

HTML

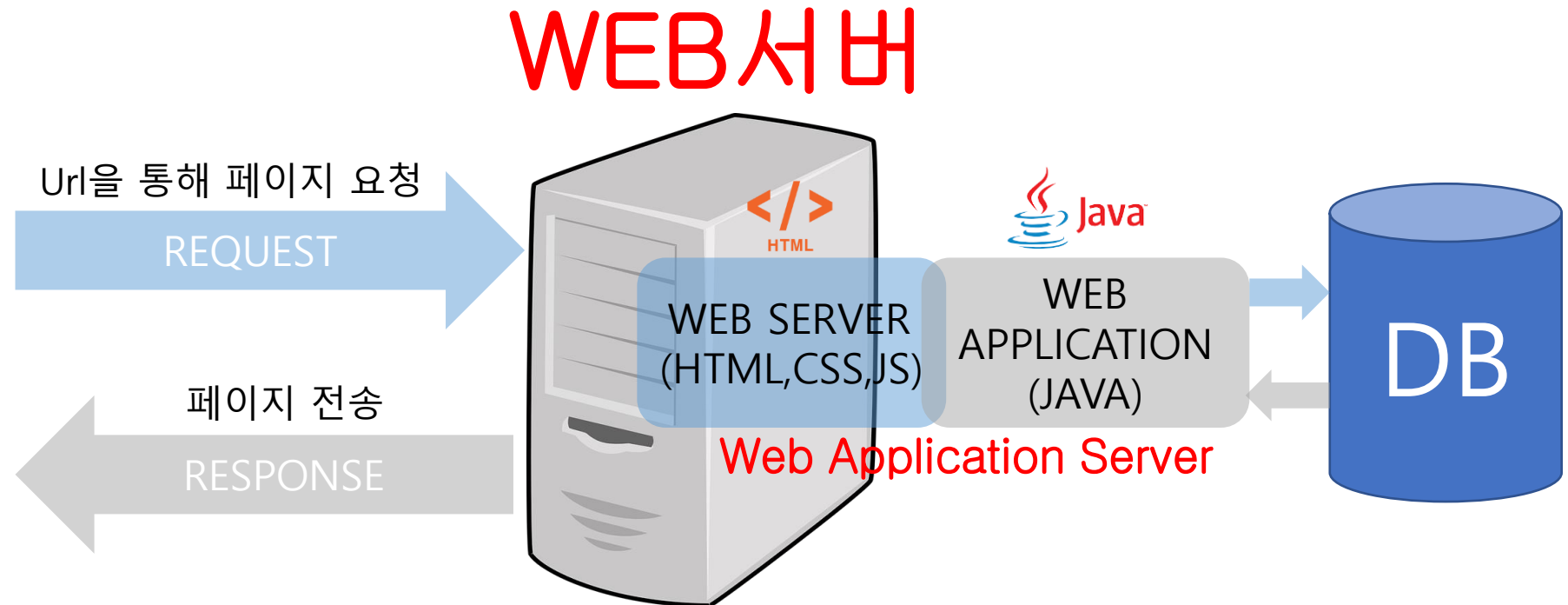
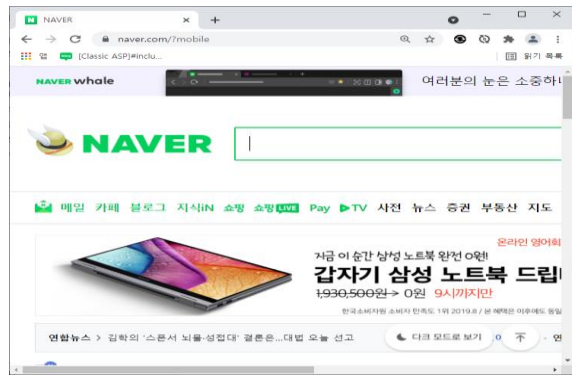
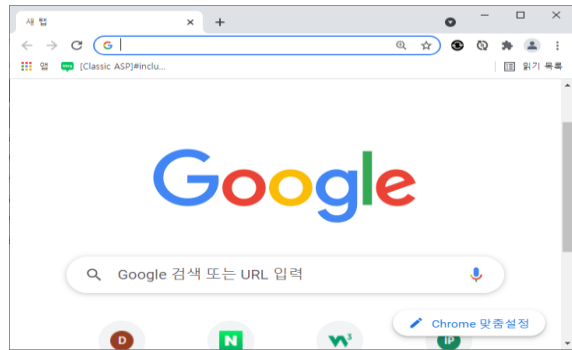
By Jeon.k.w

목차

- I. 웹 프로그래밍과정 이해하기 및 서버설치
- II. HTML
- III. CSS
- IV. JAVASCRIPT

I. 웹프로그래밍 과정 이해 및 서버 설치

웹프로그래밍 과정 이해하기



URL(Uniform Resource Locator)

- 인터넷 상의 자원의 위치
- 특정 웹 서버의 특정 파일에 접근하기 위한 경로 혹은 주소
 - 프로토콜(Protocol) : 네트워크 상에서 약속한 통신규약
ex) https, http, FTP, SMTP, POP
 - IP : 네트워크 상에서 컴퓨터를 식별할 수 있는 주소
 - DNS : IP 주소를 인간이 쉽게 외우도록 맵핑한 문자열
 - Port : 해당 컴퓨터의 구동되고 있는 프로그램을 구분할 수 있는 번호

Port
↑
http://www.naver.com:80/kr/index

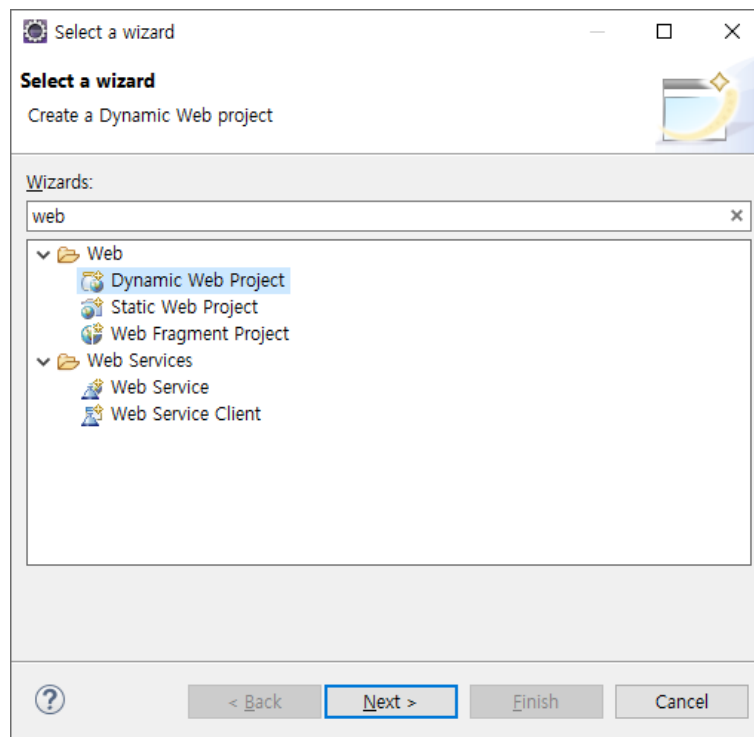
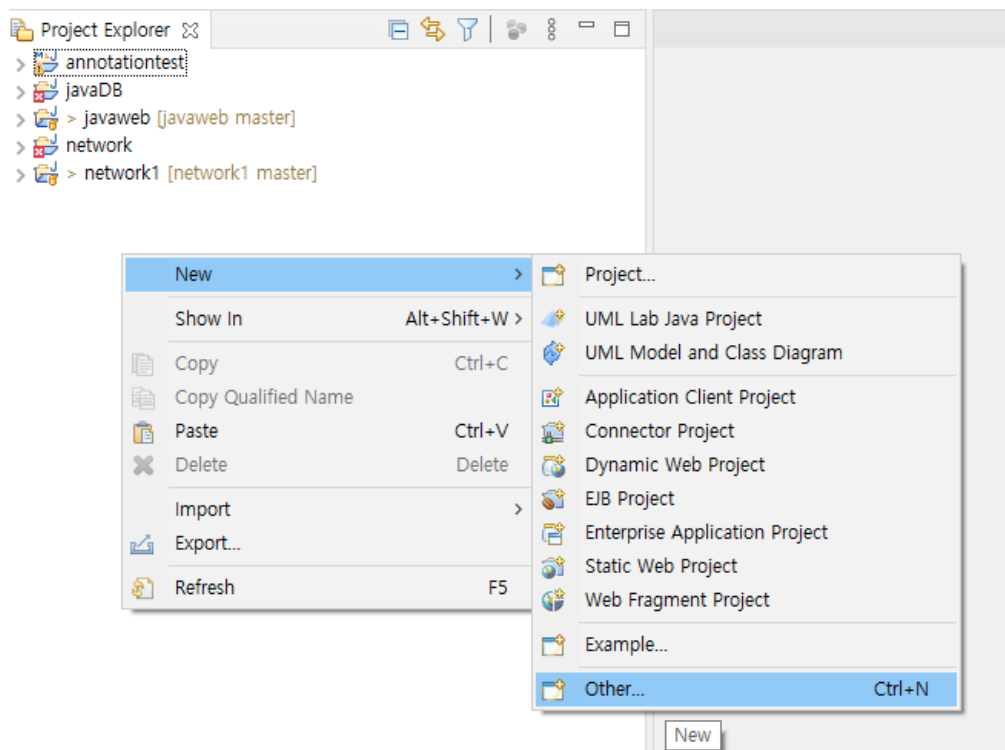
프로토콜 컴퓨터 주소(DNS를 통한 IP주소로 변경) Information path

HTTP



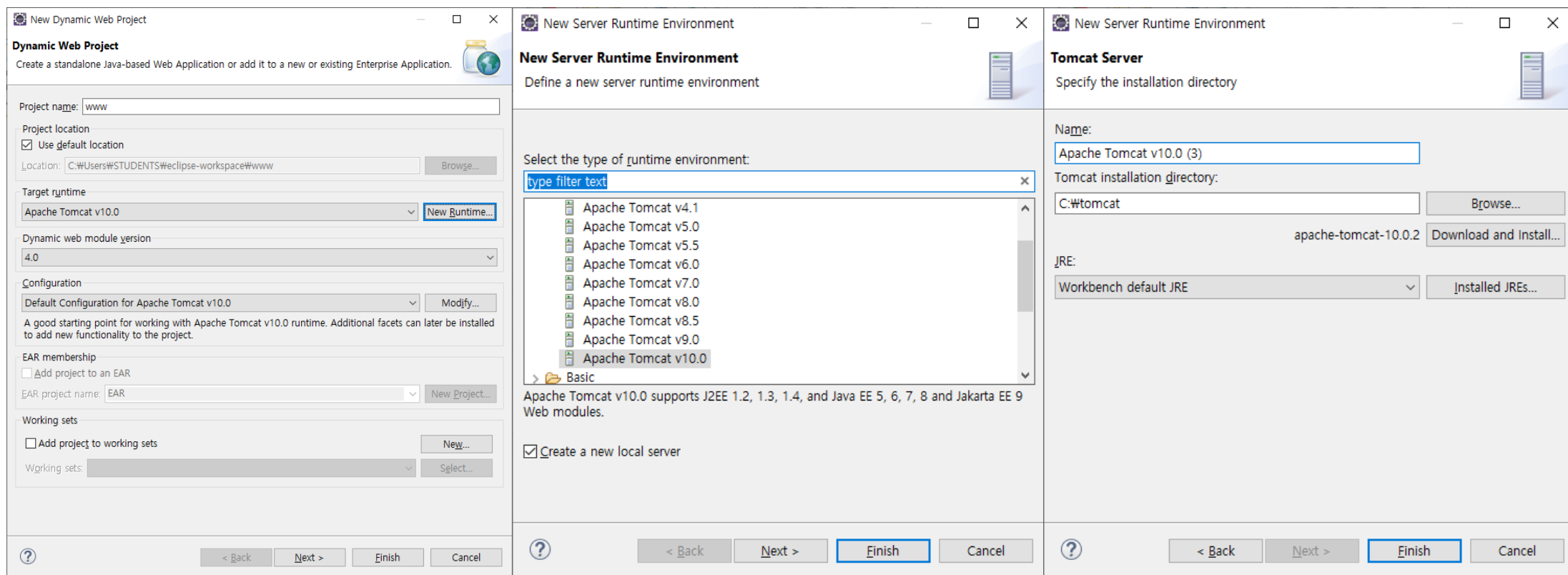
웹프로젝트 생성

- 웹프로젝트를 다음과 같이 선택하여 생성합니다.

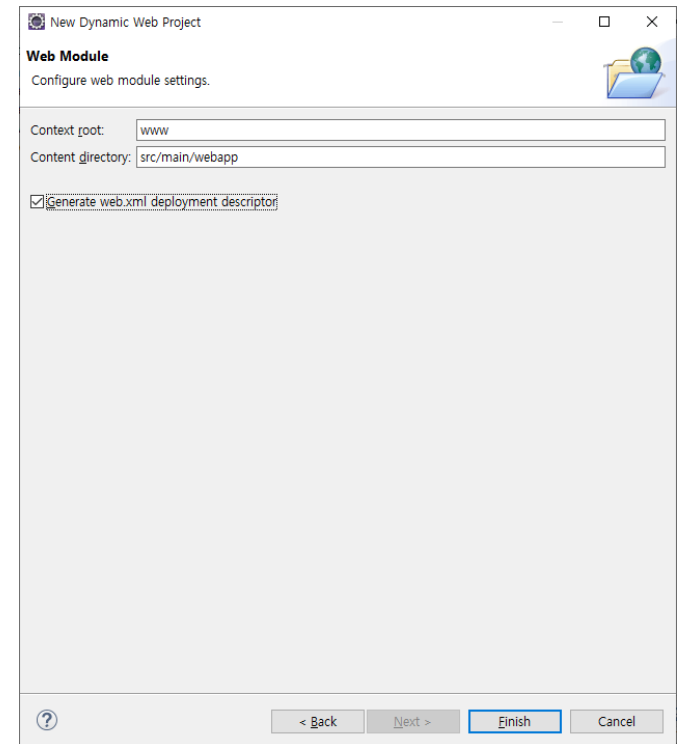
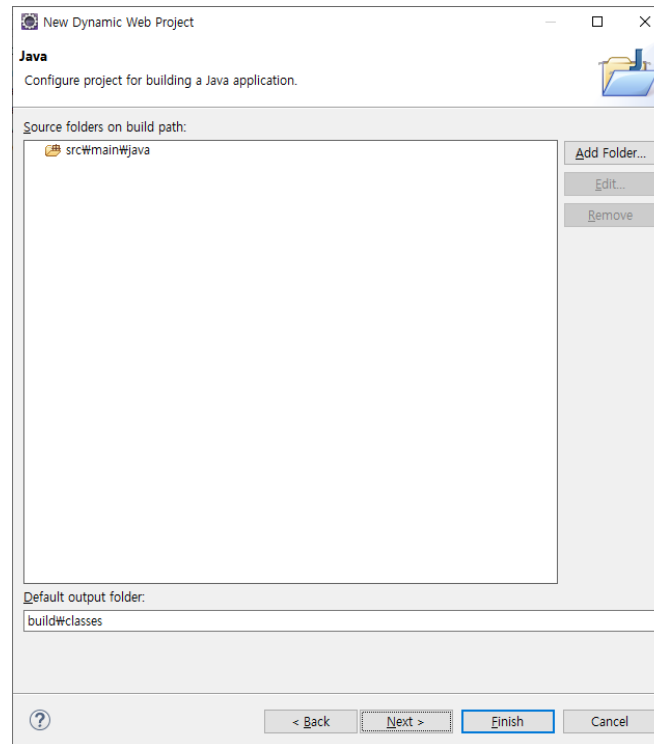
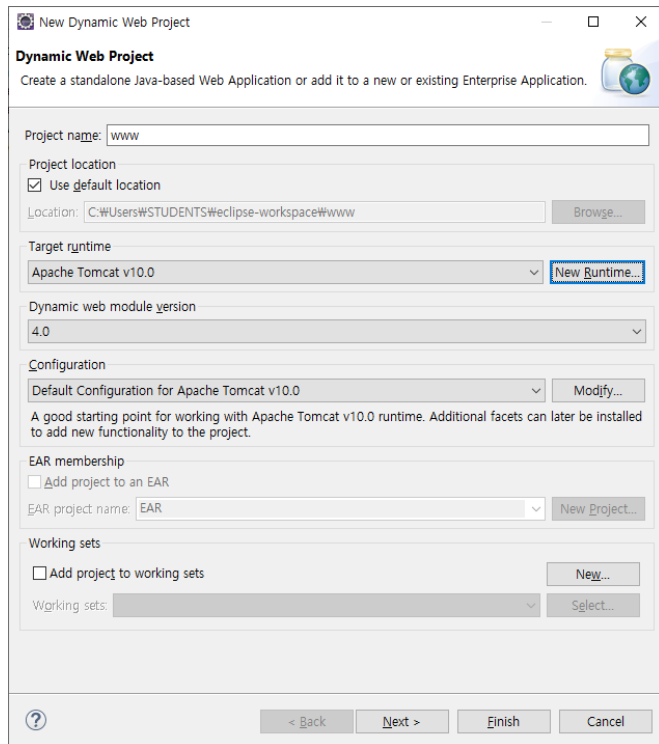


서버설정

- New runtime을 클릭하여 타겟 서버를 설정하는 동시에 서버도 생성함



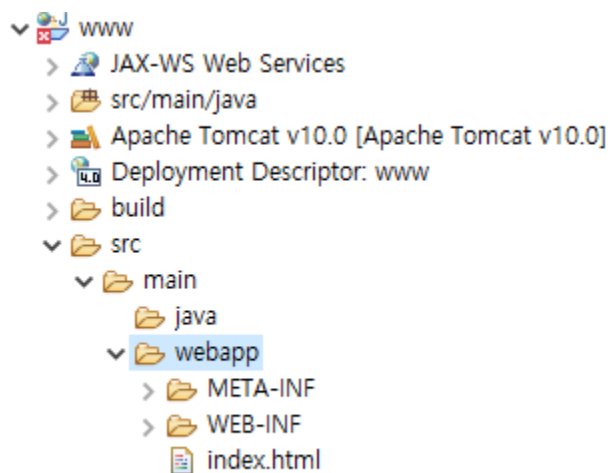
New runtime을 완료하면 서버는 자동으로 생성됨
마법사를 계속 진행하여 프로젝트 설정을 완료함.



파일 생성 및 실행

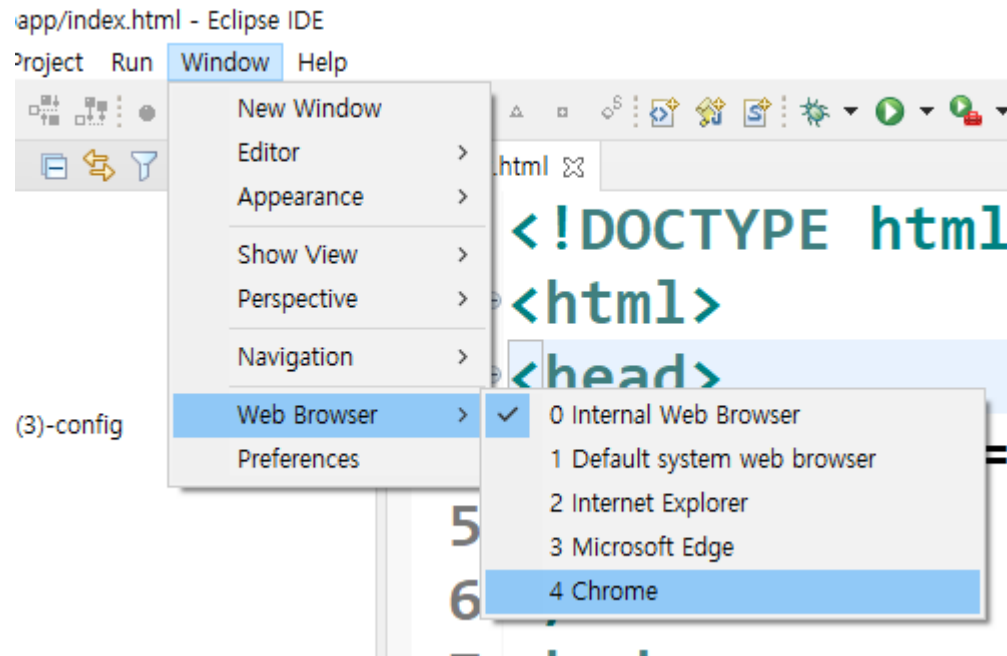
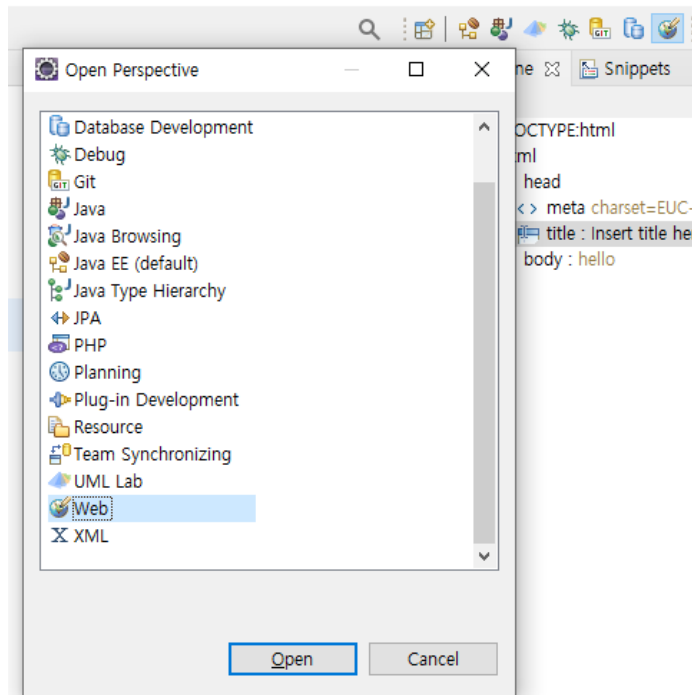
Html 파일의 시작 위치는 webapp폴더임

Index.html 파일을 만들고 실행(웹페이지를 실행할 때 ctrl+f11)



실행 브라우저 바꾸기

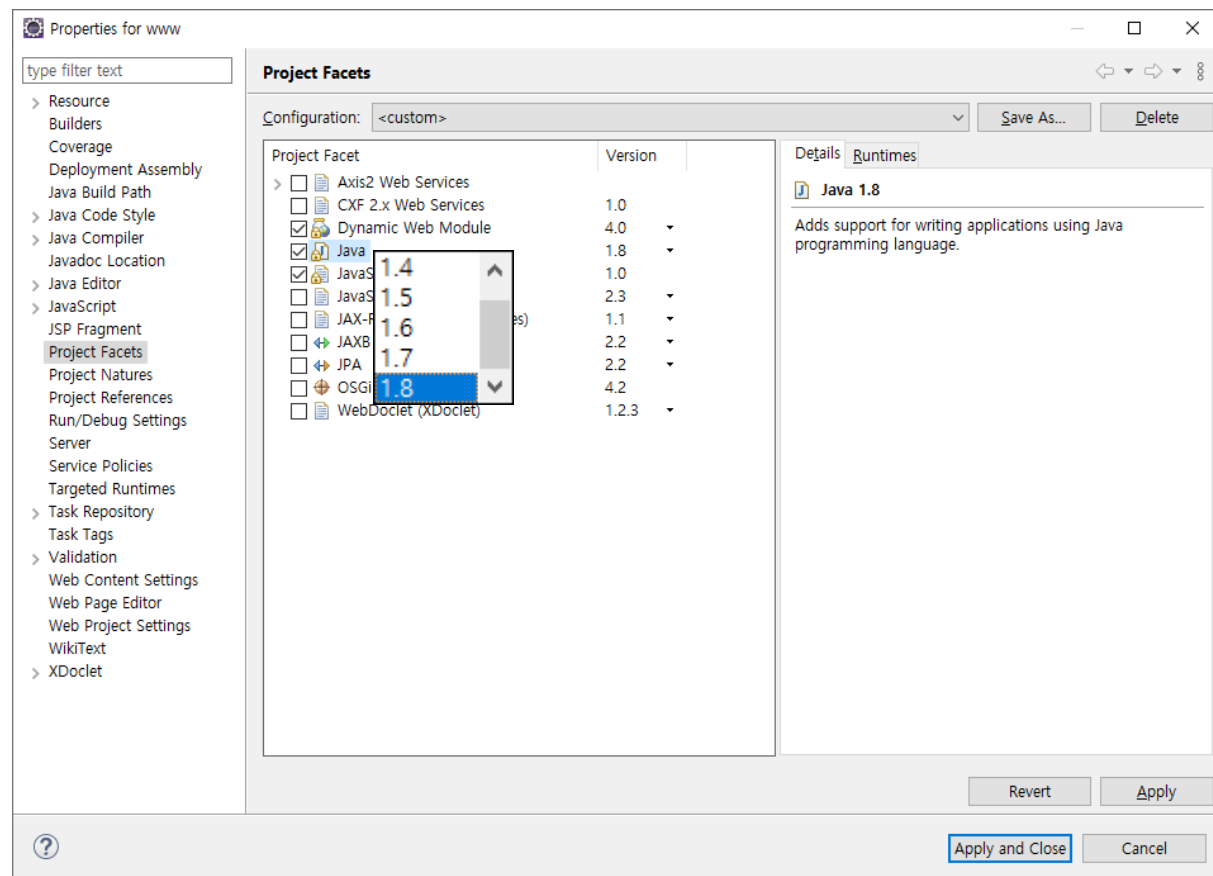
perspective에서 web을 추가하여 프로젝트 작업환경 변경
실행브라우저 선택



프로젝트 오류사항

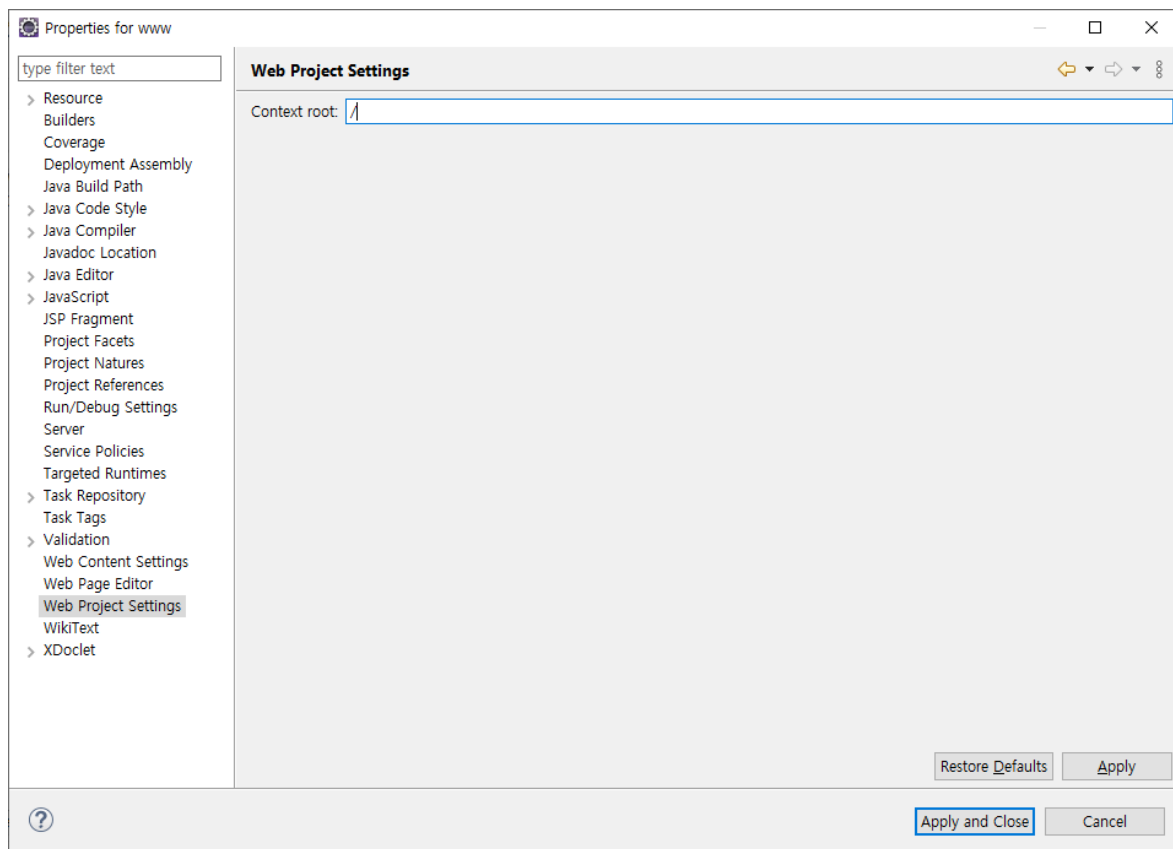
Project facets 1.8버전으로 선택

Problems Servers Properties Console	
11 errors, 251 warnings, 0 others (Filter matched 141 of 292 items)	
Description	Resource
Map cannot be resolved to a type	Main.java
MessagePack cannot be resolved to a type	Main.java
MessagePack cannot be resolved to a type	Main.java
MessagePack cannot be resolved to a type	Main.java
OutputStreamWriter cannot be resolved to a type	ChatServer.java
Project facet Java 16 is not supported by target runtime Apache Tomcat v10.0.	www
ServerSocket cannot be resolved to a type	ChatServer.java
ServerSocket cannot be resolved to a type	ChatServer.java
Socket cannot be resolved to a type	ChatServer.java

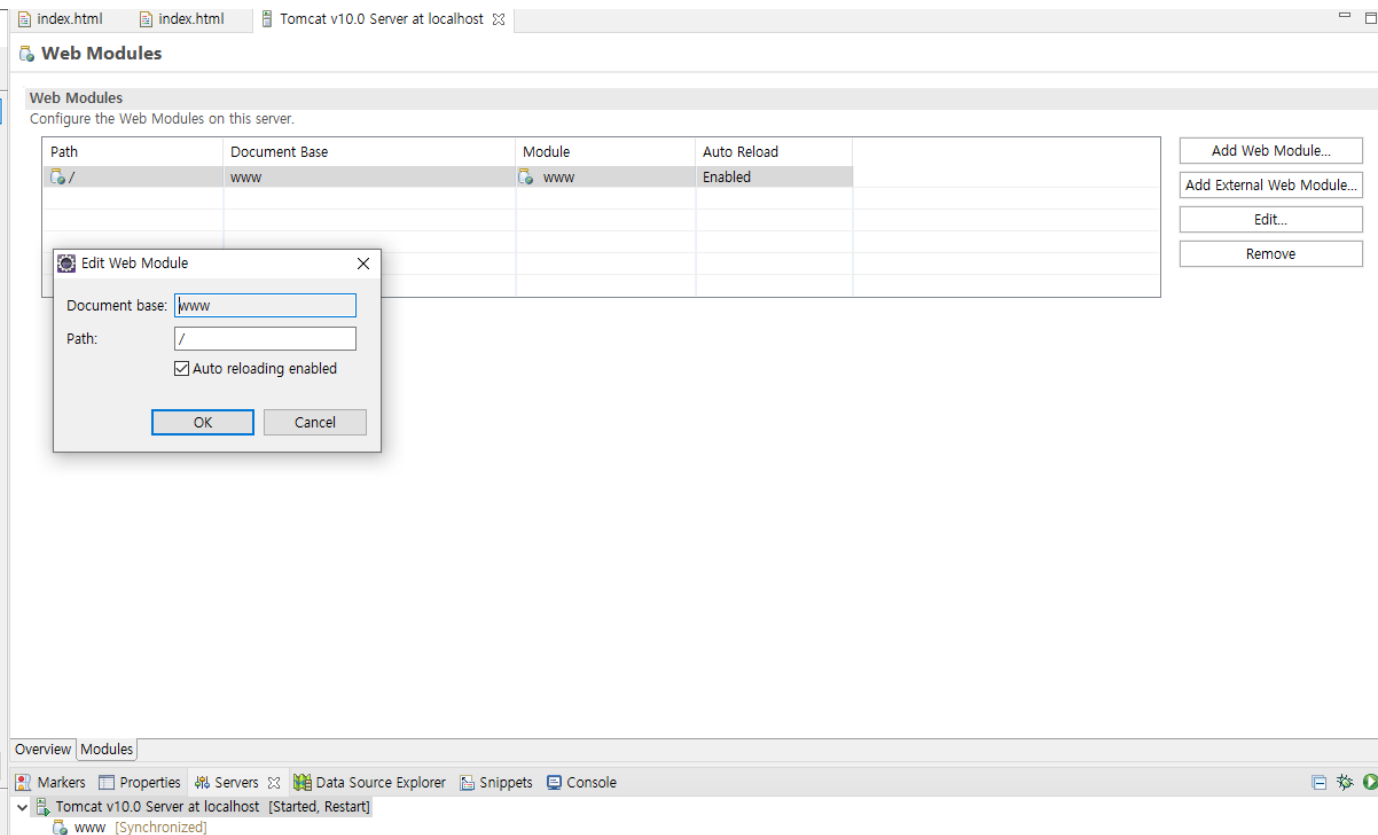


Context 경로 설정 2가지 방법

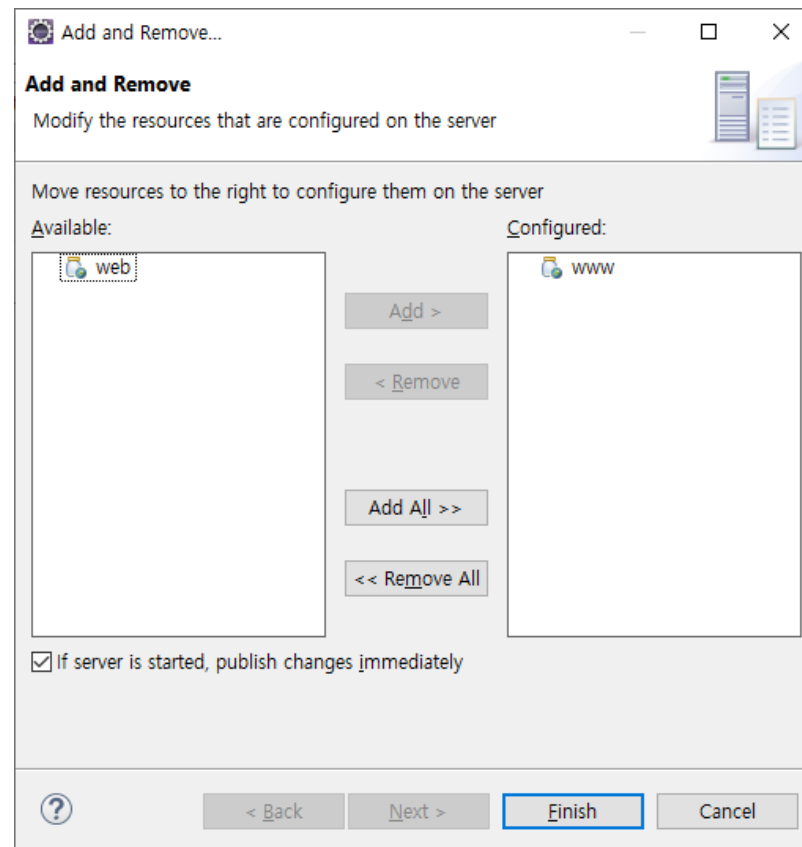
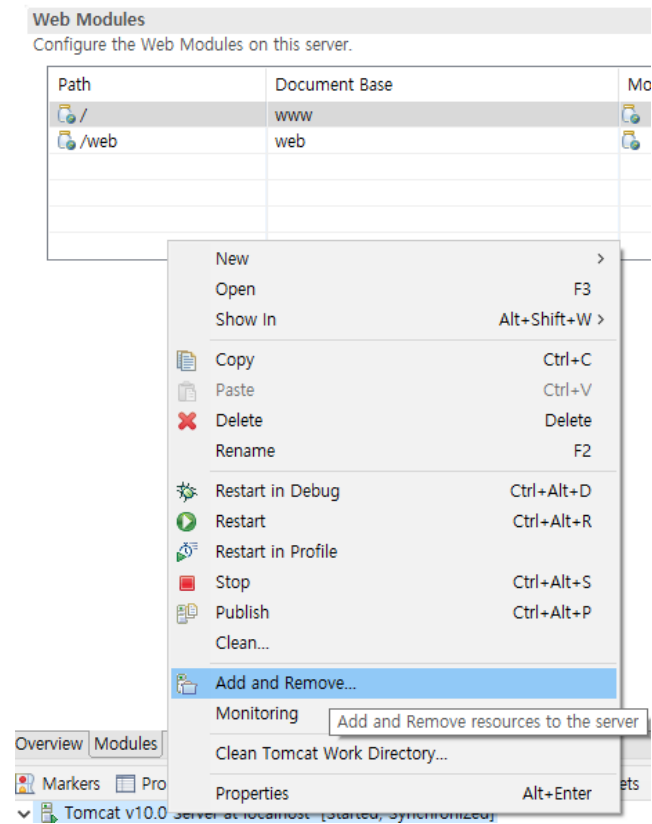
프로젝트의 properties에서 변경



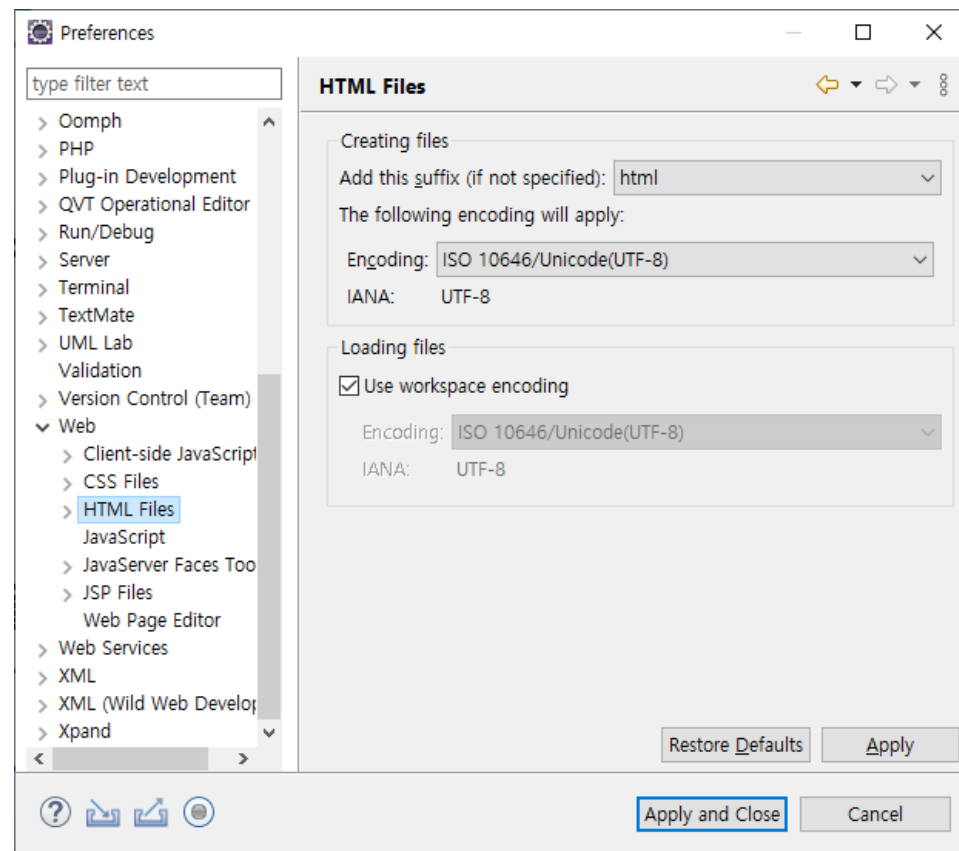
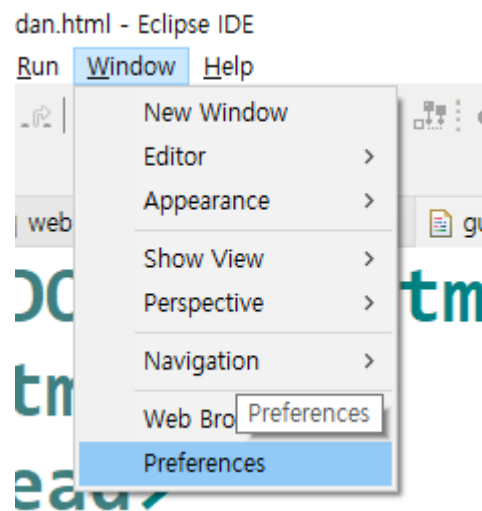
서버의 설정에서 변경



실행중인 서버 추가 제거



새로 만드는 파일을 utf-8로 변경



II. HTML

HTML - 문서의 구조

```
<!DOCTYPE html>  → HTML이 지켜야 하는 표준을 알려주는 DTD
<html lang="ko">  → HTML 문서의 시작(웹 문서 전체를 감싸고 있음)
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta http-equiv="X-UA-Compatible" content="ie=edge" />
  <title>Document</title>
</head>
<body></body>  → 브라우저에 표현되는 내용이 담김
</html>
```

<head> ~ </head> : <title>, <script>, <link>, <meta> 와 같은 태그들이 존재

<meta> : 브라우저에는 표현되지 않지만 현재 HTML의 모든 정보를 요약해서 담고 있는 데이터

❖ 구성요소

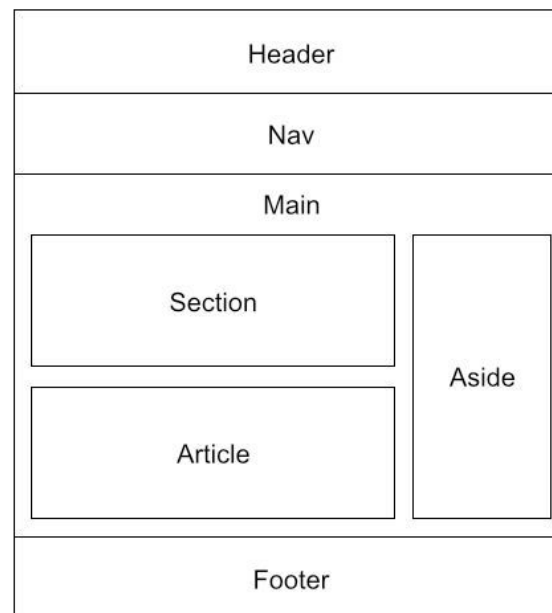
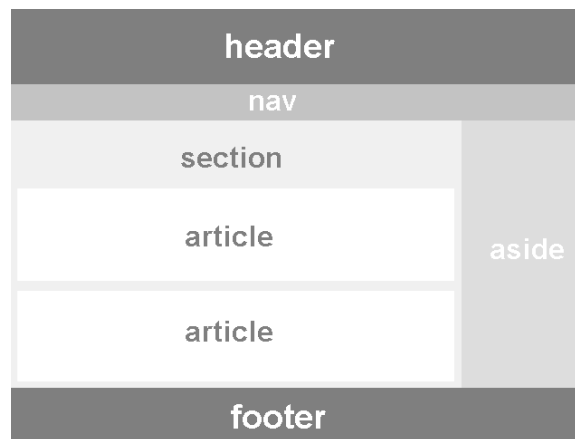
- 태그(Tags) : 문서 중간에 들어가는 일종의 꼬리표로, ‘<’ 기호와 ‘>’ 기호로 둘러싸여 있는 코드
- 속성(Attributes) : 태그의 내부에서 그 태그의 형태를 구체적으로 나타냄
- 변수(Arguments) : 태그를 구성하는 속성의 실제 값

태그	기능
<html> ... </html>	프로그램이 HTML로 작성되어 있다는 것을 나타내는 태그
<head> ... </head>	화면에 표시되는 문서의 제목이나 스타일, 정보 등을 포함하는 태그
<title> ... </title>	화면의 상단에 표시되는 문서의 제목을 나타내는 태그
<body> ... </body>	HTML 문서의 본문이 들어가는 부분을 표시하는 태그

HTML - 레이아웃 TAG

html5 semantic 태그

- 1) <header>
- 2) <nav>
- 3) <section>
- 4) <article>
- 5) <aside>
- 6) <footer>
- 7) <main>



Sementic - header

- 로고, 메뉴, 검색바, 로그인, 로그아웃 버튼, 회원가입 버튼 등이 보여지는 부분



- header 태그 사용 예

```
<header class="page-header">  
  <h1>Cute Puppies Express!</h1>  
</header>
```

- header 태그 잘못된 예

```
<header class="page-header">  
  <h1>Cute Puppies Express!</h1>  
  <header>중첩 헤더</header>  
  <footer>풋터</footer>  
</header>
```

<https://developer.mozilla.org/ko/docs/Web/HTML/Element/header>

Sementic - footer

- 사이트의 하단에 위치, 회사 주소, 저작권 정보, 이메일 주소 등



- footer 태그 사용 예

```
<footer>
  <p>© 2018 Gandalf</p>
</footer>
```

- footer 태그 잘못된 예

```
<footer class="page-footer">
  <h1>Cute Puppies Express!</h1>
  <header>헤더</header>
  <footer>풋터</footer>
</footer>
```

<https://developer.mozilla.org/ko/docs/Web/HTML/Element/footer>

Sementic - main

- 문서 내부의 핵심적인 콘텐츠를 담고 있는 영역
- main 태그 사용 예

```
<header>Gecko facts</header>
```

```
<main role="main">
```

```
  <p>Geckos are a group of usually small, usually nocturnal lizards.  
  They are found on every continent except Australia.</p>
```

```
  <p>Many species of gecko have adhesive toe pads which enable  
  them to climb walls and even windows.</p>
```

```
</main>
```

- IE 지원 불가

Sementic - article

- 독립적으로 구분되거나 재사용 가능한 영역을 설정.
(매거진/신문 기사, 블로그 등)
- 일반적으로 <h1> ~ <h6> 를 포함하여 식별
- 작성일자와 시간을 <time> 의 datetime 속성으로 작성.
- article 태그 사용 예
 - <https://developer.mozilla.org/ko/docs/Web/HTML/Element/article>

Sementic – section

- 문서의 일반적인 영역을 설정.
- 일반적으로 <h1> ~ <h6> 를 포함하여 식별
- section 태그 사용 예
 - ✓ <https://developer.mozilla.org/ko/docs/Web/HTML/Element/section>

Sementic – aside

- 문서의 별도 콘텐츠를 설정.
(보통 광고나 기타 링크 등의 사이드바(Side bar)를 설정)
- aside 태그 사용 예
 - ✓ <https://developer.mozilla.org/ko/docs/Web/HTML/Element/aside>

Sementic – nav

- 다른 페이지 링크를 제공하는 영역을 설정.
(Navigation, 보통 메뉴(Home, About, Contact), 목차, 색인 등을 설정)
- nav 태그 사용 예
 - ✓ <https://developer.mozilla.org/ko/docs/Web/HTML/Element/nav>
 - ✓ https://www.w3schools.com/tags/tag_nav.asp

Sementic – address

- <body>, <article>, <footer> 등에서 연락처 정보를 제공하기 위해 포함하여 사용.

nav 태그 사용 예

- ✓ <https://developer.mozilla.org/ko/docs/Web/HTML/Element/address>
- ✓ https://www.w3schools.com/tags/tag_address.asp

HTML - Div 태그

- Division 의 약자로 분할을 의미
- 문서의 부분이나 섹션을 정의함
- 명확한 의미를 가지지 않기 때문에 특정 범위를 묶는(wrap) 용도로 사용

```
<div>
  <p>
    Lorem ipsum dolor sit, amet consectetur adipisicing elit. Ipsa, nesciunt
    nulla fugit facilis voluptatum quaerat debitis quibusdam facere tenetur
    ullam? Ullam alias obcaecati numquam doloremque molestias dolore,
    tenetur nesciunt amet!
  </p>
</div>
<div>
  <div>
    <h1>제목</h1>
    <p>문단 내용</p>
  </div>
</div>
```

전역 속성(Global Attributes)

- 모든 HTML 요소에서 공통적으로 사용 가능한 속성

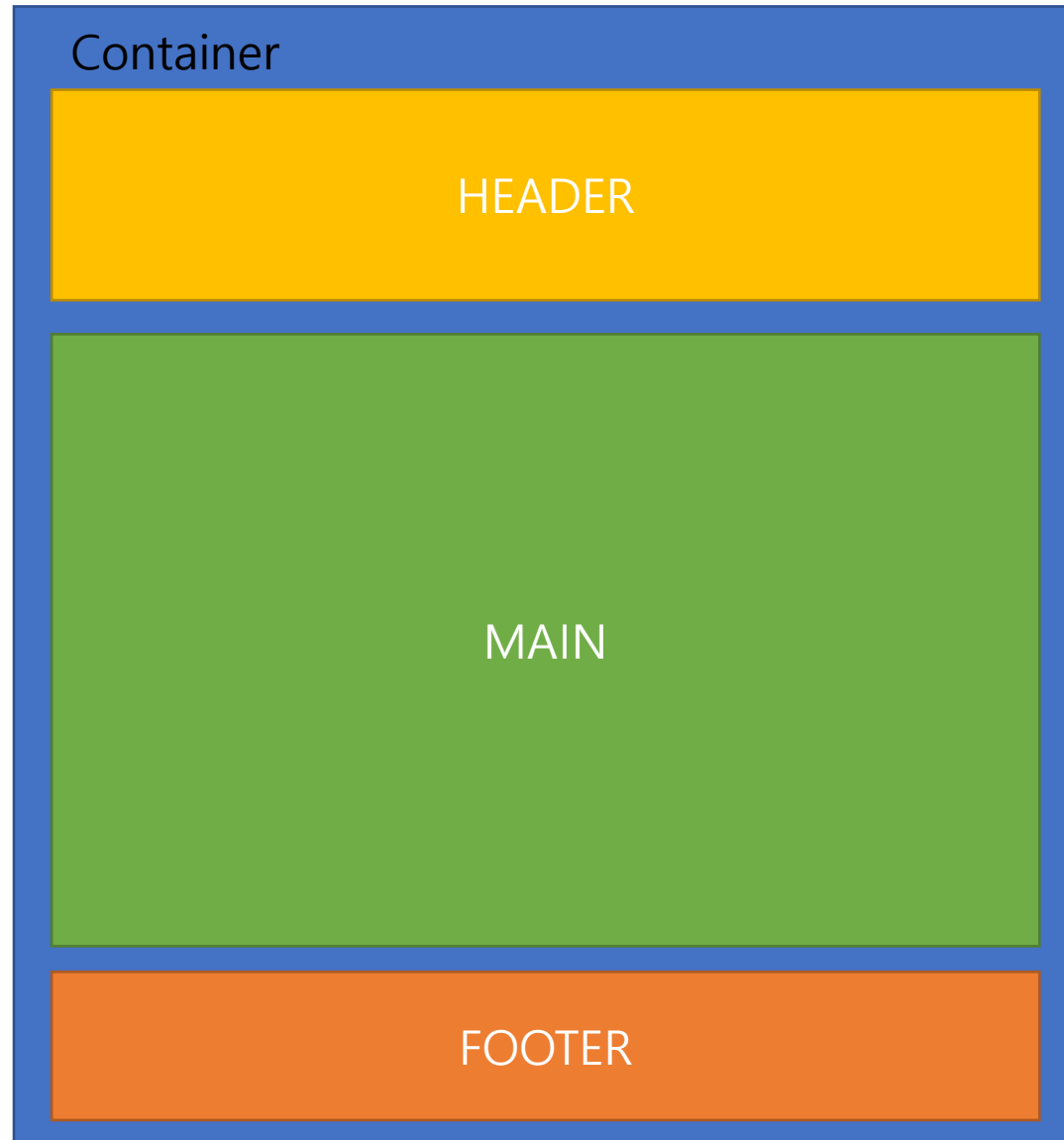
속성	의미
class	공백으로 구분된 요소의 별칭을 지정 CSS 혹은 JavaScript의 요소 선택기를 통해서 요소를 선택하거나 접근
id	문서에서 고유한 식별자(identifier, ID)를 정의. CSS 혹은 JavaScript의 요소 선택기(CSS 선택자 나 GetElementsByClassName , QuerySelectorAll 같은)를 통해서 요소를 선택하거나 접근
style	요소에 적용할 CSS를 선언.
title	요소의 정보(설명)을 지정.
lang	요소의 언어(ISO 639-1)를 지정
data-*	사용자 정의 데이터 속성을 지정. HTML에 JavaScript에서 이용할 수 있는 데이터(정보)를 저장하는 용도로 사용.

전역 속성(Global Attributes)

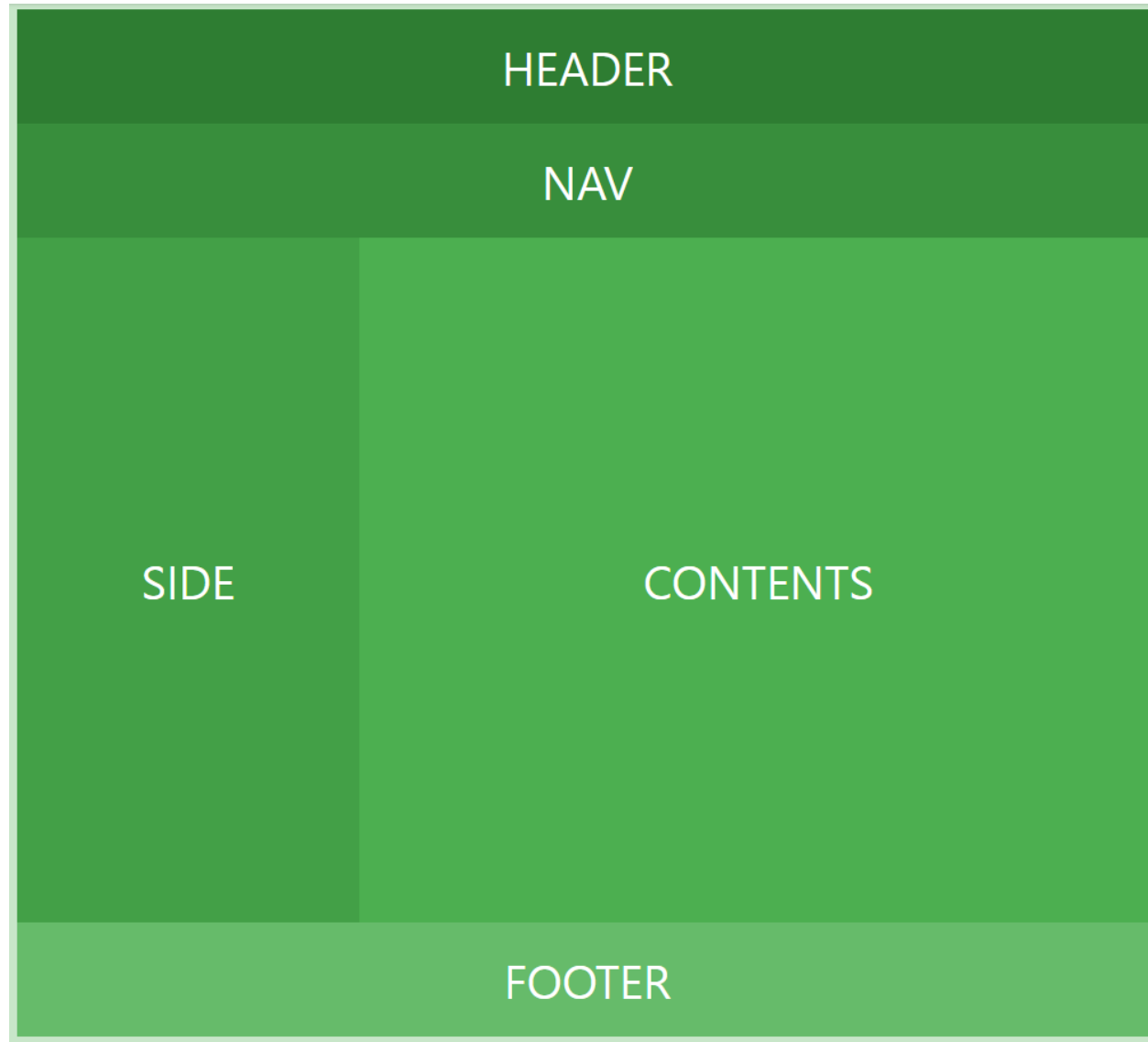
- 모든 HTML 요소에서 공통적으로 사용 가능한 속성

속성	의미
draggable	요소가 <u>Drag and Drop API</u> 를 사용 가능한지 여부를 지정.
hidden	요소를 숨김.
tabindex	Tab키를 이용해 요소를 순차적으로 포커스 탐색할 순서를 지정.

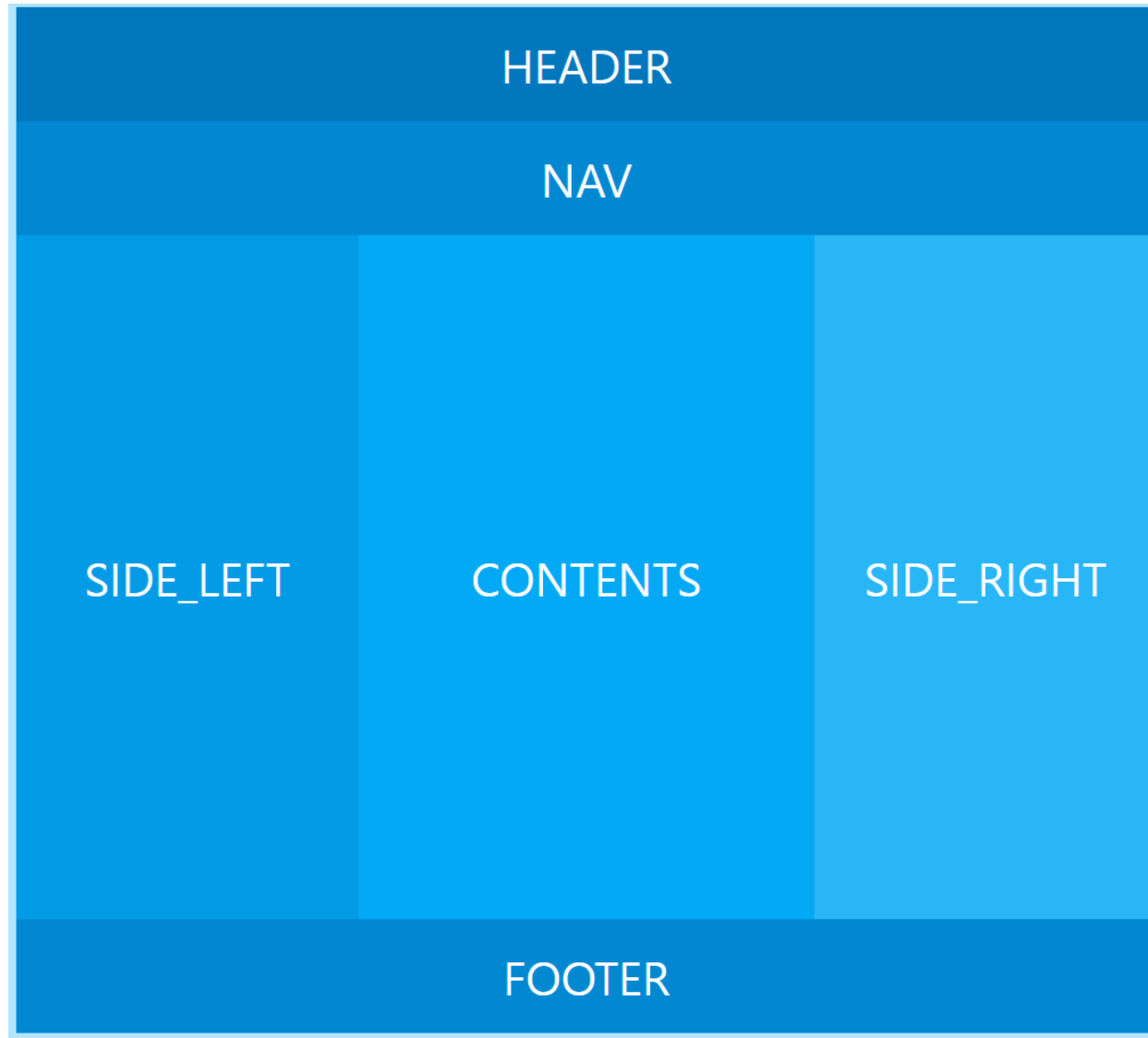
[실습] Semantic LAYOUT



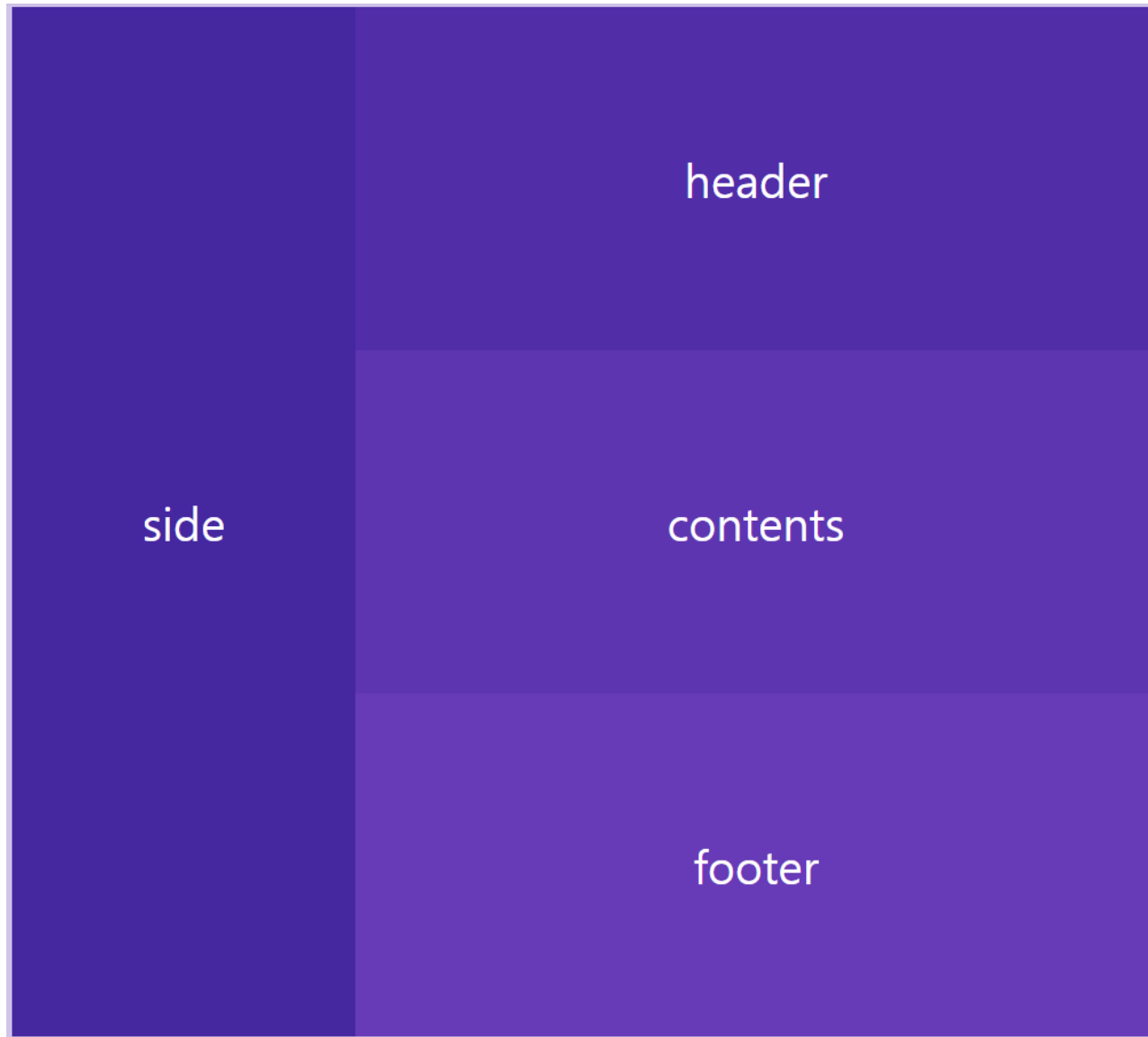
[실습] Semantic LAYOUT



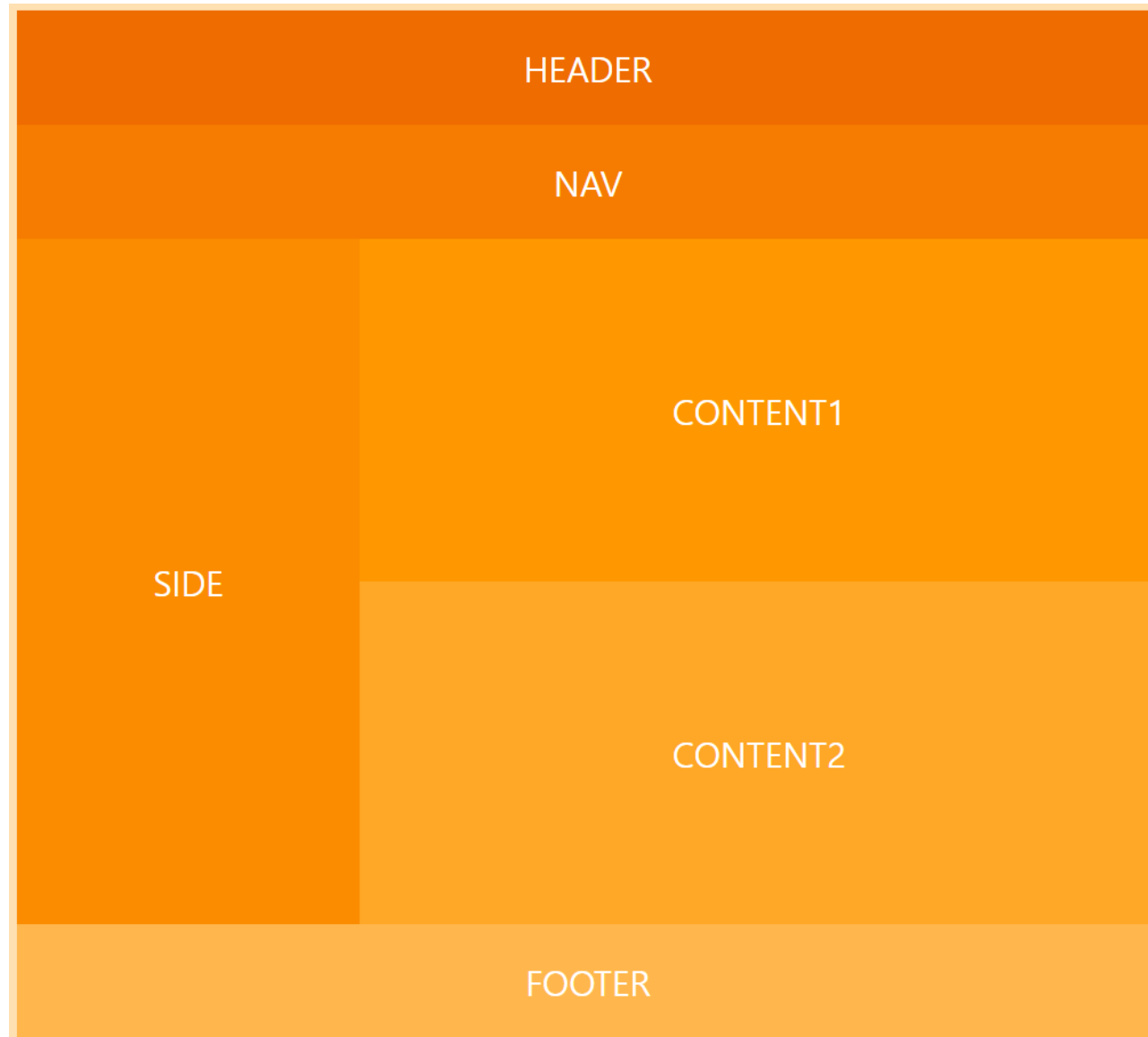
[실습] Semantic LAYOUT



[실습] Semantic LAYOUT



[실습] Semantic LAYOUT



[실습] Sementic LAYOUT

메뉴1 메뉴2 메뉴3

Section

Article 1

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Aperiam, et assumenda quo magni iste officiis doloremque esse sint quisquam non soluta ut obcaecati tempore dolorem sit natus, beatae laudantium ipsam!

Article 2

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Aperiam, et assumenda quo magni iste officiis doloremque esse sint quisquam non soluta ut obcaecati tempore dolorem sit natus, beatae laudantium ipsam!

Article 3

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Aperiam, et assumenda quo magni iste officiis doloremque esse sint quisquam non soluta ut obcaecati tempore dolorem sit natus, beatae laudantium ipsam!

Aside

hong123@gmail.com
[010-1234-5678](tel:010-1234-5678)

[실습] Sementic LAYOUT

logo

MENU

MENU

MENU

MENU

사이트맵

홈

로그인

회원가입

search

HTML – 입력폼 TAG

<input type="" name="" id="" value="">

이름 :

<input type="text">

비밀번호

<input type="password">

좋아하는 과일을 선택하세요

- ☒ 사과
- ☐ 바나나
- ☒ 딸기
- ☒ 포도
- ☐ 수박
- ☒ 키위

<input type="checkbox">

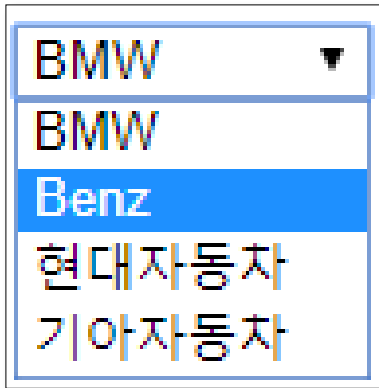
고객의견

<input type="text">

성별 ☐ 남성 ☒ 여성

<input type="radio">

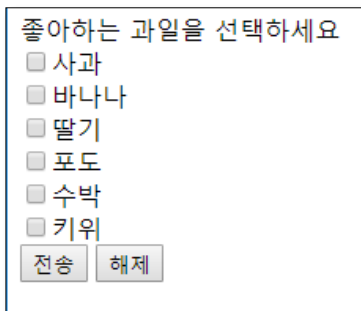
HTML – 입력폼 TAG



BMW ▼
BMW
Benz
현대자동차
기아자동차



파일 선택 선택된 파일 없음



좋아하는 과일을 선택하세요

- ☐ 사과
- ☐ 바나나
- ☐ 딸기
- ☐ 포도
- ☐ 수박
- ☐ 키위

전송 해제

```
<select>  
    <option value="BMW">BMW</option>  
    <option value="자동차">자동차</option>  
</select>
```

```
<form enctype="multipart/form-data">  
    <input type="file" accept="image/jpg,image/gif">  
</form>
```

```
<input type="submit"> : 버튼 + 전송 기능  
<input type="reset"> : 현재 작성된 폼 초기화  
<input type="button"> : 버튼
```

HTML - FORM

- 입력 양식(Form)
 - 아이디와 패스워드를 입력하는 화면처럼, 사용자에게 무엇인가를 입력하게 하는 것을 입력 양식, 혹은 폼(Form)이라고 부름
 - 폼 태그는 <input> 태그를 기본으로 일반 텍스트, 비밀번호, 체크박스, 라디오 버튼 등의 입력 방법을 속성으로 지정 가능

```
<form action="input.jsp" method="post">  
  <input type="text" name="input">  
  <input type="submit">  
</form>
```

method

- ① get : URL 주소 뒤에 파라미터를 붙여서 데이터 전달
- ② post : 사용자가 입력한 데이터를 HTTP Request 헤더에 포함시켜서 전송하는 방식

HTML - FORM

- 입력 양식(Form) - input 태그 속성

속성	기능
type	일반 텍스트, 비밀번호 등과 같은 입력 방식을 결정
name	사용자의 입력 내용의 구별자
size	입력 화면에 나타나는 글 크기를 설정
value	텍스트 입력 상자에 기본적으로 나타나는 텍스트 문장 체크박스나 라디오 버튼에서 입력값이 임시 저장되는 곳
checked	해당 아이템이 기본값으로 설정되어 있음을 표시

HTML - FORM

- 입력 양식(Form) - input 태그 속성

속성	설명
autocomplete	자동으로 입력을 완성한다.
autofocus	페이지가 로드되면 자동으로 입력 포커스를 갖는다.
placeholder	입력 값에 대한 힌트를 희미하게 보여준다.
readonly	읽기 전용 필드임을 나타낼 때 사용
required	입력 양식을 제출하기 위해 반드시 채워져 있어야 함을 나타낸다.
pattern	허용하는 입력의 형태를 정규식으로 지정한다.

HTML - FORM

- type 에 올 수 있는 속성 값

type 속성	설명
text	텍스트를 입력할 수 있는 한 줄 짜리 필드 생성
password	비밀번호를 입력할 수 있는 한 줄 짜리 필드 생성
radio	라디오 버튼 생성
checkbox	체크 박스 생성
file	파일 이름을 입력하는 필드 생성
reset	초기화 버튼 생성, 버튼을 누르면 모든 입력 필드가 초기화 됨
image	이미지를 전송 버튼으로 만든다
hidden	사용자에게는 보이지 않지만 서버로 전송된다.
submit	제출 버튼 생성

HTML - FORM

<input type="text">

- ① 사용자가 입력할 수 있는 한 줄 짜리 입력 필드를 정의
- ② 텍스트의 기본 크기는 20글자

<input type="password">

- ① 패스워드 필드를 정의
- ② 사용자가 입력한 글자는 보이지 않는다

이름 :

비밀번호

HTML - FORM

<input type="checkbox">

- checkbox는 사용자에게 여러 개의 체크를 입력하는 화면을 제공하는 속성
- type 속성에 checkbox가 설정되면 사용자가 체크한 결과가 value로 지정된 변수에 저장되고, 이 변수 값은 프로그램에서 사용

좋아하는 과일을 선택하세요

- ☒ 사과
- ☐ 바나나
- ☒ 딸기
- ☒ 포도
- ☐ 수박
- ☒ 키위

<input type="radio">

- radio는 제공되는 여러 개의 선택 사항 중 오직 1개의 체크만을 받아들일 때 사용되는 속성

성별 ☐ 남성 ☒ 여성

HTML - FORM

<input type="textarea">

- 여러 줄의 텍스트를 입력 받을 때 사용하는 태그
- 사용자가 초기 공간보다 더 많은 텍스트를 입력하면 자동으로 스크롤바가 생성됨

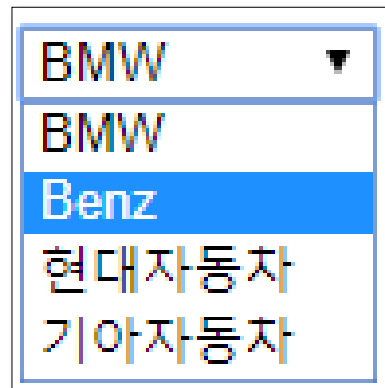


A screenshot of a web form. It features a label '고객의견' (Customer Opinion) in a small box at the top left. Below the label is a large, empty text area with a thin border. A vertical scrollbar is visible on the right side of the text area, indicating that it can handle more text than the initial visible space.

HTML - FORM

```
<select>  
    <option value="BMW">BMW</option>  
    <option value="자동차">자동차</option>  
</select>
```

- 메뉴를 표시하고 사용자로 하여금 선택하게 한다
- <option> 요소와 함께 사용되며, <option> 요소는 value 속성을 가지고 있다.
- selected 속성을 사용해 특정 항목을 초기에 선택할 수 있다.



HTML - FORM

```
<form enctype="multipart/form-data">  
  <input type="file" accept="image/jpg,image/gif">  
</form>
```

- 사용자가 파일을 선택해서 서버로 업로드 하는 경우에 사용



```
<input type="hidden" name="email" value="funcom@gmail.com">
```

- 브라우저 창에는 표시되지 않지만 폼 안에는 포함되어 있는 필드를 말함
- 사용자가 입력하는 데이터가 아님
- 클라이언트 컴퓨터가 서버로 특정한 데이터를 전송하고 싶을 때 사용

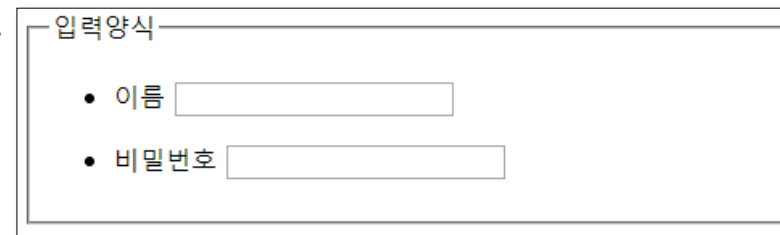
HTML - FORM

```
<label for="username">이름</label>  
<input type="text" id="username">
```

- 텍스트 설명글과 직접 내용을 연결해야 할 텍스트 필드가 서로 연관되어 있음을 알려줌

```
<fieldset>  
  <legend>입력양식</legend>  
  <label for="username">이름</label>  
  <input type="text" id="username">  
</fieldset>
```

- <fieldset> : 여러 태그들을 하나의 그룹으로 묶어주는 태그
- <legend> : 그룹에 제목을 붙일 때 사용



입력양식

- 이름
- 비밀번호

HTML - FORM

<input type="submit"> : 버튼 + 전송 기능

<input type="reset"> : 현재 작성된 폼 초기화

<input type="button"> : 버튼

좋아하는 과일을 선택하세요

- ☐ 사과
- ☐ 바나나
- ☐ 딸기
- ☐ 포도
- ☐ 수박
- ☐ 키위

물품가격 :

수량 :

HTML - 추가된 FORM 양식

type	설명
date	날짜를 입력할 수 있는 컨트롤
datetime	UTC 날짜/시간 형식을 이용한 날짜와 시간 표시 컨트롤
datetime-local	현지 날짜/시간
month	월/연도
time	시간
week	주와 연도를 선택할 수 있는 컨트롤
color	색상 코드를 입력할 수 있는 컨트롤
email	표준 이메일 주소를 입력받아서 검증하는 컨트롤
tel	전화번호를 입력받아서 검증하는 컨트롤
search	검색어 입력 양식을 생성
range	2개의 숫자를 선택할 수 있는 슬라이더 컨트롤
number	숫자만 입력받는 컨트롤
url	Url만 입력받는 컨트롤

HTML - 자주사용하는 TAG

- 1) 리스트
- 2)
- 3) <div style="border:1px solid red">
- 4)
- 5) <table border>

HTML - 리스트

- ul 태그

```
<ul>  
  <li>첫째  
  <li>둘째  
  <li>셋째  
</ul>
```

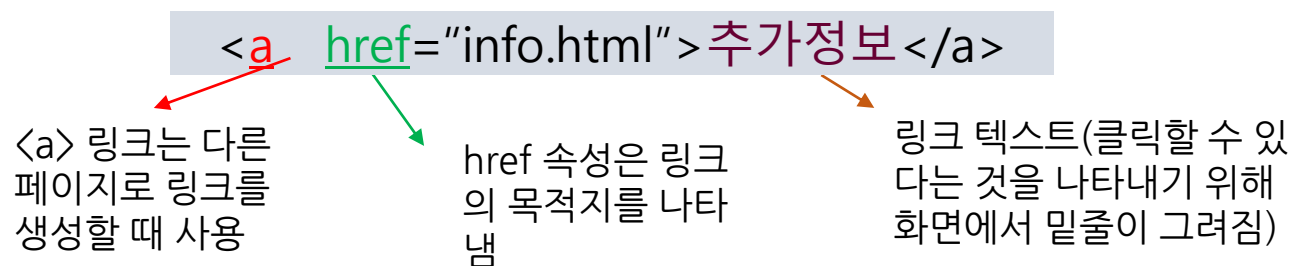
- ol 태그

```
<ol>  
  <li>모닝  
  <li>런치  
  <li>이브닝  
</ol>
```

HTML - a

- a 태그
 - 하이퍼텍스트는 사용자의 선택에 따라 관련된 특정 정보로 이동할 수 있도록 조직된 문서를 의미하는 것으로, 브라우저에서 특정 링크를 클릭하면 해당 사이트로 이동
 - 속성

속성	기능
name	같은 HTML 문서 내에서 이동할 곳의 이름을 지정
href	이동할 다른 문서의 이름을 지정
target	새롭게 열리는 웹 브라우저의 이름



HTML - a

- 하이퍼링크 기본 색상
 - ① 청색 : 방문하지 않은 링크는 밑줄이 그어져 있고 청색
 - ② 보라색 : 방문한 링크는 밑줄이 그어져 있고 보라색
 - ③ 빨간색 : active link 는 밑줄이 그어져 있고 빨간색
- 하이퍼링크 경로 지정

범위	예제	설명
절대경로	href="http://www.google.com	다른 웹 사이트의 페이지
상대경로	href="../doc/info.html	웹 사이트 안에서의 다른 페이지
내부 파일	href="#anchor1"	현재 페이지 안의 다른 위치

HTML - a

- 하이퍼링크 target 속성

Target	예제
_blank	새로운 윈도우에서 새로운 페이지를 연다.
_self	현재 윈도우에 새로운 페이지를 적재한다.
_parent	부모 프레임에 새로운 페이지를 적재한다.
_top	현재 프레임에 새로운 페이지를 적재하고 모든 프레임을 취소

- 주요 예제

```
<a href="mailto:hong@google.com?subject=feedback">hong@google.com</a>
```

```
<a href="http://www.company.com/data.zip">다운로드</a>
```


HTML - 이미지

- 이미지를 HTML 문서에 삽입할 때는 태그를 사용
- 외부파일을 HTML에 불러들이는 것이므로 사이즈가 큰 이미지의 사용이 나 과도한 호출은 HTML의 실행 속도를 느리게 함
- 속성

속성	기능
src	이미지의 위치
href	이미지 클릭 시 하이퍼링크 주소
alt	이미지 캡션

HTML - 테이블

- 표 형태의 데이터를 표시하는 데 사용
- 행(Row)은 ‘줄’이라고 하며 글이나 그림이 들어갈 수 있는 위에서 아래로 연결된 사각형 공간을 말함
- 열(Column)은 ‘칸’이라고 하며 왼쪽에서 오른쪽으로 연결되는 공간을 말함
- 표를 생성하는 태그는 <table>을 비롯하여 <tr>, <td>, <th>가 있음

영화제목	연도	감독	평가
라이프 오브 파이	2013	이안	8.68
레미제라블	2012	토머스퍼	8.28
장고:분노의 추적자	2012	타란티노	8.29

[실습] 로그인

Login

이메일 주소

비밀번호

☐ 로그인유지

로그인

[회원가입](#) ID/PW찾기

[실습] 회원가입

회원가입	
아이디	<input type="text"/> 중복확인
비밀번호	<input type="password"/>
비밀번호 확인	<input type="password"/> *비밀번호를 다시 입력하여 주세요.
성명	<input type="text"/>
성별	<input type="radio"/> 남 <input type="radio"/> 여
이메일	<input type="text"/> - <input type="text"/>
주소	<input type="text"/> 검색
상세주소	<input type="text"/>
휴대폰	<div><div><input type="radio"/> <input type="radio"/> <input type="radio"/></div><div>010 ▼ - <input type="text"/> - <input type="text"/></div></div>
관심	<input type="checkbox"/> 수입차 <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
자기소개	<div><div></div></div>
<div>회원가입 취소</div>	

[실습] 게시판

> 게시판 목록

<div>게시판명 <input type="text"/></div> <div>조회 등록</div>				
번호	게시판명	등록자	등록일	조회수
21	게시판2	테스트1	20170910	0
22	게시판2	테스트1	20170831	0
23	게시판2	테스트1	20170615	0
24	게시판2	테스트1	20160920	0
25	게시판2	테스트1	20160920	0
26	게시판3	테스트1	20160912	0
27	게시판2	테스트1	20160801	0
28	게시판4	테스트1	20160727	0
29	게시판5	테스트1	20160727	0
30	게시판6	테스트1	20160707	0

1 2 3 4 5

IV. CSS

CSS - 구조

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <style>
    * { background-color: red;}
  </style>
</head>
<body>

</body>
</html>
```

CSS 적용방법

```
div { color : blue; }
```

↑
선택자
(selector)

↑
속성
(property)

↑
값
(value)

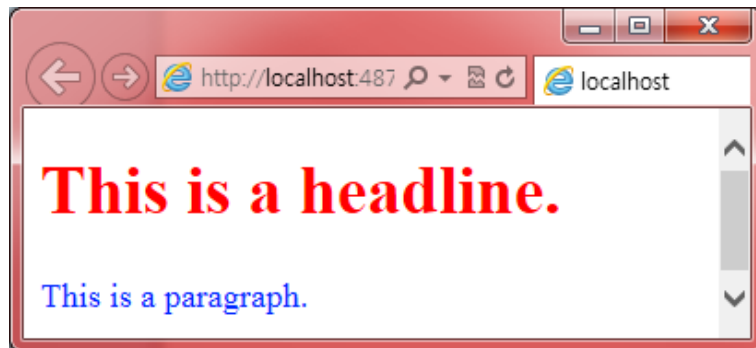
- 선택자 : 스타일 적용 대상자
- 속성 : 변경하고자 하는 부분
- 값 : 변경 값

CSS 선언 방식

1) 인라인 스타일 시트(Inline Style Sheet)

- ① 스타일 시트가 겹칠 때 **제일 높은 우선 순위가 부여**
- ② 각각의 HTML 요소마다 스타일을 지정함

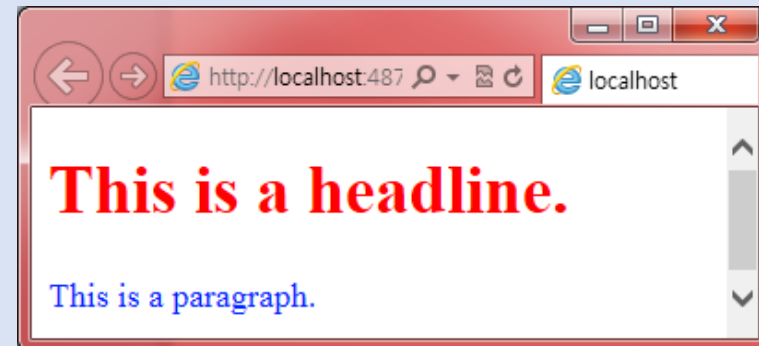
```
<h1 style="color: red">This is a headline.</h1>  
<p style="color: #0026ff">This is a paragraph.</p>
```



CSS 선언 방식

2) 내부 스타일 시트(Internal Style Sheet)

```
<head>
  <style>
    h1 { color : red;    }
    p  { color : #0026ff; }
  </style>
</head>
<body>
  <h1>This is a headline.</h1>
  <p>This is a paragraph.</p>
</body>
```

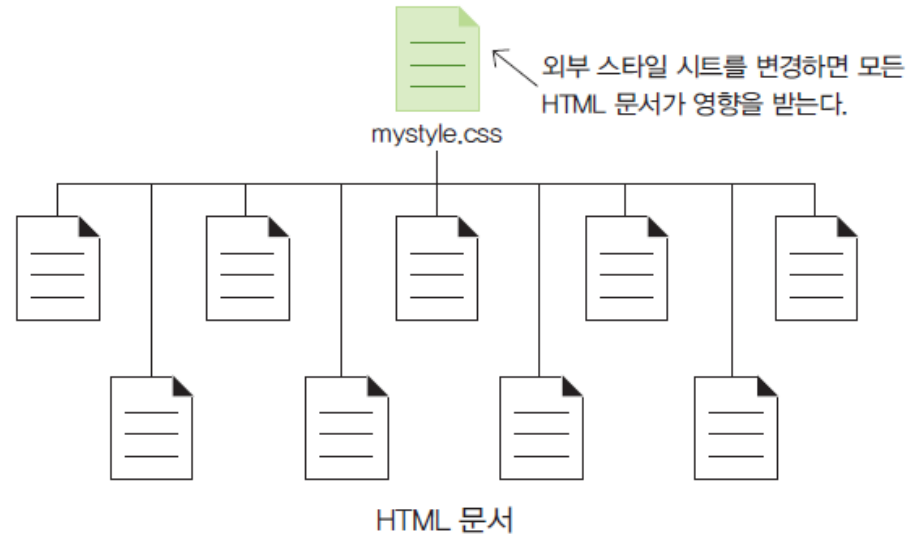


CSS 선언 방식

3) 외부 스타일 시트(External Style Sheet) - 추천

- ① 외부 스타일 시트는 스타일 시트를 외부에 파일로 저장하는 것
- ② 많은 페이지에 동일한 스타일을 적용하려고 할 때 좋은 방법

```
<link rel="stylesheet" href="mystyle.css">
```



CSS 선택자

- ① 태그 선택자
- ② 전체 선택자
- ③ 클래스 선택자
- ④ 아이디 선택자
- ⑤ 자손 선택자
- ⑥ 자식 선택자
- ⑦ 인접형제 선택자
- ⑧ 그룹 선택자
- ⑨ 속성 선택자
- ⑩ 가상 선택자
- ⑪ 의사 요소 선택자
- ⑫ 구조 선택자(child, nth-child)
- ⑬ 형태 구조 선택자
- ⑭ 부정 선택자

CSS – 선택자(Selector)

1. 태그 선택자

```
h1 { color : green; }
```

☞ <body> 태그 안에 선언된 모든 h1 에 css 적용

2. 전체 선택자

```
* { color : green; }
```

☞ 페이지 안 모든 요소를 선택할 때 사용(모든 요소에 공통적인 속성 지정)

CSS – 선택자(Selector)

3. 클래스 선택자

ex) `.target { color : red; }`

`<p class="target">`이곳에 빨간색으로 지정됩니다.`</p>`

☞ `<body>` 태그 안에 class 가 target 인 태그만 css 적용

CSS – 선택자(Selector)

4. 아이디 선택자

ex) `#special { color : red; }`

`<p id="special">`id 가 special 인 단락입니다`</p>`

`<p>` 문단 태그 `</p>`

☞ `<body>` 태그 안에 id가 special 인 태그만 css 적용

CSS – 선택자(Selector)

5. 자손 선택자

```
div h2 { color : blue; } /* div 하위에 모든 h2 요소*/
```

6. 자식 선택자

```
div > h2 { color : blue; } /* div h2 자식 요소*/
```

CSS – 선택자(Selector)

7. 인접형제 선택자

```
div h1 + h2 { color : blue; }  
div h1 ~ h2 { color : blue; }
```

8. 그룹 선택자

```
#item, nav, h2 {  
    text-decoration : underline;  
}
```

CSS – 선택자(Selector)

9. 속성 선택자

```
h1[title] { color : blue; } /* title 속성을 가지고 있는 h1 요소*/  
p[class="example"] { color:blue;} /*class 속성이 example인 p 요소*/
```

10. 가상 클래스 선택자

```
a:link { color : blue; }  
a:visited{ color:green;}  
a:hover{ color: green;}
```

CSS – 선택자(Selector)

11. 의사요소 선택자

```
h1:before{  
  content : “”;  
  color : red;  
}
```

```
<h1>  
</h1>
```

before

내용

```
<h1>  
</h1>
```

내용

after

12. 구조 선택자

```
li:first-child{ text-decoration : underline; }  
li:last-child { background-color: silver; }
```

CSS – 선택자(Selector)

13. 형태 구조 선택자

```
li:first-of-type { text-decoration : underline; }  
li:last-of-type { background-color: silver; }
```

14. 부정 선택자

```
H1{  
    color : black;  
}  
:not(h1){  
    color: red;  
}
```

CSS – 속성

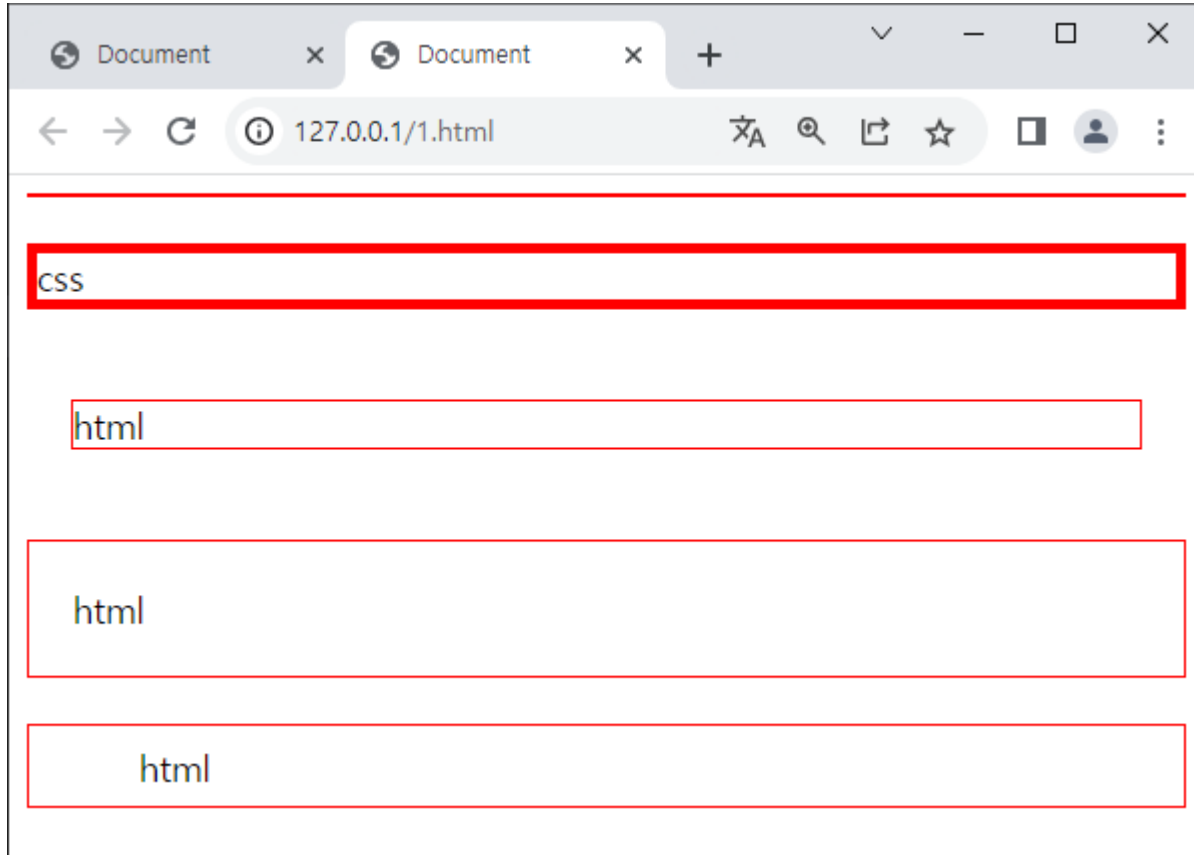
		A	B	C	D	E	F
border	1		선	선의 굵기	선의 색깔	선의 모양	
	2	테두리					
	3						
font Color Text-align	4		모양	크기	굵기	색	위치
	5	글자	CSS	CSS	css	CSS	CSS
	6						
background	7		색	모양			
	8	배경	CSS	CSS			

CSS – display

Inline / block / inline-block / flex

- 1) 인라인 요소 : 내용물의 크기에 맞게 border가 구성, width/height설정이 안됨
- 2) 블록요소 : 한 줄을 모두 갖는 구성, width/height설정됨
- 3) 인라인 블록 : 내용물의 크기에 맞게 구성되지만 width/height설정됨

CSS – box(margin/border/padding)



```
<style>
```

```
div { border:1px solid red }
```

```
</style>
```

```
<div></div>
```

```
<div style="border:5px solid red">css</div>
```

```
<div style="margin:20px">html</div>
```

```
<div style="padding:20px">html</div>
```

상하/좌우 설정

```
<div style="padding:8px 50px">html</div>
```

CSS – position

- html 요소들을 좌우로 배치하는 것에서 벗어나 기준위치 내에서 자유롭게 배치하고자 할 때 사용

위치	설명
position : static	정상적인 흐름에 따른 배치(기본값)
position : relative	원래 자신이 위치해야 할 곳을 기준으로 이동 top/left/right/bottom 으로 설정
position : absolute	기준점에 따라서 특별한 위치에 위치함. top/left/right/bottom 으로 설정
position : fixed	무조건 화면 을 기준으로 위치를 잡을 때(고정)

CSS – float

- html 요소들을 좌우로 배치할 때 적용
- float 특징
 - ① left, right 만 가능.
 - ② float 가 적용된 요소가 해당 방향으로 이동.
 - ③ float가 적용된 요소의 너비가 해당 요소의 내용의 크기만큼 자동으로 인지됨
 - ④ float 이 적용된 요소를 뒤 요소들이 둘러싸아 보여짐
 - ⑤ float 속성을 해제하기 위해 clear 필요함.(자식요소 모두 float 가 적용된 상태인 경우 부모 요소의 높이가 잡히지 않음)

overflow

① overflow: visible(기본속성)

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Exercitationem voluptatem, quasi ipsum aspernatur quos hic officia quae modi autem, ullam voluptas repudiandae nobis! Qui vero iusto et cum animi perspiciatis minus natus optio facere reiciendis, rem eveniet nobis at, ipsum explicabo eligendi! Libero tempore earum dolores officia animi, possimus ipsum cupiditate, amet consequuntur blanditiis excepturi totam velit voluptatem

② overflow : hidden

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Exercitationem voluptatem, quasi ipsum aspernatur quos hic officia quae modi autem, ullam voluptas repudiandae nobis! Qui vero iusto et cum animi perspiciatis minus natus optio facere reiciendis, rem eveniet nobis at, ipsum explicabo eligendi! Libero

③ overflow : scroll

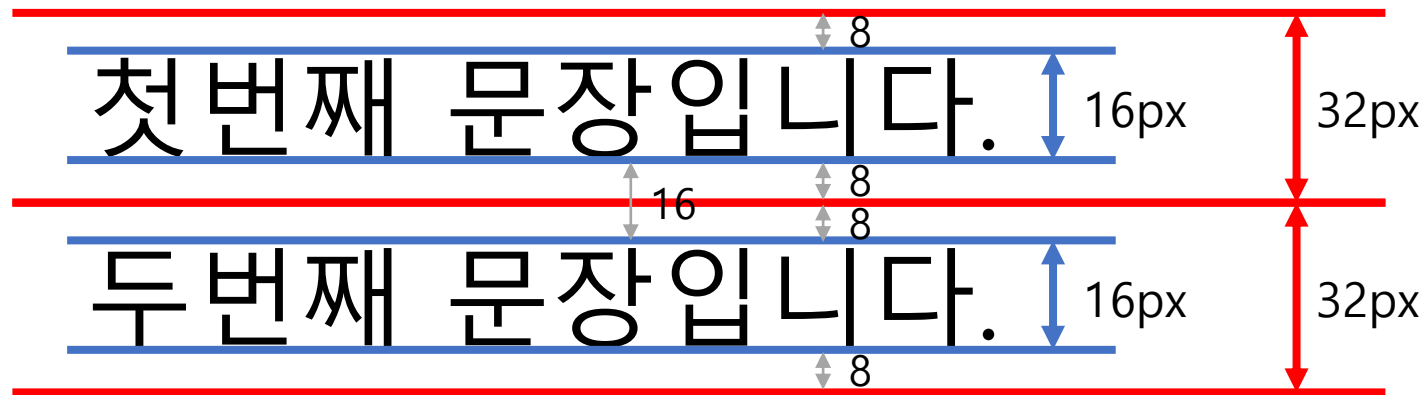
Lorem ipsum, dolor sit amet consectetur adipisicing elit. Exercitationem voluptatem, quasi ipsum aspernatur quos hic officia quae modi autem, ullam voluptas repudiandae nobis! Qui vero iusto et cum animi perspiciatis minus natus optio

④ overflow : auto

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Exercitationem voluptatem, quasi ipsum aspernatur quos hic officia quae modi autem, ullam voluptas repudiandae nobis! Qui vero iusto et cum animi perspiciatis minus natus optio facere reiciendis, rem eveniet nobis at, ipsum explicabo eligendi! Libero

[참고] line-height

- line-height : 32px; 일 때



[참고] Text 스타일

속성	설명	사용할 수 있는 값
font-size	글자 크기	키워드(xx-small ~ xx-large), 상대크기, 수치
font-weight	글자 굵기	normal, bold, 수치(100~900)
text-decoration	텍스트에 밑줄	none, underline, overline, line-through
font-style	이탤릭체	normal, italic(oblique)
text-align	텍스트 정렬	left, right, center, justify
letter-spacing	글자 간격	수치
word-spacing	단어 간격	수치
line-height	단락의 줄 간격	수치, 백분율
text-transform	텍스트 변형	Uppercase, lowercase, capitalize
font-variant	작은 대문자	normal, small-caps
text-indent	첫 글자 들여쓰기	수치, 백분율

[실습] 로그인/회원가입/게시판 메뉴 만들기

[실습] 로그인/회원가입/게시판 메뉴 페이연결하기

V. JAVASCRIPT

JAVASCRIPT - 구조

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <script>
    var i=100;           변수 / 배열 / 객체
    let str="hello";
    const fruit=["apple","banana","mango"];
    obj={name:'홍길동', age:30, favorite:fruit, intro:function(){alert('my name is '+this.name)}};

    function insa(){     함수 선언
      alert('hello');
    }

    insa();              실행

    obj.intro();          객체 함수 실행

    document.write(fruit+"<br>");  내장 객체 함수 실행
  </script>
</head>
```


JavaScript 사용하기

1. 내부 스크립트 : head나 body 안에 삽입 가능

```
<head>  
  <script>  
    alert('Hello World');  
  </script>  
  <title>Document</title>  
</head>
```

2. 외부 스크립트

```
<head>  
  <script src="myscript.js"></script>  
</head>
```

3. 인라인 스크립트

```
<button type="button" onclick="alert('안녕하세요!')">버튼 클릭</button>
```

JavaScript 사용하기

❖ 기본 문법

- 각 문장의 끝에는 ;(세미콜론) 붙임
- 블록 사용 가능

```
{  
    document.write("Hello World!");  
    document.write("How are you");  
}
```

- 유니코드 지원
- 대소문자 구별
- 주석문 : 자바랑 동일

변수(Variable)

❖ 변수 이름 규칙

- 변수 이름은 문자로 시작
- 변수 이름은 \$나 _로 시작
- 변수 이름은 대소문자 구별
- 값의 유형을 지정하는 타입은 없음

```
var name;  
name="홍길동"
```

```
var x=10;  
var x = undefined;  
var x=[];  
var x={}  
var x='aaa';  
var x="aaa";
```

```
let name;  
name="홍길동"
```

```
let x1=10;  
let x2 = undefined;  
let x3=[];  
let x4={}  
let x5='aaa';  
let x6="aaa";
```

배열

- 자바스크립트에서 배열은 객체로 다루어짐
- 배열 생성

① 리터럴로 배열 생성

```
const fruits=["apple", "banana"," peach"]
```

② Array 객체로 배열 생성

```
const fruits=new Array();  
fruits[0] = "Apple";  
fruits[1] = "Banana";  
fruits[2] = "Peach"
```

```
const fruits = new Array("Apple","Banana", "Peach");
```

객체

- 객체 생성

```
const product = {  
  pname : '7D 건조 망고',  
  type : '당절임',  
  ingredient : '망고, 설탕, 치자황색소',  
  origin : '필리핀'  
}
```

조건문

형식	if(조건식) 문장;
설명	조건식이 참인 경우에만 문장이 실행된다.
형식	if(조건식) 문장1; else 문장2;
설명	조건식이 참이면 문장 1 수행. 거짓인 경우 문장 2가 실행.
형식	switch(제어식) { case 'A' : alert("참 잘했어요!"); break; case 'B' : alert("좋은 점수군요!"); break; case 'C' : alert("괜찮은 점수군요!"); break; default: alert("알 수 없는 학점입니다."); break; }
설명	조건이 여러 개인 경우 사용

반복문

형식	<code>while(조건식) 문장;</code>
설명	조건식이 참인 경우에만 문장이 반복된다.
형식	<code>for(초기식;조건식;증감식) 반복 문장1;</code>
설명	초기식을 실행한 후에 조건식의 값이 참인 동안 반복문장을 반복한다. 한 번 반복이 끝날 때마다 증감식이 실행된다
형식	<code>do{ 반복문장; }while(조건식);</code>
설명	일단 문장을 한 번 실행하고 나서 조건을 검사하고 싶을 때 사용

함수

```
function 함수이름( ){  
    함수몸체;  
}
```

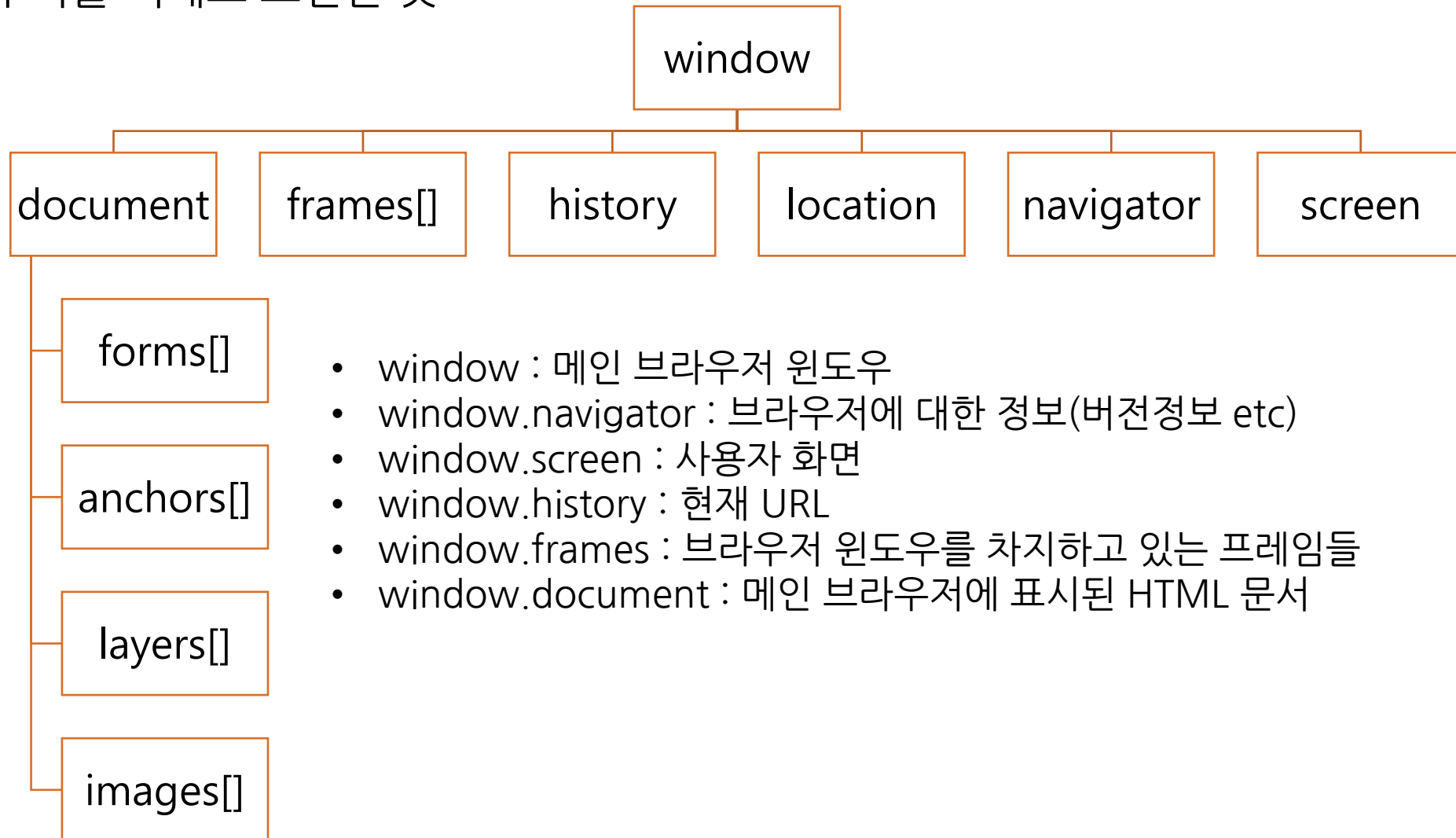
```
function 함수이름(a, b){  
    함수 몸체;  
}
```

```
<script>  
    function showDialog(){  
        alert("안녕하세요");  
    }  
</script>
```

```
<script>  
    function showDialog(greeting,name){  
        alert(greeting+"!" +name);  
    }  
</script>
```


브라우저객체모델(BOM)

- 웹 브라우저를 객체로 표현한 것



window 객체

- 웹 브라우저 윈도우
- 모든 전역 자바스크립트 객체, 함수, 변수는 자동적으로 window 객체의 멤버임

```
alert("Hello World!!");
```

=

```
window.alert("Hello World!!");
```

- 자주 사용되는 메소드
open()

```
window.open(URL, name, specs, replace);
```

- ① URL : 오픈할 페이지의 URL
- ② name : 타겟을 지정하거나 윈도우의 이름(_blank, _self, _top...)
- ③ specs : 여러가지 속성들(창의 크기, 메뉴바, 스크롤바..)
- ④ replace : 히스토리 리스트에서 새로운 엔트리인지 현재 엔트리를 대체하는지 여부

window객체

- alert() 함수 : 사용자에게 경고를 하는 윈도우를 띄우는 함수

```
<script>  
    alert("이것은 경고창입니다.");  
</script>
```

- confirm() 함수 : 사용자에게 어떤 사항을 알려주고 확인이나 취소를 요구하는 윈도우를 화면에 띄운다.

```
<script>  
    var user=confirm("confirm()은 사용자의 답변을 전달합니다.");  
</script>
```

- prompt() 함수 : 사용자에게 어떤 사항을 알려주고 사용자가 답변을 입력할 수 있는 윈도우를 화면에 띄운다

```
<script>  
    var age=prompt("나이를 입력하세요","만나이로 입력합니다.");  
    document.write(age);  
</script>
```

document 객체

- `document.write()` : HTML 문서에 쓰기
- `document.getElementById("id명")` : id로 선택 태그 객체 선택
- `document.getElementById("main").innerHTML='hello'`
- `document.getElementById("main").text='hello'`
- `document.getElementById("id").value='hello'`
- `document.querySelector("태그명")` : tag명, id(#), class(.) 명을 입력하여 객체 선택
- `document.form[0].id.value` : id를 통해 태그의 value 속성 접근

document 객체

```
<head>
<script>
  document.forms[0].id.value='hello'
</script>
</head>
<body>
  <form action="">
    <input type="text" id="id" value="">
    <input type="text" name="name" value="">
  </form>
  <script>
    document.forms[0].id.value='test1'
    document.forms[0].name.value='test2'
  </script>
</body>
```

태그 인식 전에 접근 불가

태그 인식 후에 접근 가능
id, name 속성으로 모두 접근 가능

location 객체

- 현재 URL에 대한 정보를 가지고 있는 객체
- window객체의 일부로서 window.location을 통하여 접근

속성	설명
href	전체 URL을 반환

메서드	설명
replace()	현재 문서를 새로운 문서로 대체한다.
reload()	현재 문서를 다시 로드한다.

history 객체

- 브라우저를 통해 방문한 사이트들에 대한 정보(URL)를 관리하는 객체

메소드	설명
back()	이전 page로 이동
forward()	다음 page로 이동
go(number)	지정된 단계의 page로 이동

이벤트

- HTML 이벤트

이벤트	설명
onchange	HTML 요소에 변화가 일어났을 때
onclick	HTML 요소에 사용자가 CLICK 할 때
onmouseover	HTML 요소에 마우스가 올려졌을 때
onmouseout	HTML 요소에서 마우스가 떠날때
onkeydown	키보드를 누를 때
onload	브라우저에 페이지가 로딩이 끝난후

```
<h1 onclick="change()">이것은 클릭 가능한 헤딩입니다.</h1>
```


기타 함수

1) 문자열을 숫자로 변환

- parseInt() / parseFloat()
- Number()

2) 숫자를 문자열로 변환

- String()

3) 문자열 함수

- split() : 문자열 분리
- indexOf() : 위치 찾기
- substring() : 문자열 잘라내기
- length : 문자열 길이
- charAt() : 한 문자 추출

<script>

```
var str="eng:math:kor";  
console.log(str.split(":"));  
console.log(str.indexOf(":"));  
console.log(str.substring(4,8));  
console.log(str.length);  
console.log(str.charAt(0));
```

</script>

정규표현식(Regular Expression)

- 역할

- ① 문자검색(search)
- ② 문자대체(replace)
- ③ 문자추출(extract)

- 생성

- 1) 생성자 함수 방식

```
const regexp1 = new RegExp("^abc");
```

- 2) 리터럴 방식

```
const regexp1 = /^abc/;
```

메소드

- 정규표현식에서 사용하는 메소드

메소드	문법	설명
exec	정규식.exec(문자열)	일치하는 하나의 정보(Array) 반환
test	정규식.test(문자열)	일치 여부(Boolean) 반환
match	문자열.match(정규식)	일치하는 문자열의 배열(Array) 반환
search	문자열.search(정규식)	일치하는 문자열의 인덱스(Number) 반환
replace	문자열.replace(정규식, 대체문자)	일치하는 문자열을 대체하고 대체된 문자열(String) 반환
split	문자열.split(문자열)	일치하는 문자열을 분할하여 배열(Array)로 반환
toString	생성자_정규식.toString()	생성자 함수 방식의 정규식을 리터럴 방식의 문자열(String)로 반환

플래그 문자

- 기호

g	모든 문자와 여러 줄 일치(global)
i	영어 대소문자를 가리지 않고 비교(ignore case)
m	여러 줄 검사(multi line)
U	유니코드(Unicode)
Y	lastIndex 속성으로 지정된 인덱스에서만 1회 일치(sticky)

```
const regexp1 = new RegExp("^abc","gi");
```

정규식 패턴(표현식)

정규식 패턴	설명
^	줄(Line)의 시작에서 일치, /^abc/
\$	줄(Line)의 끝에서 일치, /xyz\$/
.	임의의 한 문자와 일치
a b	a 또는 b와 일치, 인덱스가 작은 것을 우선 반환
*	0회 이상 연속으로 반복되는 문자와 가능한 많이 일치, {0,}와 동일
*?	0회 이상 연속으로 반복되는 문자와 가능한 적게 일치(lazy), {0}와 동일
+	1회 이상 연속으로 반복되는 문자에 가능한 많이 일치, {1,}와 동일

정규식 패턴(표현식)

정규식 패턴	설명
+?	1회 이상 연속으로 반복되는 문자에 가능한 적게 일치(lazy), {1}와 동일
?	없거나 1회 가능한 많이 일치
??	없거나 1회 가능한 적게 일치(lazy)
{3}	3(숫자)개 연속 일치
{3,}	3개 이상 연속 일치
{3,5}	3개 이상 5개 이하(3~5개) 연속 일치
{3,5}?	3개 이상 5개 이하(3~5개) 연속 중 가능한 적은 3개 연속 일치(lazy), {3}와 동일

정규식 패턴(표현식)

정규식 패턴	설명
()	캡처(Capture)할 그룹
(?<>)	캡처 그룹 이름 지정, /(?!<name>pattern)/ ES2018
₩1~9	정규식 내 캡처된 값 참조, /(abc)₩1/
(?:)	캡처(Capture)하지 않는 그룹
(?=)	앞쪽 일치(Lookahead), /ab(?!=c)/
(?!)	부정 앞쪽 일치(Negative Lookahead), /ab(?!c)/
(?<=)	뒤쪽 일치(Lookbehind), /(?!<=ab)c/ ES2018
(?!<=)	부정 뒤쪽 일치(Negative Lookbehind), /(?!<=ab)c/ ES2018

정규식 패턴(표현식)

정규식 패턴	설명
[abc]	a 또는 b 또는 c와 일치, 점(.)이나 별표(*) 같은 특수 문자는 []안에서 특수 문자가 아님, /₩.[.]/
[a-z]	a부터 z 사이의 문자 구간에 일치(영어 소문자)
[A-Z]	A부터 Z 사이의 문자 구간에 일치(영어 대문자)
[0-9]	0부터 9 사이의 문자 구간에 일치(숫자)
[가-힣]	가부터 힣 사이의 문자 구간에 일치(한글)
[2-7]	2부터 7 사이의 문자 구간에 일치(2,3,4,5,6,7)
[b-f]	b부터 f 사이의 문자 구간에 일치(b,c,d,e,f)
[다-바]	다부터 바 사이의 문자 구간에 일치(다,라,마,바)

정규식 패턴(표현식)

정규식 패턴	설명
[^abc]	a 또는 b 또는 c가 아닌 나머지 문자에 일치(부정)
\w	이스케이프 문자, /\w.\w?\w/\w\$\w^/
\wb	63개 문자(영문 대소문자 52개 + 숫자 10개 + _(underscore))가 아닌 나머지 문자에 일치하는 경계(boundary)
\WB	63개 문자에 일치하는 경계
\wd	숫자(Digit)에 일치
\WD	숫자가 아닌 문자에 일치
\wp{}	유니코드 속성(Property) 집합에 맞는 문자에 일치, /\wp{Emoji}/u ES2018
\WP{}	유니코드 속성 집합에 맞지 않는 문자에 일치, /\wp{Uppercase}/u ES2018

정규식 패턴(표현식)

정규식 패턴	설명
<code>\s</code>	공백(Space, Tab 등)에 일치
<code>\S</code>	공백이 아닌 문자에 일치
<code>\w</code>	63개 문자(Word, 영문 대소문자 52개 + 숫자 10개 + _)에 일치
<code>\W</code>	63개 문자가 아닌 나머지 문자에 일치
<code>\x</code>	16진수 문자에 일치, <code>/\x61/</code> 는 a에 일치
<code>\0</code>	8진수 문자에 일치, <code>/\0141/</code> 은 a에 일치
<code>\u</code>	유니코드(Unicode) 문자에 일치, <code>/\u0061/</code> 는 a에 일치
<code>\c</code>	제어(Control) 문자에 일치

정규식 패턴(표현식)

정규식 패턴	설명
\f	폼 피드(FF, U+000C) 문자에 일치
\n	줄 바꿈(LF, U+000A) 문자에 일치
\r	캐리지 리턴(CR, U+000D) 문자에 일치
\t	탭 (U+0009) 문자에 일치
\$`	문자 대체(replace) 시 일치한 문자 이전 값 참조
\$'	문자 대체(replace) 시 일치한 문자 이후 값 참조
\$+	문자 대체(replace) 시 마지막으로 캡처된 값 참조
\$&	문자 대체(replace) 시 일치한 문자 결과 전체 참조
\$_	문자 대체(replace) 시 입력(input)된 문자 전체 참조
\$1~9	문자 대체(replace) 시 캡처(Capture)된 값 참조

폼 유효성 검증

예시)

- 사용자가 필수적인 필드를 채웠는가?
- 사용자가 유효한 길이의 텍스트를 입력하였는가?
- 사용자가 유효한 이메일 주소를 입력하였는가?
- 사용자가 유효한 날짜를 입력하였는가?
- 사용자가 숫자 필드에 텍스트를 입력하지 않았는가?

[실습] 로그인 폼 유효성 검사하기

[실습] 회원가입 폼 유효성 검사하기