 파일의 내용을 읽어들이(File API)

• Semantic Markup

- 문서 구조의 의미나 문서 안에 삽입된 데이터의 의미 등을 명확히 하기 위한 사양이 다수 포함되어 Semantic 웹의 구현
- <header>, <section>, <footer>, <article>, <nav> 등

• CSS3

- 다양한 CSS 기법
- CSS3 Media Query
- CSS3 Animation

Semantic Tag

Semantic Tag 필요성

- 화면 낭독기 같은 보조 기기에서 사이트의 구조를 이해
- 문서 구조가 정확히 나뉘지므로 PC나 모바일 등 다양한 화면에서 웹 문서를 표현하기가 쉽다
- 웹 사이트를 검색할 때 필요한 내용을 정확히 찾을 수 있다



Semantic Tag

<header>

- 헤더 영역
- 사이트 전체의 헤더 or 특정 영역의 헤더
- 검색 창이나 사이트 메뉴 삽입

<nav>

- 내비게이션 영역

```
<header>
  <div id="logo">
    <a href="#"><h1>Dream Jeju</h1></a>
  </div>
  <nav>
    <ul id="topMenu">
      <li><a href="#">단체 여행</a></li>
      <li><a href="#">맞춤 여행</a></li>
      <li><a href="#">갤러리</a></li>
      <li><a href="#">문의하기</a></li>
    </ul>
  </nav>
</header>
```

로고 영역

내비게이션 영역

Semantic Tag

<main>

- 웹 문서에서 핵심이 되는 내용
- 웹 문서마다 다르게 보여주는 내용으로 구성
- 웹 문서에서 한 번만 사용 권장

<article>

- 독립된 웹 콘텐츠 항목 (따로 떼어도 콘텐츠가 되는 내용)
- <section> 태그를 포함할 수 있음

Semantic Tag

<section>

- 콘텐츠 영역
- 몇 개의 콘텐츠를 묶는 용도로 사용
- `css` 적용을 위해 묶는 용도로 쓰지 말 것을 권장, `<div>` 사용

```
<main class="contents">  
  <section id="headling">  
    <h2>몸과 마음이 치유되는 섬</h2>  
    .....  
  </section>  
  <section id="activity">  
    <h2>다양한 액티비티가 기다리는 섬</h2>  
    .....  
  </section>  
</main>
```

두 개의 <section> 태그 영역으로 나누었습니다.

Semantic Tag

<aside>

- 본문 내용 외에 왼쪽이나 오른쪽, 혹은 아래쪽에 사이드 바 표시
- 필수 요소가 아니므로 필요할 경우에만 사용

<footer>

- 사이트 제작 정보나 저작권 정보, 연락처 등

<div>

- CSS, JS 적용위한 영역 구분

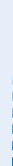
```
<footer>
  <div id="bottomMenu">
    <ul>
      <li><a href="#">회사 소개</a></li>
      <li><a href="#">개인정보처리방침</a></li>
      <li><a href="#">여행약관</a></li>
      <li><a href="#">사이트맵</a></li>
    </ul>
  </div>
</footer>
```

```
<header>
  <div id="logo">
    <a href="#"> <h1>Dream Jeju</h1>
  </a>
</div>
  <nav>
    <ul id="topMenu">
      .....
    </ul>
  </nav>
</header>
```



감사합니다

단단히 마음먹고 떠난 사람은
산꼭대기에 도착할 수 있다.
산은 올라가는 사람에게만 정복된다.



윌리엄 셰익스피어
William Shakespeare