프로젝트 기반의 프론트엔드 프로그래밍

보수교육 - 2023.12.09 ~ 2023.12.10

강의 일정

1일차

교시	단원명	세부학습 내용
1	프론트엔드 시작하기	- 프론트엔드(HTML, CSS, JavaScript) 소개 - 개발환경(VSCode) 설치 - 코딩 프로젝트(PBL) 안내
2	HTML 기본문법	- 마크업(Markup)과 요소(Element) - 프로젝트에 필요한 기본요소 구성 - html, head, body 태그 등
3	HTML 주요요소	- img 태그, list 태그 - 리스트를 활용한 메뉴 생성하기

강의 일정

1일차

교시	단원명	세부학습 내용
4	CSS 설정하기	- 스타일시트(StyleSheet), 선택자(Selector) - 스타일 초기화 하기 - 외부 CSS문서의 경로와 관계 설정하기
5	CSS 스타일	- 문자(font), 문단(paragraph), 배경(background) - div 레이아웃 - Google font, Material Icons, 파비콘 설정 - div 레이아웃을 활용한 헤더, 본문, 푸터 설정
6	CSS 활용하기	- 배치(position), flex(정렬) - z-index(요소쌓임) - position을 활용한 요소 배치 - flex와 z-index를 활용한 순서 및 정렬

강의 일정

2일차

교시	단원명	세부학습 내용
7	CSS 활용하기	- 전환(transition), 변환(transform) - 전환과 변환을 활용한 애니메이션 및 요소와 이미지 슬라이드기능구성
8	JavaScript 개요	- 데이터의 종류(자료형) - 변수(let)와 상수(const) - 요소에 해당하는 데이터의 자료형 정의
9	JavaScript 활용하기	- 함수(Function) - 메소드 체이닝(Method Chaining) - 프로젝트에 사용되는 함수 정의 - 메소드 체이닝을 활용한 기능 연결

강의 일정

2일차

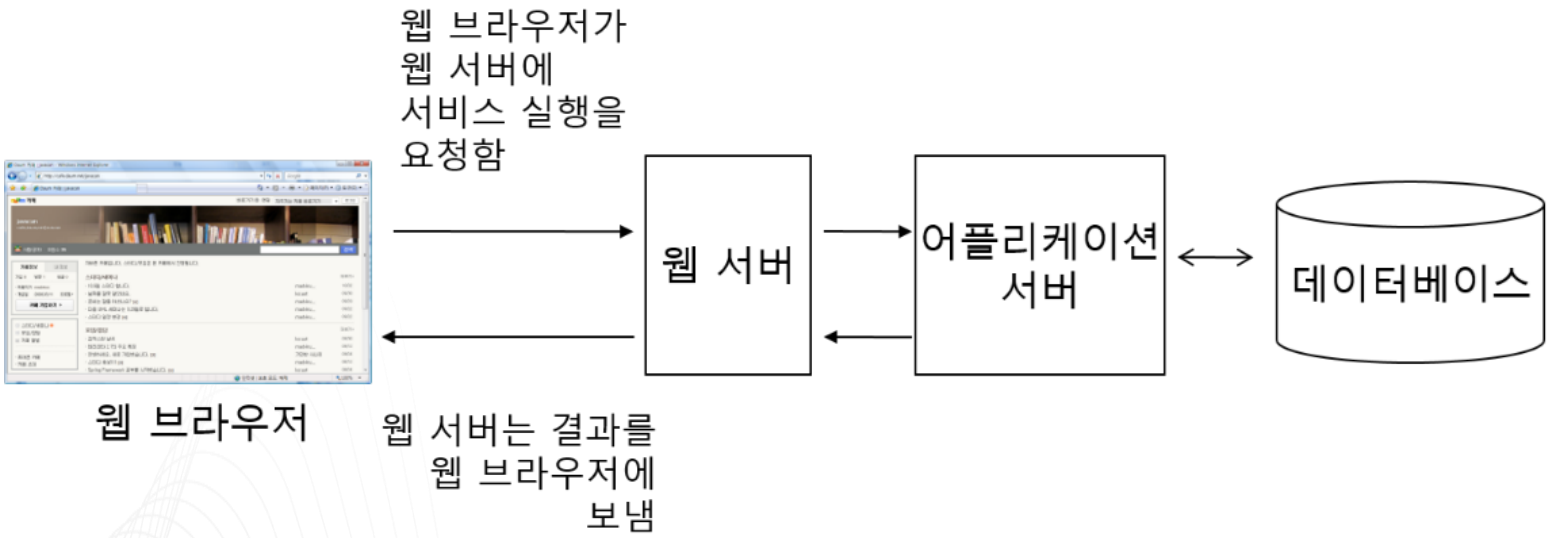
시간	단원명	세부학습 내용
10	JavaScript 주요기능	- DOM(Document Object Model) - 인수(Arguments), 이벤트(Event), 핸들러(Handler) - DOM을 활용한 애니메이션, ScrollTo 기능 구현
11	코딩 프로젝트	- 헤더와 풋터 작성하기 - 메인 화면 설정하기 - 목록 보이기
12		- 입력 페이지 작성하기 - 상세보기 화면 작성하기

1교시

프론트엔드 시작하기

개요

기본적인 웹 운영 형태



사용자는 웹 브라우저를 통해 서버에 요청(Request)을 전송하면 서버는 요청한 내용(자원, 데이터 등)을 처리하여 응답(Response)을 전송하는 구조.

웹 애플리케이션의 구성

웹 애플리케이션 = 프론트엔드 + 백엔드

- 프론트엔드(Front-end)
  - 웹 사용자가 브라우저 프로그램(MS 엣지, 구글 크롬, 사파리 등)을 통해 보는 화면을 개발하는 분야.

  - HTML
    - HTML은 하이퍼텍스트와 마크업 언어로 구성. 하이퍼텍스트는 페이지들 사이의 링크, 마크업 언어는 웹페이지의 구조를 정의. 웹 페이지의 뼈대.
  - CSS
    - CSS는 종속된 스타일시트(Cascading Style Sheets). 웹페이지에 다양한 스타일을 적용할 수 있게 해줌으로써 애플리케이션 페이지를 표시하는 프로세스를 단순하게 만들어주는 디자인 언어. 웹 페이지의 모습.
  - Javascript
    - 사용자들과 상호작용하는 기능을 제공. (최근 JS는 백엔드 개발에도 활용되고 있음)

웹 애플리케이션의 구성

웹 애플리케이션 = 프론트엔드 + 백엔드

- 백엔드(Back-end)
  - 서버 측 개발 분야로, 사용자가 요청한 기능(서비스 로직)을 내부적으로 수행하며, 데이터를 저장하고 관리하는 프로그램을 개발하는 분야.

  - Java
    - 가장 인기 있는 프로그래밍 언어 중 하나이자 객체지향 프로그래밍 언어(본 과정 주요 언어)
  - Javascript
    - 백엔드와 프론트엔드 모두에서 사용.(Node.js)
  - Python
    - 다양한 개발에 활용되며, 특히 딥러닝, 데이터 사이언스, 인공지능 분야에서 많이 사용

개발 환경 설정

개발 도구

- Visual Studio Code(VS Code)
  - https://code.visualstudio.com/

- 수업 관련 자료 공유(깃허브)
  - https://github.com/tiblo/font-end_edu

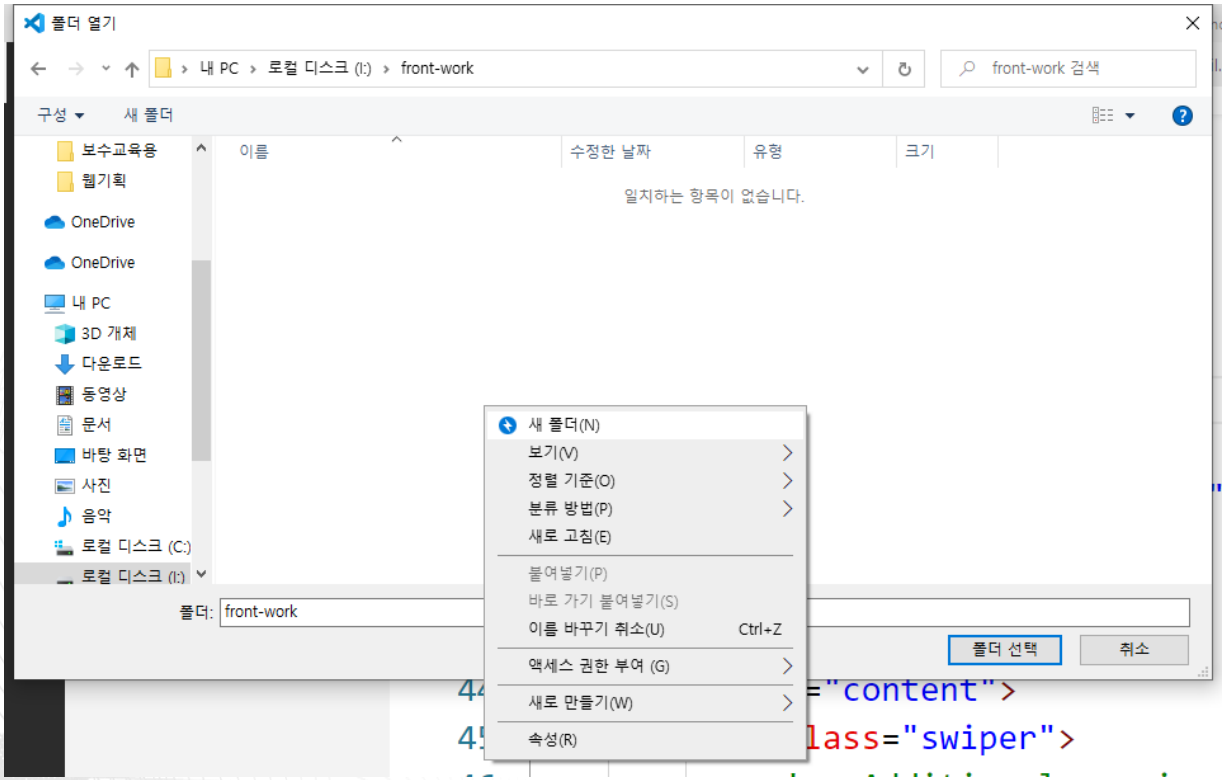
2-3교시

HTML

프론트엔드 프로젝트 생성

프로젝트 폴더 생성

- 파일 메뉴 > 폴더 열기...
  - '폴더 열기 창'에서 프로젝트용 폴더를 생성(start01) > '폴더 선택' 클릭



HTML

Hyper Text Mark-up Language

- 웹 문서의 골격을 만드는 부분

- HTML 구조
  - 한 HTML 문서는 다양한 요소(Elements)로 구성
  - 한 요소는 하위 요소를 포함(계층 구조)
    - 포함된 하위 요소를 자식요소(Child Element)
    - 포함하는 상위 요소를 부모요소(Parent Element)
  - 하나의 요소는 태그(Tag)와 내용(Contents)로 구성
    - 태그는 시작태그와 종료태그가 있음
    - 시작태그만 있는 것은 빈태그(Empty tag)라고 함
      - » 원래부터 시작 태그만 있는 것
      - » Self closing tag : 작성할 내용이 없을 경우 시작태그와 종료태그을 합친 형태
        예) ->

HTML

Hyper Text Mark-up Language

- 요소(Element)
  - 구조

    <시작태그 [**속성1="속성값" 속성2="속성값" ...**]>**내용**</종료태그>

  - 속성(attribute)
    - 요소에 대한 추가적인 정보를 지정할 때 사용
    - 한 시작태그에는 다양한 속성을 작성할 수 있음(속성이 없을 수도 있음)
    - 종료태그에는 속성을 넣지 않음

HTML

작성규칙

- 태그명은 대소문자를 구분하지 않지만, HTML5에서는 소문자를 권장
- 내용(Content)에 연속된 공백이나 줄바꿈은 화면(브라우저)에서 한 칸의 공백으로 출력
- 요소의 포함관계(계층 관계) 표현을 위해 들여쓰기 사용
- 한 요소는 시작태그와 종료태그로 작성하며, 다른 요소의 태그와 중첩되지 않아야 함
  - 예) **<h1>**제목** **어떤 내용… **</h1> ** X
    <h1>제목</h1> 어떤 내용… O
- 주석은 '<!--'로 시작하여 '-->'로 끝남

문서의 구조

```
<!DOCTYPE html>
<html>
    <head>
        ...
    </head>
    <body>
        ...
    </body>
</html>
```

html 문서 버전 -> html5임을 나타냄

문서의 시작과 끝을 나타내는 태그

문서의 정보를 나타내는 태그

문서의 내용을 나타내는 태그

문서의 구조

계층 구조

- 문서를 구성하는 요소는 계층 구조를 형성
- '상위 요소' - '하위 요소' 또는 **'부모 요소' - '자식 요소'**로 표현
- 요소 내부에 요소가 중첩되는 형상

```
<div>                          --> 부모(Parent) 요소
    <div>                      --> 자식(Child) 요소(후손 요소이기도 함)
        <div>내용</div>
    </div>                     --> 후손(Grand Child) 요소
</div>
```

- 들여쓰기를 사용하여 상위 요소(부모)와 하위 요소(자식)를 구분

초기 화면 작성 및 실행

index.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Document</title>
</head>
<body>


</body>
</html>
```

브라우저의 제목 표시줄에 출력

HTML 요소의 종류

블록(Block) 요소

- 브라우저 화면의 너비(가로 길이)를 모두 사용하는 요소
- 요소 사이에 줄 바꿈이 발생
- 대표 태그
  - 제목 태그들(h1, h2 등), <p>, <hr>, 등

인라인(Inline) 요소

- 내용(Contents) 만큼만 너비를 사용하는 요소
- 요소 사이에 줄 바꿈이 없음
- 대표 태그
  - <img>, <b>, 등

HTML Tags - head 영역

<title>

- HTML 문서의 제목을 정의하는 태그
- 브라우저 창(또는 탭)의 제목으로 출력

<link>

- 다른 파일을 가져와 현재의 문서와 연결할 때 사용
- 외부 스타일시트(CSS), 파비콘(favicon, favorite Icon) 등을 문서에 포함시킴
- 사용 속성
  - rel(Relationship) - 연결할 파일과 현재 문서의 관계(stylesheet, icon 등)
  - href(Hypertext Reference) - 연결할 파일의 위치

<style>

- 문서 내부에 스타일시트를 작성할 때 사용

HTML Tags - head 영역

<script>

- 외부 자바스크립트 파일을 현재 문서에 연결할 때 사용
- 현재 문서에 적용할 자바스크립트 코드를 작성할 때 사용
- <script> 태그는 <html> 내부의 어느 곳에서도 작성할 수 있음

<meta>

- 검색엔진이나 브라우저에 제공할 정보를 작성하는 태그
- 문서의 제작자, 문서의 개요, 검색 키워드, 문자셋 등
- 사용 속성
  - charset(Character Set) - 문자 인코딩 방식(euc-kr, utf-8 등)
  - name - 제공되는 정보의 종류(author, viewport 등)
  - content - 제공되는 정보의 내용

HTML Tags - body 영역

텍스트 관련 태그

- Heading Tag - <h1> ~ <h6>. 제목을 작성하는데 사용
  - <title>은 사이트의 이름. Heading tag는 페이지에서의 제목
  - 숫자가 클 수록 글자의 크기가 작아짐.
  - 블록 요소
- <p> - Paragraph. 문단을 작성하는데 사용
  - 문장을 작성하는 주요 태그
  - 블록 요소
-
 - Break. 줄바꿈 태그
  - 인라인 요소에 줄바꿈을 제공
  - 빈태그
- <hr> - Horizontal Rule. 페이지에 가로선을 작성하는데 사용
  - 블록요소이며 빈태그

HTML Tags - body 영역

목록(List) 태그

- 화면에 목록을 출력하는 요소
- 블록 요소
- 자식 요소로 <li> 태그를 사용하여 목록의 항목을 작성

- 비순서 목록 - Unordered List
  - 출력하는 항목의 객관적인 순서가 없는 형태
- 순서 목록 - Ordered List
  - 항목에 나름의 순서를 부여
- 목록 항목 - List Item
  - 목록 태그의 자식 태그로 항목을 출력하는데 사용

HTML Tags - body 영역

표 태그

- <table>
  - 표를 표현하는데 사용하는 태그
  - 행과 열 단위로 값을 출력하는 구조
- <tr> - Table Row
  - 표의 행을 표현하는 태그
- <th> - Table Header cell
  - 열의 제목을 표현하는 태그
  - 굵은 글씨체로 중앙에 정렬
- <td> - Table Data cell
  - 각 열의 값을 작성하는데 사용
  - 좌측 정렬

<table>

<tr>	**<th>**	**<th>**	**<th>**
<tr>	<td>	<td>	<td>
<tr>	<td>	<td>	<td>

HTML Tags - body 영역

이미지 태그

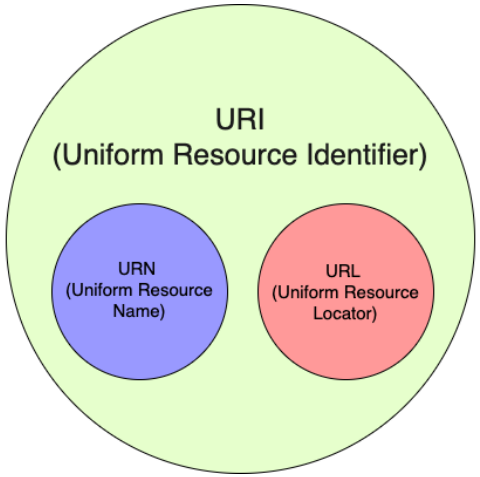
- <img>
  - 그림 파일을 문서에 추가하기 위한 태그
  - 빈태그, 인라인 태그
  - 사용 속성
    - src : 필수 속성. 삽입할 이미지 파일의 경로와 파일명을 작성.
    - alt : 이미지가 출력되지 않을 경우 대신 출력할 문장을 작성.
    - title : 이미지 위에 마우스가 오버되었을 경우 출력할 문장을 작성.
    - width : 이미지의 출력 너비 값을 지정.
    - height : 이미지의 출력 높이 값을 지정.
  - 너비(width)와 높이(height)를 지정하지 않으면 이미지의 원래크기로 출력
- 참고
  - Pixel(픽셀) - Pictures Element의 줄인말. 화소
  - 요소의 크기(너비/높이)를 지정하는 기본 단위(px)
    - 예) 화면 크기 : 1920X1080 - 가로 1920픽셀, 세로 1080픽셀

HTML Tags - body 영역

연결 태그

- <a> - Anchor(닻). Hyper link를 구현(페이지를 연결)
  - 클릭 이벤트를 통해 페이지를 전환(연결된 페이지로 이동)
  - 인라인 요소
  - 사용 속성
    - href - 이동할 문서의 URI를 지정하는 속성
    - target - 이동할 문서의 출력 대상을 지정하는 속성.
      - » _self(기본값), _blank(새 탭)

- 참고
  - URI(Uniform Resource Indentifier) : 통합자원 식별자
  - URL(Uniform Resource Locator) : 통합자원 위치정보
  - URN(Uniform Resource Name) : 통합자원 이름



HTML Tags - body 영역

공간 태그

-
  - 브라우저 전체 공간을 기준(너비)으로 일정 영역을 설정하는 태그
  - 특별한 형태나 모양을 표현하지 않고, 영역의 구분을 위해 사용
  - 문서 전체를 몇 가지 영역으로 분할하여 사용할 때 활용
  - 구분한 영역에 각기 다른 스타일을 부여
  - 블록 요소
-
  - 브라우저의 일부 공간(블록 요소의 일부)을 설정하는 태그
  - 특별한 형태나 모양을 표현하지 않고, 블록 내에서 영역의 구분을 위해 사용
  - 구분한 영역에 각기 다른 스타일을 부여
  - 인라인 요소

HTML Tags - body 영역

입력 양식 태그

- <input>
  - 입력란을 만드는 태그(사용자의 입력값을 받는 요소 작성)
  - 사용 속성
    - type : 입력란의 모양을 지정하는 속성
    - value : 사용자의 입력을 저장하는 속성
    - placeholder : 입력값을 힌트를 작성하는 속성
    - disabled : 입력란을 비활성화로 설정하는 속성
    - readonly : 입력란의 값을 수정할 수 없도록 설정하는 속성.(value 필요!)
  - 주요 type
    - text : 일반 텍스트 입력 형태
    - password : 비밀문자 입력 형태(입력한 값은 '●'로 변경)
    - button : 버튼 형태(value에 출력할 문구를 작성)
    - checkbox : 체크박스 형태(다중 선택). name 속성이 같은 <input>을 하나로 그룹화
    - radio : 라디오버튼 형태(단일 선택). name 속성이 같은 <input>을 하나로 그룹화

HTML Tags - body 영역

입력 양식 태그

- <textarea>
  - 여러 줄의 문장을 입력받는 요소용 태그
  - 블록 요소
  - 크기 지정 속성
    - rows : 높이 지정. 줄 수
    - cols : 너비 지정. 글자 수
- <select>
  - 선택 목록을 출력하는 태그
  - <option> 태그로 목록의 항목을 작성
- <label>
  - 꼬리표(이름표) 태그
  - 입력 태그의 클릭 영역을 확장하는데 사용하는 태그
  - 입력 태그의 id 속성과 <label> 태그의 for 속성을 사용하여 연결

HTML Tags - Attribute

전역 속성

- title
  - 요소에 대한 설명글(tool tip)을 지정하는 속성
  - 요소에 마우스를 올리면 출력됨
- style
  - 요소 단독의 스타일을 지정하는 속성
  - CSS 인라인 선언 방식
- class
  - 요소에 분류명(classification name)을 지정하는 속성
  - 요소에 스타일을 부여하거나 동적인 처리(자바스크립트)에 활용하는 값
- id
  - 요소에 식별자(Identifier)를 지정하는 속성
  - 고유(유일)한 값을 넣는 것이 좋음.

4-7교시

CSS

CSS

Cascading Style Sheets

- HTML 문서의 요소에 스타일을 적용하는 언어
- 요소가 어떻게 보여질 것인지를 기술
  - style 이란 웹 문서의 글꼴이나 색상, 정렬, 배치, 크기 등의 겉모습
- 형식

```
selector {
        property : value;
        property : value;
        ...
}
```

  - selector : 선택자. HTML 요소를 선택하는 지시자
  - property : 요소에 적용할 스타일 속성
  - value : 스타일 속성의 값

CSS 작성 위치

인라인 스타일

- HTML 태그의 style 속성에 작성
- 해당 요소에만 스타일 적용

내부 스타일

- <head> 요소 내부의 <style> 영역에 작성
- 한 문서 내부를 범위로 하여 selector로 선택할 수 있는 모든 요소에 동일한 스타일을 적용

외부 스타일

- 독립된 css 파일을 생성하여 작성 - '파일명.css'
- <link> 태그를 통해 연결하면 사이트 내의 모든 문서에 동일한 스타일을 적용

참고> 상위 요소에 적용한 스타일은 하위 요소에도 반영됨(상속)

CSS 적용 방식

외부 스타일 적용 방식

- <link> 방식
  - 예) html 파일이 있는 폴더 내에 css 폴더를 생성하여 style.css 파일을 작성
    <link rel='stylesheet' href='css/style.css'>

- import 방식
  - css 파일 또는 <style> 요소 내부에서 사용

```
<style>
@import url("css/style.css");
```

Selector(선택자)

기본 선택자

- 전체 선택자(Universal Selector) - *
  - 모든 요소를 선택하여 동일한 스타일을 적용
- 태그 선택자(Type Selector) - 태그명(h1, p, div 등)
  - 태그 이름으로 요소를 선택
  - 문서 내의 모든 같은 태그를 선택하여 동일한 스타일을 적용
- class 선택자(Class Selector)
  - 태그의 class 속성값으로 요소를 선택
  - 태그의 종류와 상관없이 같은 클래스값을 가진 모든 요소를 선택하여 스타일 적용
- id 선택자(Id Selector)
  - 태그의 id 속성값으로 요소를 선택
  - 태그의 종류와 상관없이 같은 id값을 가진 모든 요소를 선택하여 스타일 적용

Selector(선택자)

복합 선택자(Combinator)

- 일치 선택자(Basic Combinator)
  - 두 선택자를 빈칸없이 연속으로 작성
  - 태그와 class, 태그와 id, class와 id 등의 조합으로 사용
- 자식 선택자(Child Combinator) - selector1>selector2
  - selector1의 자식 요소 중 selector2에 해당하는 요소를 선택하여 스타일 적용
- 후손 선택자(Descendant Combinator) - selector1 selector2
  - 두 선택자 사이에 공백(' ')
  - selector1의 후손 요소 중 selector2에 해당하는 요소를 선택하여 스타일 적용
- 인접 형제 선택자(Adjacent Sibling Combinator) - selector1+selector2
  - selector1의 바로 다음 요소가 selector2인 경우 선택하여 스타일 적용
- 일반 형제 선택자(General Sibling Combinator) - selector1~selector2
  - selector1 다음의 모든 요소 중 selector2에 해당하는 요소를 선택하여 스타일 적용

Selector(선택자)

가상 클래스 선택자(Pseudo-Classes)

- :link
  - 링크 요소를 선택
- :visited
  - 연결된 페이지가 이미 방문한 페이지의 링크라면 선택
- :hover
  - 요소에 마우스 커서가 올라간 동안 선택
- :active
  - 요소를 클릭하는 동안 선택

Selector(선택자)

가상 클래스 선택자(Pseudo-Classes)

- :focus
  - 요소가 포커스를 가진 동안 선택
- :first-child
  - 형제 요소 중 첫 번째 요소라면 선택
- :last-child
  - 형제 요소 중 마지막 요소라면 선택
- :nth-child(n)
  - 형제 요소 중 n에 해당하는 요소를 선택
  - even(짝수), odd(홀수), 또는 간단한 수식으로 다수의 요소를 선택

Selector(선택자)

가상 요소 선택자(Pseudo-Elements)

- ::before
  - 요소의 내용(content) 앞에 가상의 요소를 생성하여 삽입
- ::after
  - 요소의 내용(content) 뒤에 가상의 요소를 생성하여 삽입

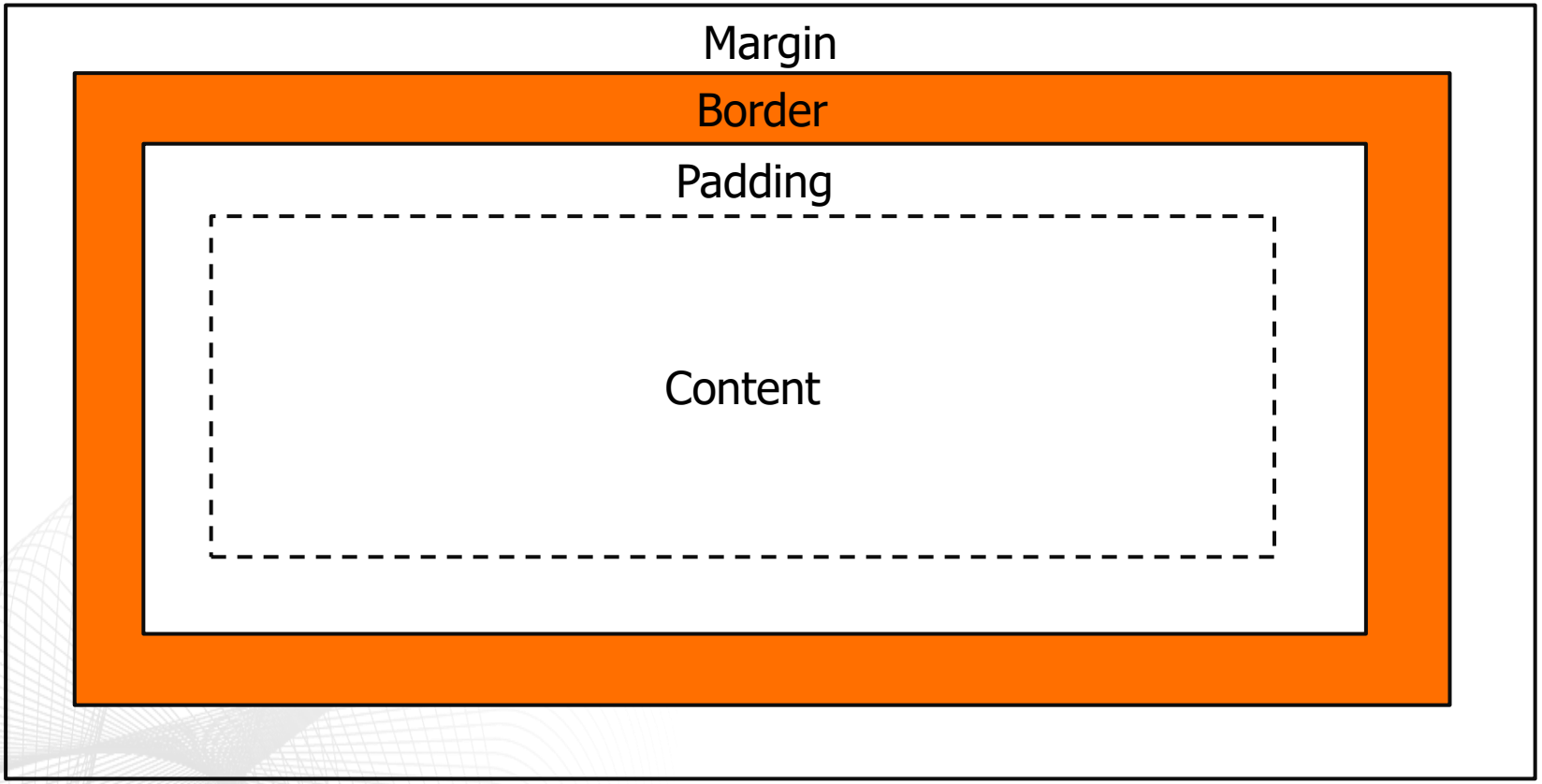
속성 선택자(Attribute)

- [attr]
  - 지정한 속성이 있는 요소를 선택
- [attr="value"]
  - 지정한 속성과 값이 일치하는 요소를 선택

CSS Properties

Box Model

- 모든 요소는 사각형 상자(Box) 형태로 문서에 배치됨



CSS Properties

width, height

- 요소의 너비와 높이를 지정

- 표현 단위
  - **px - 픽셀**
  - % - 상대적 백분율
  - **em – 부모 요소의 글꼴 크기**
  - **rem - 루트 요소(html)의 글꼴 크기**
  - vw - 뷰포트 가로 너비의 백분율
  - vh - 뷰포트 세로 너비의 백분율

CSS Properties

margin

- 요소의 외부 여백을 지정하는 속성

```
margin: top, right, bottom, left ;
margin: top, bottom  left, right ;
margin: top  left, right  bottom ;
margin: top  right  bottom  left ;
```

Padding

- 요소의 내부 여백을 지정하는 속성

```
padding: top, right, bottom, left ;
padding: top, bottom  left, right ;
padding: top  left, right  bottom ;
padding: top  right  bottom  left ;
```

CSS Properties

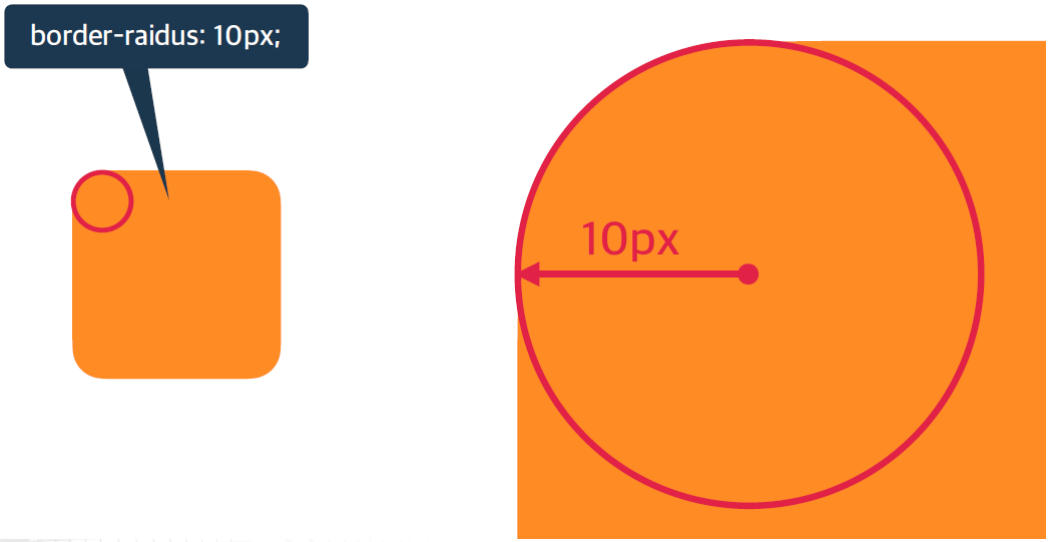
border

- 요소의 테두리 선을 지정하는 속성

border: 선-두께 선-종류 선-색상;

border-radius

- 요소의 모서리를 둥글게 설정하는 속성



CSS Properties

box-sizing

- 요소의 크기 계산 기준을 지정
- margin은 포함되지 않음
- 설정 값
  - content-box
    - 요소의 내용(content)으로 크기 계산
    - 실제 크기 = 내용크기 + padding + border
      width 200px, padding 10px, border 1px -> width = 222px
  - border-box
    - 크기에 padding과 border가 포함되어 계산
    - 실제 크기 = width 또는 height
      width 200px, padding 10px, border 1px -> width = 200px

CSS Properties

글자 관련 속성

- color - 글자의 색상을 지정하는 속성

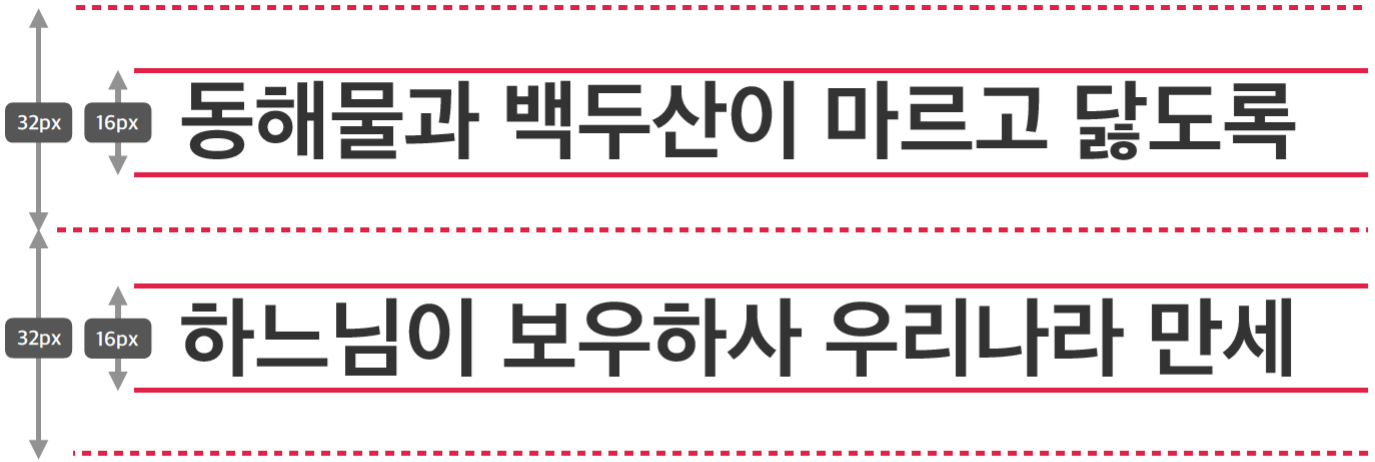
- 색상 표현
  - 색상 이름 - 브라우저에서 제공하는 색상(red, green, tomato 등)
  - 16진수 코드(HEX, Hexadecimal Colors) - #000000
  - rgb 함수 - 빨간색(Red), 초록색(Green), 파란색(Blue)로 표현. 각 0~255
    초록색 -> rgb(0, 255, 0)
  - rgba 함수 - 투명도(Alpha channel)을 포함
    반투명 초록색 -> rgba(0, 255, 0, 0.5)

- text-align - 문자의 정렬 방식
  - 요소의 내용 또는 하위 요소를 정렬
  - left, center, right

CSS Properties

글자 관련 속성

- text-decoration - 글자의 장식선을 지정하는 속성
  - none(선없음), underline(밑줄), overline(윗줄), line-through(중앙선)
- line-height - 한줄의 높이. 행간 간격




```
font-size: 16px;
line-height: 32px;
```

2배 차이

CSS Properties

글꼴 관련 속성

- font-style - 글씨체의 모양
  - 굵은 글씨(Bold), 이텔릭(Italic)
- font-weight - 글자의 두께
  - normal(기본), bold(두껍게)
- font-size - 글자의 크기
  - 기본 크기 - 16px
- font-family - 서체 지정 속성

  font-family: 글꼴1, "글꼭2", ... 글꼴계열;
  - 띄어쓰기 등 특수 문자가 들어간 경우 큰 따움표로 묶음
- 웹 폰트(구글 폰트 등) 사용
  - https://fonts.google.com/
  - <link>와 import 방식으로 문서에 추가 후 font-family 속성을 사용하여 지정

CSS Properties

배경 관련 속성

- background-color - 요소의 배경 색상
- background-image - 요소의 배경에 삽입
  - background-image: url("이미지파일");
- background-repeat - 배경 이미지 반복 여부 설정
  - repeat(가로, 세로 반복), repeat-x(가로 반복), repeat-y(세로 반복), no-repeat
- background-position - 배경 이미지 위치 설정
  - top, bottom, left, right, center
  - x, y 좌표 값 형태의 px 수치
- background-size - 배경 이미지 크기 설정
  - cover : 비율 유지. 요소의 너비에 이미지 너비를 맞춤
  - contain : 비율 유지. 요소의 내부에 모든 이미지가 나오도록 축소 또는 확대

CSS Properties

position

- 요소의 위치 지정 기준
- 주요 설정 값
  - static - 기준 없음(기본값). 원래 위치
  - relative - 요소의 static 위치를 기준으로 이동
  - absolute - 위치 상 부모 요소를 기준으로 이동
  - fixed - 브라우저에서의 위치를 기준으로 이동

  - 함께 사용하는 속성(static 제외)
    - top, bottom, left, right
  - z-index
    - 요소 쌓임 순서 지정
    - 숫자 값이 큰 요소는 위로, 작은 요소는 아래로 쌓임

CSS Properties

display

- 요소의 화면 출력 특성을 설정
- 주요 설정 값
  - block - 요소를 블록 요소로 변경
  - inline - 요소를 인라인 요소로 변경
  - inline-block - 요소가 인라인의 특성과 블록의 특성을 모두 갖도록 설정
  - flex - 요소를 플렉스 박스 컨테이너로 변경
  - none - 요소의 보여짐 특성을 제거(화면에서 사라짐)

CSS Properties

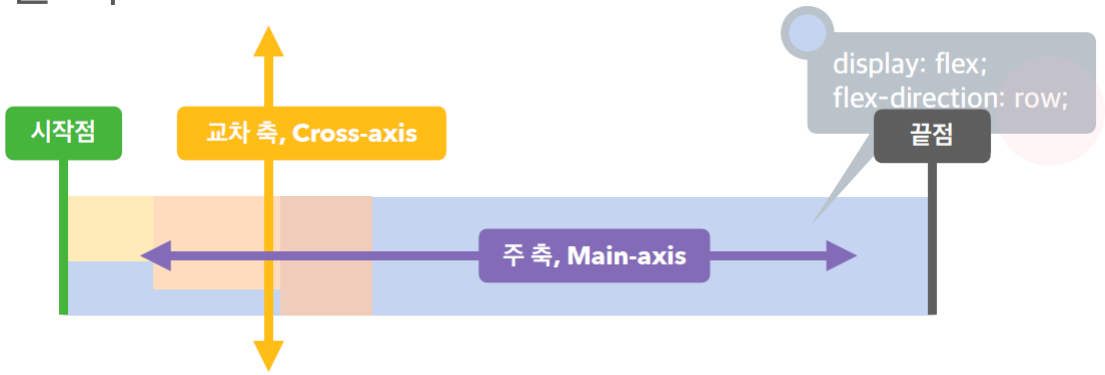
display: flex;

- 내부(자식) 요소를 배치하기 위해 사용하는 방식으로 요소의 크기가 불분명하거나 동적인 경우에도 각 요소를 정렬할 수 있는 효율적인 방법을 제공
- 해당 요소를 플렉스 박스(Flexible Box)의 컨테이너(Container)로 설정
- 자식 요소는 아이템(Item)이라고 함.
- 컨테이너 관련 속성
  - flex-direction : 아이템 정렬 방향 지정
    - row(좌->우), row-reverse(우->좌), column(위->아래), column-reverse(아래->위)
  - flex-wrap : 아이템 감싸임 여부(overflow 허용 여부)
    - nowrap(아이템이 컨테이너 밖으로 돌출됨), wrap(아이템이 컨테이너 내부에 존재)
    - wrap일 경우 컨테이너의 크기에 따라 여러줄로 아이템이 표현될 수 있음

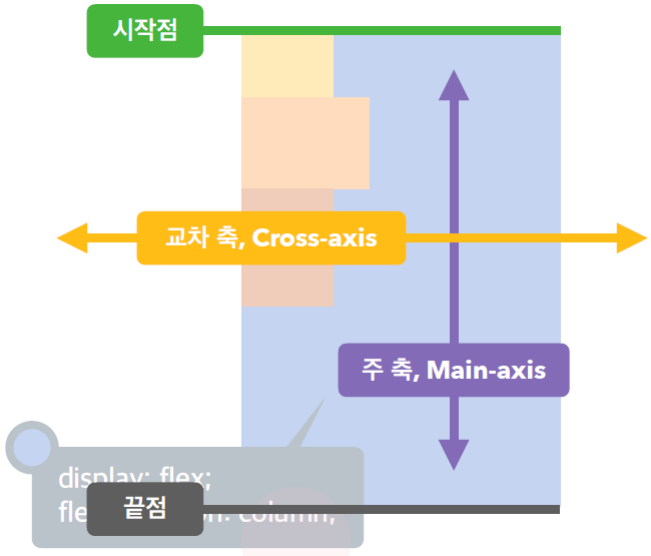
CSS Properties

display: flex;

- flex-direction에 따른 축
  - row인 경우



  - column인 경우



CSS Properties

display: flex;

- 컨테이너 관련 속성(cont.)
  - justify-content :주 축의 정렬 방법 설정
    - flex-start, flex-end, center, space-between, space-around
  - align-content : 교차 축의 여러 줄 정렬 방법 설정
    - stretch, flex-start, flex-end, center, space-between, space-around
  - align-items : 교차 축의 한줄 정렬 방법 설정
    - stretch, flex-start, flex-end, center, baseline

CSS Properties

display: flex;

- 아이템 관련 속성
  - order : 아이템 배치 순서 설정
    - 작은 수로 설정한 아이템을 먼저 배치
  - flex-grow : 아이템의 증가 비율 설정
    - 0(증가 비율 설정 안함), 숫자
  - flex-shrink : 아이템의 감소 비율 설정
    - 1(컨테이너의 크기에 따라 감소 비율 적용), 숫자
  - flex-basis : 아이템의 공간 배분 전 기본 크기
    - auto(요소의 내용 크기), 0(flex-grow에 따라), 숫자(px, em, rem 등)

CSS Properties

transition

- 요소의 전환(시작과 끝) 효과를 지정하는 속성
  - transition: 속성명 지속시간 타이밍함수 대기시간;
    - 속성명 : 배경색(이미지), 너비/높이, 글자색 등 대부분의 속성 사용 가능
    - 지속시간 : 전환 효과의 지속 시간(second)
    - 타이밍 함수 : 지속 시간 내에서 전환 효과의 타이밍 함수를 지정
      - » ease : 느리게 시작 - 빠르게 진행 - 느리게 멈춤
      - » linear : 일정한 속도
      - » ease-in : 느리게 시작 - 빠르게 멈춤
      - » ease-out : 빠르게 시작 - 느리게 멈춤
      - » ease-in-out : 느리게 시작 - 일정한 속도(linear) - 느리게 멈춤
    - 대기 시간 : 전환 효과 시작 전의 지연(대기) 시간 설정(second)

8-10교시

JAVASCRIPT

Javascript

Data Type

- String - 따옴표(', ")를 사용하여 표현
- Number - 정수와 실수
- Boolean - true와 false
- Undefined - 값이 할당되지 않은 상태
- Null - 변수를 의도적으로 비워놓음
- Object - 객체 데이터(key, value 쌍으로 구성되는 데이터 묶음)
- Array - 여러 데이터를 순차적으로 저장한 배열

변수(variable)

- 데이터를 저장하고 참조(사용)하는 공간
- var, let, const로 선언
  - 변수는 먼저 **선언**하고 **사용**해야 함
  - const(상수)는 데이터의 재할당이 불가능한 공간(초기화한 데이터 유지)

Javascript

연산자

- 비교 연산자
  - ==, != : 값만 비교
  - ===, !== :값과 타입을 모두 비교

제어문

- 조건문
  - if, else if
    - 특정 조건이 참인 경우 수행되는 코드 블록을 정의
  - switch
    - 입력값에 따라 처리를 다르게 하는 경우의 코드 블록을 정의
- 반복문
  - while
    - 조건이 성립하는 경우 계속 반복하는 코드 블록을 정의
  - for
    - 주어진 횟수에 맞도록 반복하는 코드 블록을 정의

Javascript

함수(function)

- 특정 동작(기능)을 수행하는 일부 코드의 집합
- 함수는 먼저 선언 후 사용해야 함
  - 함수 선언

    ```
    funtion 함수명(매개변수 목록){
      코드 집합;
    }
    ```

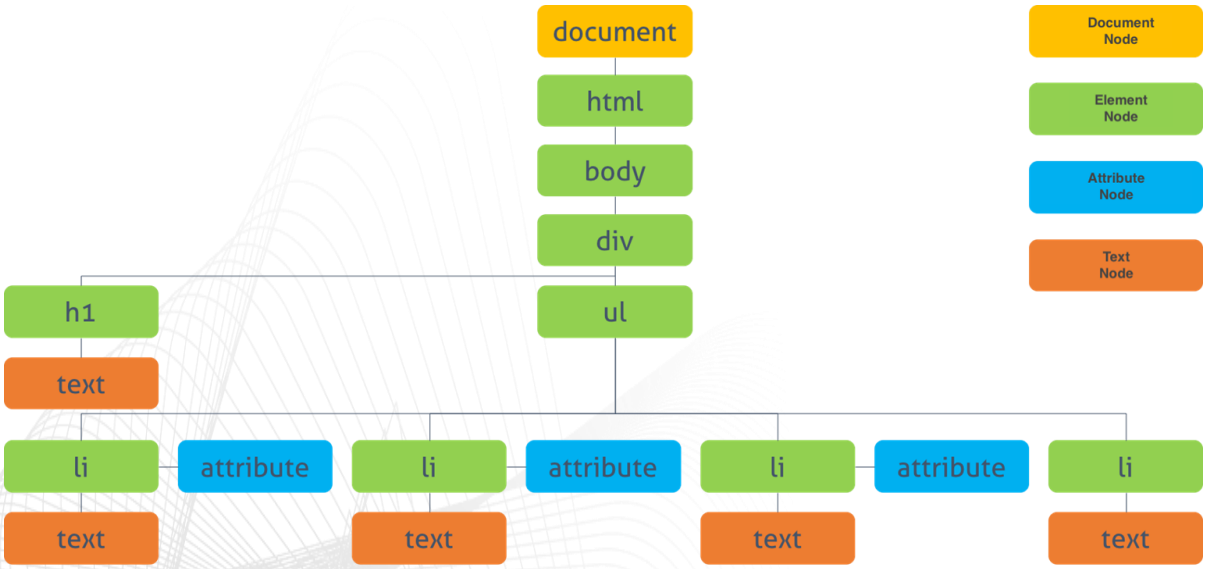
  - 함수 사용

    ```
    함수명(인자 목록);
    ```

Javascript

DOM(Document Object Model)

- 문서객체모델이라고 하며 HTML 혹은 XML문서의 구조화된 표현을 제공하는 표준
- 자바스크립트로 DOM 구조에 접근할 수 있는 방법이 제공되며 이를 통해 문서 구조, 스타일, 내용등을 변경



Document Node

DOM 트리에 접근하기 위한 최상위 노드로 모든 DOM 트리에 접근하기 위한 시작점

Element Node

HTML 구성 요소 즉 대표적으로 태그를 의미

Attribute Node

태그들의 속성을 객체화한 노드

Text Node

태그의 텍스트를 객체화한 노드

Javascript

DOM(Document Object Model)

- HTML 요소 핸들링
  - HTML DOM 요소에 접근하는 방법은 태그이름, 아이디, 클래스, 이름 등을 이용해 특정 노드 객체를 선택(css의 선택자와 유사)
  - HTML 요소선택
    - getElement[s]ByXXX() : 태그이름, class, id로 요소를 선택
    - queryselector[All]() : css의 선택자를 사용하여 요소를 선택하는 방식
- HTML 이벤트 핸들러 추가
  - 이벤트 속성에 함수를 할당하는 방식
    - document.getElementById("id_value").onclick = function() { code };
  - addEventHandler()를 활용하는 방식
    - document.getElementById("id_value").addEventHandler("click", function() { code });

Javascript

DOM(Document Object Model)

- 스타일 변경
  - 해당 요소의 style 속성변경, style 객체변경, 클래스 지정등의 방법으로 변경□
  - style 속성변경
    - document.getElementById('box1').setAttribute('style','background-color:yellow');
  - style 객체 변경(style.properties)
    - document.getElementById('box1').style.backgroundColor = 'yellow';
  - 클래스 지정 : 별도의 디자인 클래스를 지정해 놓고 해당 요소의 class 속성을 지정하는 형태
    - document.getElementById('box1').className = 'bgyellow';

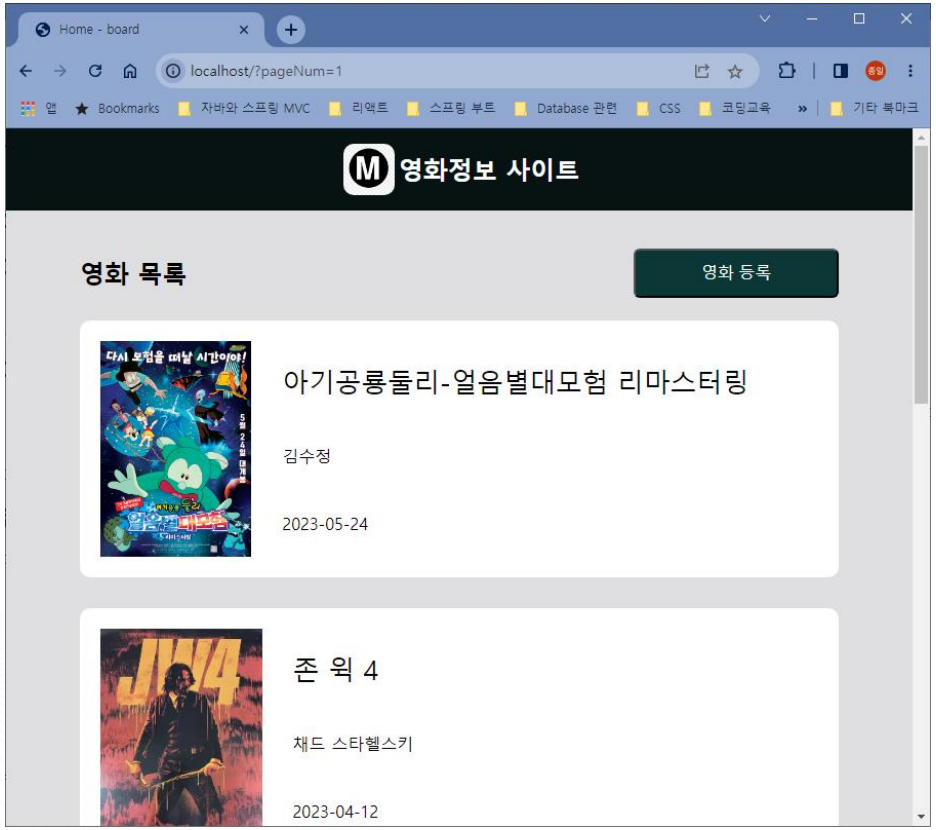
11 - 12교시

PROJECT

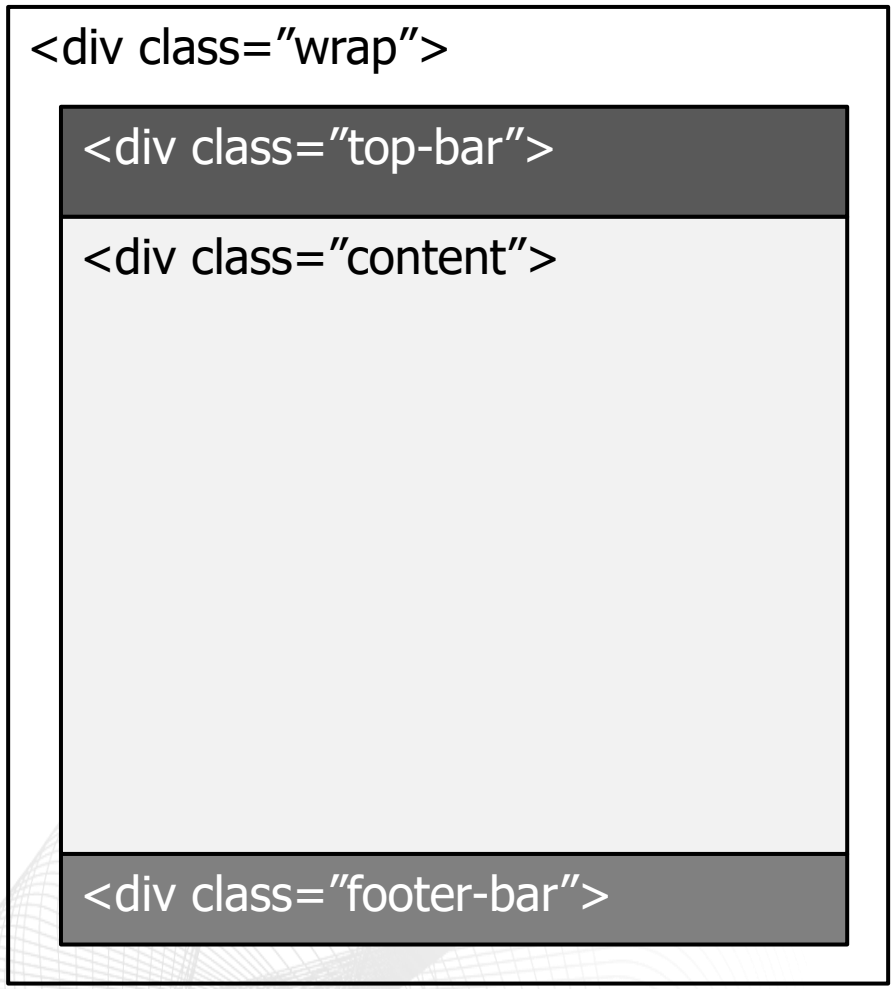
Project : 영화 정보 사이트

제공 페이지

- 영화 정보 등록
- 영화 정보 조회
- 영화 정보 상세보기



CSS 적용을 위한 HTML 구성



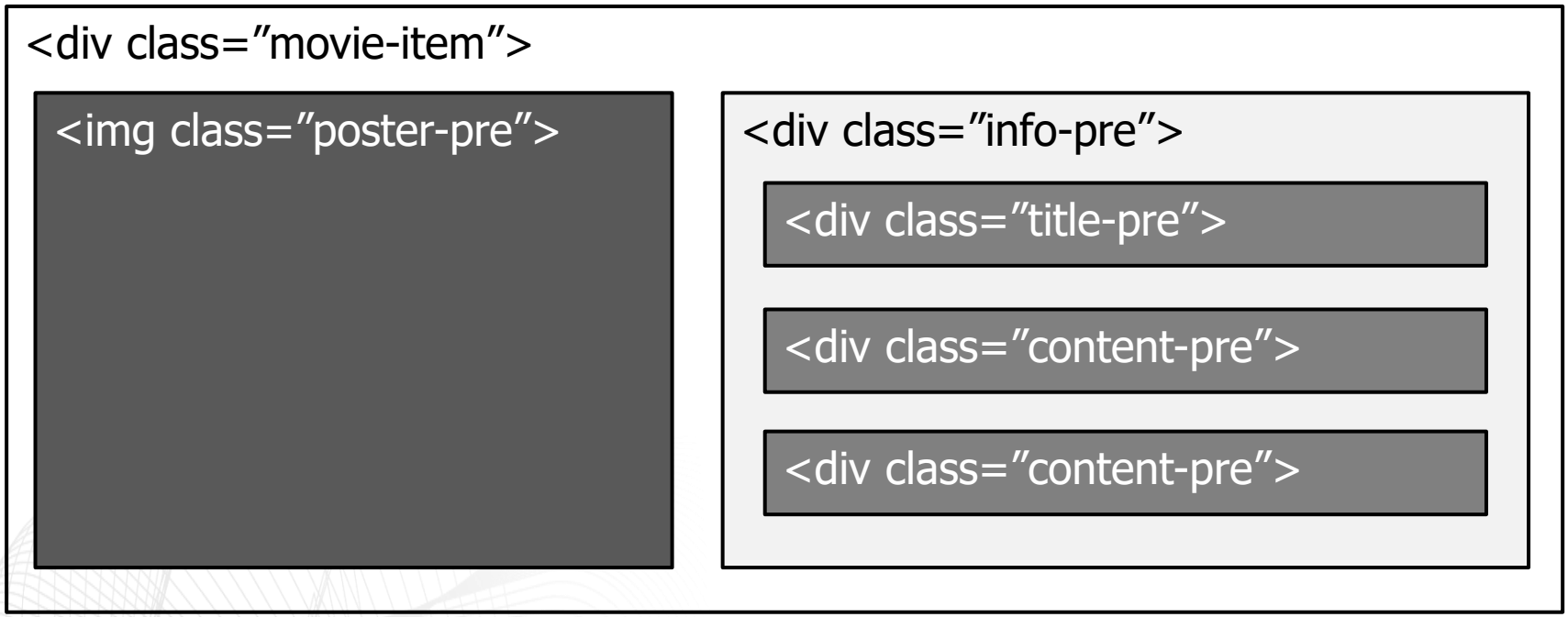
페이지 공통 구성

- 모든 페이지의 구성은 동일
- header.html과 footer.html은 같은 페이지를 insert 함.

CSS 적용을 위한 HTML 구성

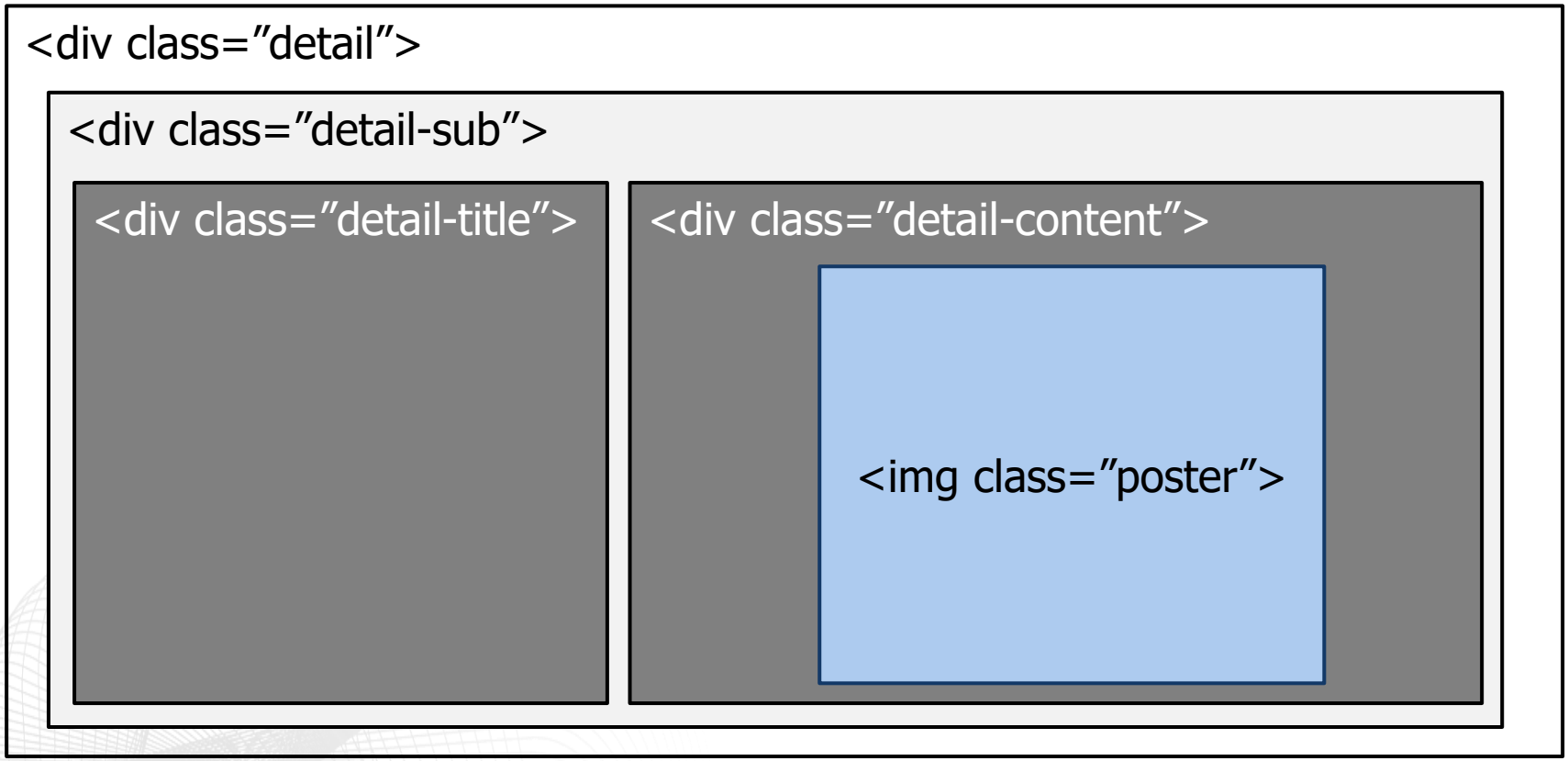
영화 목록

- 모든 영화 정보의 구성은 동일



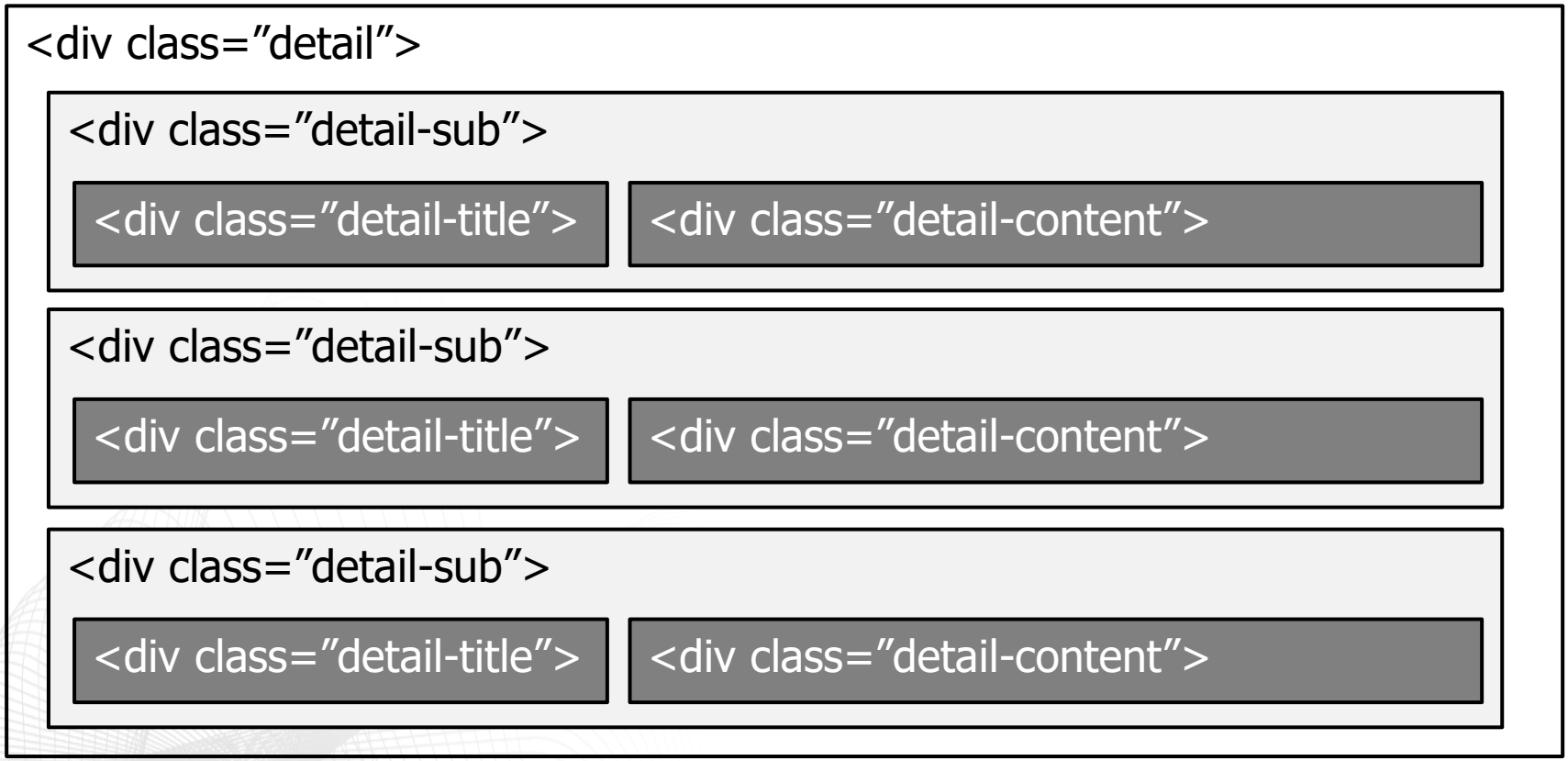
CSS 적용을 위한 HTML 구성

상세 보기 - 포스터



CSS 적용을 위한 HTML 구성

상세 보기



CSS 적용을 위한 HTML 구성

상세 보기 - 시놉시스