

# Web Dev

## ◆ 정적 웹 페이지와 동적 웹 페이지

동적 웹페이지: 사용자와 소통을 하면서 사용자의 요구에 맞게 화면을 바꿔야 하는 웹페이지이다. 데이터를 가공해 각 사용자에게 맞춤형으로 화면을 보여주기 때문에 데이터베이스와 웹 애플리케이션 서버가 필요하다. 클라이언트에서 요청을 보내면 웹 애플리케이션 서버에서 데이터베이스를 조회하거나 외부 서버에서 데이터를 읽어와 HTML을 렌더링합니다. 내용이 자주 바뀌는 콘텐츠에 유리하다.

정적 웹페이지: 사용자의 요구와는 상관없이 단방향으로 사용자에게 저장되어 있는 화면을 보여준다. 자주 바뀌지 않는 정적인 콘텐츠를 보여줄 때 좋다.

## ◆ DMS이란

### ▷ip와 domain

ip는 인터넷에 연결되어 있는 각 장치들의 주소 같은 것이다. 이 주소를 통해서 어떤 장치로 인터넷에 연결되었는지 알 수 있다.

도메인은 기억하기 어려운 ip를 위해 ip에 이름을 부여한 것이다.

도메인 -> ip주소

opentutorials.org -> 115.68.24.88

naver.com -> 220.95.233.172

daum.net -> 114.108.157.1

### ▷DMS

domain name system의 약자로 컴퓨터는 ip를 이용해 서로를 찾고 사람들은 domain을 이용해 원하는 컴퓨터를 찾기 때문에 DMS가 ip주소와 domain을 서로 바꿔주면서 연결을 도와준다.

### ▷URL

도메인이나 IP주소는 서버 컴퓨터의 위치를 나타내지만 해당 서비스를 제공하는 컴퓨터의 특정 정보 자원을 지칭하지는 않는다. URL은 원하는 정보자원을 찾기 위해서 해당 정보 자원의 위치와 종류가 기록되어 있는 위치 지정자이다.

## ◆github를 웹호스팅으로 이용하기

<https://opentutorials.org/course/2473/16117>

### ▷웹호스팅이 뭇데?

<https://brunch.co.kr/@webbible/41>

인터넷 홈페이지를 대신 운영해주는 서비스업이다. 대형 통신업체나 전문회사가 자신의 인터넷 서버를 고객에게 할당해주고, 고객이 직접 홈페이지를 운영하는 것과 같은 효과를 제공하는 서비스이다.

큰 서버를 가진 회사가 그 서버를 조금조금씩 나누어 주어 개인이나 작은 사업이 그 서버를 운영할 수 있게 해주는 것이다.

#### ▷서버는 뭐지?

서버도 그냥 컴퓨터이다. 우리는 보통 정보를 요청하는 입장으로만 컴퓨터를 이용한다. 그런데 내가 정보를 제공하고 싶으면 내 컴퓨터에 정보들을 올려놓고 다른 컴퓨터들이 이를 접속해 확인할 수 있도록 하면 된다. 이런 역할을 하는 것이 서버 컴퓨터이다. 그리고 이런 컴퓨터는 클라이언트 컴퓨터를 위해 24시간 켜져있고 사양도 좋다.

랙에는 서버 외에도 저장소나 스위치 등이 탑재되어 있고 랙에는 여러 서버가 들어가 있다.

#### ▷github와 웹

git으로 버전관리를 하는 나의 프로젝트는 github를 통해 다시 관리가 된다. github는 이 프로젝트를 웹사이트로 퍼블리싱하는 것을 도와주어 다른 사람들이 이 웹사이트에 여러 경로로 접근하는 것을 가능하게 한다.

---

# 생활코딩 WEB1 - HTML & INTERNET

## ◆HTML과 CSS

HTML은 정보를 표현하고 CSS는 웹을 꾸며준다.

## ◆코딩과 HTML

우리가 하는 일은 원하는 기능과 디자인을 구현하기 위해 코드/소스를 만들면 기계가 이 것을 토대로 구현을 해주는데 구현한 결과물을 애플리케이션/앱/응용프로그램/프로그램이라고 한다. 웹에서는 결과를 면 웹페이지라고 하고 웹페이지가 모이면 웹사이트, 기능이 많으면 웹애플리케이션이라고 한다.

HTML은 웹페이지를 만드는 코드이다.

## ◆ 태그

▷HTML은 태그가 아주 중요하다. 태그 사이에 간한 코드들은 그 태그의 속성을 가지고 있다.

- **<strong></strong>**: 강조
- **<underline></underline>**: 밑줄
- **<h1></h1> ~ <h6></h6>**: heading 즉 제목 속성이고 1~6까지 중요도에 따라 나뉜다.
- **<p></p>**: 단락 즉 하나의 문단을 나눈다고 명시하는 태그로 paragraph의 p이다.
- **<li></li>**: 목차를 만들어주는 태그로 list의 약자이다.

```
<li>HTML</li>
<li>CSS</li>
<li>JavaScript</li>
```

- **<ul></ul>**: ul태그는 두개 이상의 list가 있을 때 그 list들을 구별해주도록 <li></li>들을 묶는다. unordered list의 약자이다.

```
<ul>
  <li> HTML</li>
  <li> CSS</li>
  <li> JavaScript</li>
</ul>
<ul>
  <li>egoing</li>
  <li>k8805</li>
  <li>sorialgi</li>
</ul>
```

- <ol></ol>: ol태그는 ul과 동일하게 list들을 묶는데 ol은 ordered list으로 자동으로 앞에 숫자를 붙여준다.

```
<ol>
  <li>HTML</li>
  <li>CSS</li>
  <li>JavaScript</li>
</ol>
```

- <i></i>: italic 글자가 기울어진다.

▷이에 반해 하나의 열고 닫지 않고 혼자서 존재하는 태그도 있다. 이런 태그들은 감싸는 컨텐츠에 속성을 부여하는 것이 아니기 때문에 태그를 닫지 않는다.

- <br>: 줄바꿈으로 엔터를 치는 효과를 준다
- <img>: 본문에 이미지를 넣는 태그이다.
- <input>
- <hr>
- <meta>

▷attribute는 태그 뒤에 붙어 그 태그에 더 많은 속성을 더해주는 것이다.

```

```

▷작성되는 문장이 많아지고 페이지가 많아지면서 정보를 잘 정리하기 위한 태그가 존재한다.

- <!doctype html>: 웹페이지가 html으로 만들어졌다는 것을 표현하기 위해 문서의 시작에 <!doctype html> 태그를 추가한다.
- <html></html>: <html></html> 태그는 <head></head>태그와 <body></body> 태그를 감싼다.
- <head></head>: 본문안에 있는 data를 설명하는 정보는 <head></head>태그 안에 넣는다.
- <title></title>: 웹페이지 제목으로 설정되어 웹브라우저의 상단에 있는 탭에 제목이 표시된다.
- <meta>: meta 태그는 항상 <head></head> 태그 안에 적히며 이 안에는 페이지에 대한 metadata 정보를 가지고 있다. metadata는 data에 대한 data로 간단히 말해서 페이지 안에 적혀있는 data들에 대한 정보를 가지고 있다. metadata는 사용자가 읽지 않고

브라우저가 이를 사용하여 화면을 구성한다. **meta** 태그 안에는 보통 글자 설정, 페이지 설명, 작성자 정보, 키워드, 뷰포트 세팅이 들어간다.

```
<head>
  <title> WEB1 - html </title>
  <meta charset="UTF-8">
  <meta name="description" content="Free Web tutorials">
  <meta name="keywords" content="HTML, CSS, JavaScript">
  <meta name="author" content="John Doe">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
```

- **<body></body>**: 본문에 들어갈 data는 body태그 안에 넣는다.

```
<body>
<ol>
  <li>HTML</li>
  <li>CSS</li>
  <li>JavaScript</li>
</ol>
<h1>HTML</h1>
<p>Hypertext Markup Language (HTML)
<u>web</u> pages</strong> and web applications. Web browsers receive HTML
documents from a web server or from local storage and render them into multimedia web
pages.
</body>
```

#### ▶link 태그

▶**<a href="https://www.w3.org/TR/html5/"> </a>**

**<a></a>** 태그의 **a**는 닻을 의미하는 **anchor**의 줄임말이다. 이 태그는 다른 페이지를 하이퍼 링크로 연결할 수 있는 기능을 제공한다. **<a></a>** 태그 안에는 **href attribute**가 있는데 **hypertext reference**의 줄임말로 이 다음에 url을 적어주면 **<a></a>** 태그 안의 글을 클릭을 했을 때 적은 url로 연결된다.

**<a href="https://www.w3.org/TR/html5/" target="\_blank" title="html5 specification">Hypertext Markup Language (HTML)</a>**

**target = “ \_ blank” attribute**은 연결되는 url이 새 창에서 보이도록 한다. **title attribute**는 이 링크가 어떤 내용을 담고 있는지를 툴팁으로 보여주는 기능을 제공한다.

▶**<a href="#"> </a>**

```
<header>
  <a href="#" class = "logo">
    <!-- icon : you can find it in https://boxicons.com/-->
    <i class="ri-home-heart-fill"></i>
    <!-- writing -->
    <span>Logo</span>
  </a>
</header>
```

<a href="#" class="logo">: 이 부분은 <a> 태그로 로고를 나타냅니다. href="#"은 링크가 현재 페이지로 설정되어 있음을 의미합니다. class="logo"는 해당 요소에 "logo"라는 CSS 클래스를 부여합니다. 이를 통해 스타일링이 가능해집니다.

<i class="ri-home-heart-fill"></i>: 이 부분은 아이콘을 나타냅니다. <i> 태그 내에 class="ri-home-heart-fill"이 있으며, 이는 Boxicons 라이브러리에서 제공되는 "ri-home-heart-fill" 클래스를 적용하여 아이콘을 표시합니다. 이 아이콘은 집과 하트가 결합된 형태로 표현될 것입니다.

▷ 주석 태그

<!--와 --> 태그 사이에 있는 글자들은 주석 처리되고 무시된다.

## ◆ Internet & WEB

인터넷은 컴퓨터가 서로 연결되어 통신을 주고받는 컴퓨터끼리의 네트워크를 일컫는다. 웹은 인터넷이라는 네트워크 안에 있는 하나의 정보 네트워크로 웹 안에서 여러 정보가 클라이언트와 서버 사이에서 교환된다.

▷ Internet

인터넷(Internet)이란 여러 통신망을 하나로 연결한다는 의미의 '인터넷(inter-network)'라는 말에서 시작되었으며, 이제는 전 세계 컴퓨터들을 하나로 연결하는 거대한 컴퓨터 통신망을 의미한다. 인터넷은 클라이언트와 서버로 구성되며, TCP/IP라는 기본 프로토콜을 통해 제공되고 있다.

▷ World Wide Web

월드 와이드 웹(World Wide Web)이란 인터넷에 연결된 사용자들이 서로의 정보를 공유할 수 있는 공간을 의미한다. 간단히 줄여서 WWW나 W3라고도 부르며, 간단히 웹(Web)이라고 가장 많이 불린다.

웹은 인터넷 상에서 텍스트나 그림, 소리, 영상 등과 같은 멀티미디어 정보를 하이퍼텍스트 방식으로 연결하여 제공한다. 하이퍼텍스트(hypertext)란 문서 내부에 또 다른 문서로 연결되는 참조를 집어 넣음으로써 웹 상에 존재하는 여러 문서끼리 서로 참조할 수 있는 기술을 의미한다.

이때 문서 내부에서 또 다른 문서로 연결되는 참조를 하이퍼링크(hyperlink)라고 부른다. 웹에서는 HTML이라는 언어를 사용하여 누구나 자신만의 문서를 작성할 수 있다. 또한, 이렇게 작성된 웹상의 문서에 HTTP라는 프로토콜을 사용하면 누구나 검색하고 접근할 수 있게 된다.

웹에서는 HTML 언어를 사용하여 작성된 하이퍼텍스트 문서를 웹 페이지(web page)라고 부른다. 이러한 웹 페이지들 중에서 서로 관련된 내용으로 작성된 웹 페이지들의 집합을 웹 사이트(web

site)라고 한다. 웹은 이렇게 작성된 수많은 웹 페이지들이 하이퍼링크(hyperlink)를 통해 서로 연결되어 구성된다.

유저가 웹 페이지에 포함된 하이퍼링크를 따라 다른 웹 페이지로 계속 이동하는 것을 웹 서핑(web surfing)이라고 부르며, 이때 유저가 웹 페이지를 보기 위해 사용하는 프로그램을 웹 브라우저(web browser)라고 한다.

### ◆ 웹서버와 웹호스팅

웹서버를 운영하기 위해서는 컴퓨터가 있어야 하고 그 컴퓨터가 24시간 켜져 있어야 한다. 그래서 이런 일을 대행해서 웹서버를 전문적으로 빌려주는 비즈니스를 웹호스팅 업체라고 한다.

HTML은 웹브라우저가 해석하기 때문에 버쪽에서 특별히 해 줄 일이 없습니다. 이런 특성을 정적(static)이라고 합니다. 자연스럽게 HTML만으로 만들어진 웹사이트를 호스팅하는 서비스 중에는 무료가 많습니다. 동적(dynamic) 특성을 가진 페이지를 만들려면 php나 python, js, java등의 언어를 사용하고 이런 기술을 이용하려면 비용을 내야 한다.

---

## WEB2 - CSS

### ◆ CSS의 등장

화면에 들어갈 정보 외에 디자인에 대한 정보를 나타내기 위해 CSS이라는 언어가 등장한다. 원래는 더 많은 태그를 사용해 글자의 디자인까지 관리했지만 더 다양하고 체계적인 디자인을 위해 디자인만을 담당하는 CSS이라는 언어를 만들어 둘을 구분했다.

### ◆ CSS의 문법

▶ 어디서부터 어디까지가 HTML이고 어디까지가 CSS인지 구분하는 것이 중요하다.

- <style> </style> 태그 안에는 화면의 디자인을 이해하기 위해 웹브라우저가 알아야 할 정보가 들어간다. 그러면 자동으로 브라우저는 이 안의 정보를 CSS이라 생각하고 CSS문법으로 해석한다.

```
<style>
a {
    color:red;
    text-decoration:none;
}
</style>
```

- a { ... }: 선택자(selector)로 웹페이지에서 어떤 태그에 속성을 넣어줄지 선택하게 한다.
- color:red: 이것은 효과(declaration)으로 태그에 넣고 싶은 효과를 선택하게 한다.
- color: property
- red: value

- 두 번째 방법은 속성을 이용하는 것이다. HTML의 태그 안에 CSS를 사용하는 html속성을 넣어주면 자동으로 CSS효과를 넣어서 해석한다.  
`<a href="2.html:" style="color:red"> CSS </a>`

▷ 어떤 Property가 있을까

- `font-size: 60px` : 글자 크기를 바꿀 수 있는 속성과 값
- `text-align: center`: 글자의 좌, 우, 중간 정렬을 도와주는 속성과 값

▷ 어떤 Selector가 있을까

```

<style>
    #hhh {
        color:gray;
    }

    .abc {
        color:red;
        text-decoration:none;
    }

    a {
        color:black;
    }
</style>
...

```

```

<a href="2.html:" style="color:red" class = "abc" > CSS </a>
<a href="3.html:" style="color:red" class = "abc" id="hhh"> HTML </a>

```

- **class 선택자:** 이미 있는 태그 말고 태그 중에서도 내가 원하는 태그만 고르고 싶을 때 속성으로 `class="abc"`와 같이 임의로 `class` 이름을 골라주고 선택자를 `.abc { ... }`이라고 하면 태그 중에서도 `class` 값이 `abc`인 것들의 스타일을 따로 정할 수 있다.
- **id선택자:** `class` 선택자와 같은 역할을 하지만 `class` 선택자보다 우선 순위가 높다 왜냐하면 `id`는 한 태그에만 부여되는 고유한 속성이기 때문에 두 태그에 중복으로 부여될 수 없다. 원하는 태그의 속성에 `id="hhh"`와 같이 임의로 `id` 이름을 골라주고 선택자를 `#hhh { ... }`이라고 하면 태그 중에서도 `id` 값이 `hhh`인 것들의 스타일을 따로 정할 수 있다.
- **선택자 간의 우선순위:** `id` 선택자가 힘이 가장 쎄고 그 다음이 `class` 선택자 그리고 그 다음이 태그 선택자이다.
- 그리고 선택자를 더 구체적으로 만들 수도 있다. `#grid ol { ... }`이라고 하면 `grid`라는 `id`를 가진 태그 안에 `<ol></ol>` 태그가 있을 때 그 태그를 지칭하는 것이다.

## ◆ BOX 모델

▷ 각 태그는 성격과 쓰임에 따라서 자신의 크기만큼을 쓰기도 하고 화면 전체를 쓰기도 한다. 화면 전체를 쓰는 태그를 **block level element**이라고 하고 자신의 크기만큼 쓰는 태그를 **inline element**이라고 부른다.

- border 속성을 통해 차지하는 공간의 경계에 대한 디자인을 바꿀 수 있다.

```
h1{  
    border-width:5px;  
    border-color:red;  
    border-style:solid;  
}
```

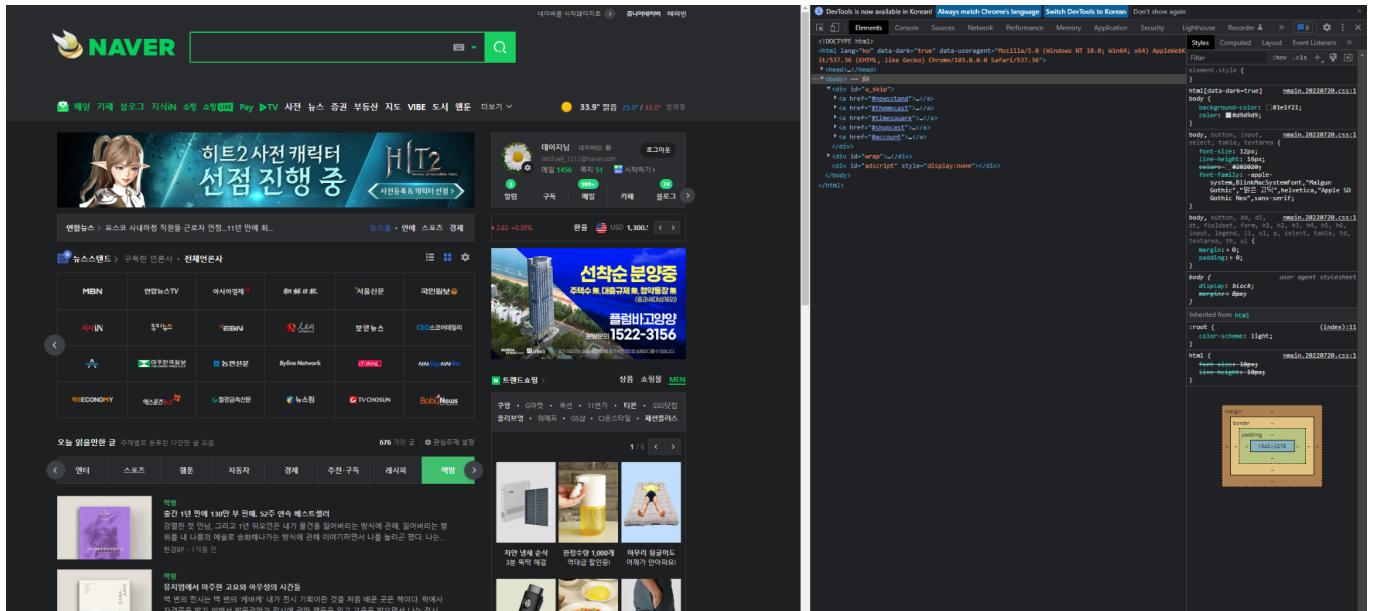
- display 속성을 통해 태그가 화면 전체를 쓸지 아니면 자기 자신만큼만 쓸지도 결정할 수 있다.

```
a{  
    display:inline;  
    /* display:none;  
    /* display:block;  
}
```

▷ Box 모델은 각 태그가 차지하는 공간의 크기를 결정할 수가 있다 내용물과 박스 사이공간은 padding값으로 조작할 수 있고, 박스와 그 주위 여백은 margin, 박스 그 자체의 크기는 width와 height으로 조작할 수 있다.

```
h1{  
    border: 5px solid red;  
    padding: 20px;  
    margin: 20px;  
    display: block;  
    width: 100px;  
}
```

▷ 웹페이지 검사 기능을 이용하면 웹페이지에서 어떤 식으로 html과 css가 쓰였는지 알 수 있다. 단축키는 f12이다.



## ◆ GRID

▶ 태그로 감싸야 하는데 아무 의미 없는 태그가 필요할 때 `<div></div>` 또는 `<span></span>` 태그를 쓸 수 있다. 그리고 그 태그로 원하는 영역을 감싸고 `<style></style>`로 div태그의 레이아웃 모양을 결정할 수 있다.

```

<head>
  <style>
    #grid{
      border:5px solid pink;
      display:grid;
      grid-template-columns: 150px 1fr;
    }
    div{
      border:5px solid gray;
    }
  </style>
</head>
<body>
  <div id="grid">
    <div>NAVIGATION</div>
    <div>Lorem ipsum dolor
  
```

▶ can i use라는 사이트를 이용하면 html과 css의 기술들을 어떤 웹브라우저에서 사용할 수 있는지 확인할 수 있다.

## ◆ 반응형 디자인

반응형 디자인은 화면의 크기에 따라서 웹페이지의 각 요소들이 반응해서 최적화된 디자인을 보여주는 디자인이다. 이를 가능하게 하는 것이 media query이라는 개념이다.

▷ if 조건문 처럼 <style></style> 태그 안에 @media()를 넣어주고 ()안에 원하는 조건을 넣은 다음에 그 밑에 조건이 참일 때 일어나는 효과를 적어주면 된다.

```
@media(min-width:800px)
div{
  display:none;
}
```

## ◆ CSS 코드의 재사용

웹페이지가 여러 개이고 그 웹페이지들에 중복적인 모양의 코드가 있을 때 디자인 코드만 따로 파일을 만들면 유지보수에 용의하다.

▷ <style></style> 태그 안에 있는 코드들은 CSS코드로 이루어지고 이 부분을 참고해서 HTML이 웹의 디자인을 완성한다. 그래서 이 부분을 따로 .css파일안에 저장하고 <style></style>태그 부분은 이 파일을 참고하라고 할 수 있다. 이 때 <link> 태그를 이용하면 이 파일에 어떤 파일과 연결되었는지 알려줄 수 있다.

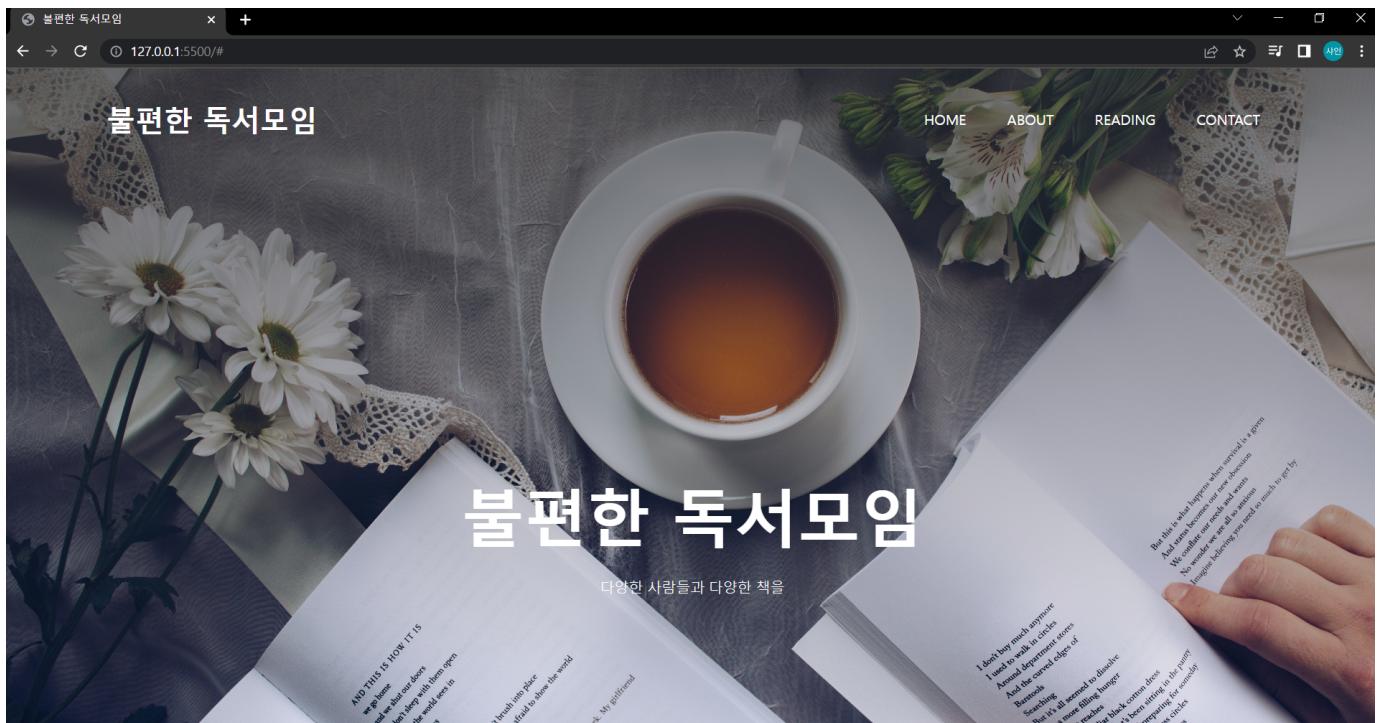
```
//xxx.html
<!doctype html>
<html>
<head>
  <title>WEB - CSS</title>
  <meta charset="utf-8">
  <link rel="stylesheet" href="style.css">
</head>
<body>
...
</body>
```

```
//style.css
body{
  margin:0;
}
a {
  color:black;
}
h1 {
  font-size:45px;
  text-align: center;
  border-bottom:1px solid gray;
  margin:0;
  padding:20px;
}
```

---

8월 2일

## ◆HOME WITH NAVIGATION BAR



▷ .banner `<div>` 안에 .navbar `<div>`와 .content `<div>`를 만든다. .navbar 안에 .logo `<h1>`과 list를 만든다.

```
<body>
  <div class="banner">
    <div class="navbar">
      <h1 class="logo">
        불편한 독서모임
      </h1>
      <ul>
        <li><a href="#">Home</a></li>
        <li><a href="#">About</a></li>
        <li><a href="#">Reading</a></li>
        <li><a href="#">Contact</a></li>
      </ul>
    </div>
    <div class="content">
      <h1>불편한 독서모임</h1>
      <p>다양한 사람들과 다양한 책을
      </p>
    </div>
  </div>
```

```
</div>
```

```
</body>
```

▷ .banner 안에는 전체 화면에 들어갈 이미지 정보가 들어가야 한다. 크게 .banner의 전체 크기를 설정하고 background 이미지 설정을 하면 된다.

```
.banner{  
    width: 100%;  
    height: 100vh;  
    background-image: linear-gradient(rgba(46, 46, 46, 0.623), rgba(0, 0, 0, 0)), url(backgroung.jpg);  
    background-size: cover;  
    background-position: center;  
}
```

▷ .banner 안에는 .navbar와 .content가 들어간다. .navbar의 경우 크기 정보와 margin과 padding을 설정하고 display 를 flex으로 설정하고 가로 세로 정렬을 한다.

```
.navbar{  
    width: 85%;  
    margin: auto;  
    padding: 35px 0;  
    display: flex;  
    align-items: center;  
    justify-content: space-between;  
}
```

▷ .navbar 안에 있는 ul li를 골라서 css에서 style 설정을 한다.

```
.navbar ul li{  
    list-style: none;  
    display: inline-block;  
    margin: 0 20px;  
    position: relative;  
}  
.navbar ul li a{  
    text-decoration: none;  
    color: #fff;  
    text-transform: uppercase;  
}
```

## ◆background 이미지 넣기

background에 이미지를 넣고 싶으면 일단 background 전체를 담당하는 .banner class를 만들어 주고 그 안에 width와 height으로 크기를 정한 다음에 background property로 설정을 해주면 된다.

▷ width and height는 그 요소의 높이와 가로길이를 설정하는 속성으로 padding, border, margin을 포함하지 않는 순수 크기이다.

- auto - This is default. The browser calculates the height and width
- px - Defines the height/width in px, cm etc.
- % - Defines the height/width in percent of the containing block
- initial - Sets the height/width to its default value
- inherit - The height/width will be inherited from its parent value

▷background를 설정하기 위해서는 여러가지 property가 있다.

- background-image: value값으로 여러 url을 넣으면 중복해서 화면을 구성한다.
- background-size: auto | px px | % % | cover
- background-position: x% y% | center | top | bottom | left | right

## ◆Navigation bar

navigation bar는 크게 로고와 navigation item으로 나뉜다. 일단 .navbar class를 만들고 그 클래스의 크기와 margin, padding을 정한 다음에 display를 정해야 한다.

▷padding

▷margin

▷display는 각 element가 차지하는 공간에 대한 정보를 담는다. 기본적으로는 block에 담길 것인지 inline에 담길 것인지 정해주지만 그것 외에도 많은 value가 있다.

- inline Displays an element as an inline element (like <span>). Any height and width properties will have no effect
- block Displays an element as a block element (like <p>). It starts on a new line, and takes up the whole width
- contents Makes the container disappear, making the child elements children of the element the next level up in the DOM
- flex Displays an element as a block-level flex container
  1. flex 컨테이너는 기본적으로 수평 방향으로 자식 요소들을 배치합니다. flex-direction 속성을 사용하여 수직 방향 배치로 변경할 수 있습니다.
  2. flex 컨테이너는 주로 1차원(수평 또는 수직)의 레이아웃을 만드는데 사용됩니다. 자식 요소들은 주축(main axis)을 따라 배치되며, 교차축(cross axis)에 대한 정렬 옵션도 제공됩니다.
  3. flex: flex 컨테이너는 자식 요소들의 크기를 유연하게 조정할 수 있습니다. flex-grow, flex-shrink, flex-basis 등의 속성을 사용하여 자식 요소들의 크기 비율을 설정할 수 있습니다.
  4. flex 컨테이너는 justify-content와 align-items 등의 속성을 사용하여 자식 요소들의 배치를 제어할 수 있습니다. 주축(main axis)과 교차축(cross axis)에서의 정렬 및 공간 분배가 가능합니다.
- grid Displays an element as a block-level grid container
  1. grid: grid 컨테이너는 행(row)과 열(column) 기반의 그리드를 생성하며, 자식 요소들을 이 그리드에 배치할 수 있습니다. 수평 방향이든 수직 방향이든 다양한 배치 방식을 설정할 수 있습니다.
  2. grid: grid 컨테이너는 2차원의 그리드 레이아웃을 만드는데 사용됩니다. 행(row)과 열(column)을 설정하여 자식 요소들을 그리드 셀에 배치할 수 있으며, 각 셀의 크기와 위치를 정밀하게 제어할 수 있습니다.
  3. grid: grid 컨테이너는 자식 요소들의 크기를 그리드 셀에 맞게 조정하거나, fr 단위를 사용하여 비율을 설정할 수 있습니다. 또한, grid-template-rows 및 grid-template-columns 속성을 사용하여 행과 열의 크기를 세밀하게 제어할 수 있습니다.

- grid: grid 컨테이너는 grid-template-areas, grid-template-rows, grid-template-columns 등의 속성을 사용하여 자식 요소들을 그리드 영역에 배치할 수 있으며, justify-items,

navbar은 기본적으로 flex box를 display으로 사용한다. display: flex;를 이용하면 된다. 그 외 많은 property를 이용해서 안의 item들을 정렬할 수 있다. align-items을 이용해 세로 정렬을 하고 justify-content를 이용해서 가로 정렬을 한다.

- align-items: stretch | center | flex-start | flex-end | baseline| : 아이템의 세로 정렬 Vertically aligns the flex items when the items do not use all available space on the cross-axis
- justify-content:center | flex-start | flex-end | baseline : 아이템의 가로 정렬 Horizontally aligns the flex items when the items do not use all available space on the main-axis

마지막으로 .navbar안에 있는 list 안의 element에 대해 display와 list-style을 정해준다.

```
.navbar ul li{  
    list-style: none;  
    display: inline-block;  
    margin: 0 20px;  
    position: relative;  
}
```

<https://developer.mozilla.org/en-US/docs/Web/CSS/list-style>

[https://www.w3schools.com/css/css\\_inline-block.asp](https://www.w3schools.com/css/css_inline-block.asp)

- list-style은 list 앞에 어떤 식으로 기호를 넣을지 정하는 것으로 둥그라미, 네모, 아이콘,,, 등등 설정할 수 있다.
- display: inline-block: element의 가로세로 크기를 정할 수 있고, 위 아래 margin/padding을 정할 수 있다. 그리고 각 element 사이에 구분선이 없다.

## ◆ Navbar의 요소 밑에 움직이는 줄 디자인 추가하기

```
.navbar ul li::after{  
    content: '';  
    height: 3px;  
    width: 0;  
    background:#009688;  
    position: absolute;  
    left: 0;  
    bottom: -10px;  
    transition: 0.5s;  
}  
.navbar ul li:hover::after{  
    width: 100%;  
}
```

# 8월 3일

## ◆ :root selector

```
:root {  
    --black: #333;  
    --white: #fff;  
}
```

<https://www.digitalocean.com/community/tutorials/css-root-pseudo-class>

:root selector는 가장 상위 parent를 가리킨다. HTML 프로젝트에 쓸 경우 html tag가 가장 부모이기 때문에 html selector와 동일한 역할을 하지만 우선순위가 html selector보다 높다.

활용 방법으로는 html selector에서 설정하고 싶은 property 모두를 사용할 수 있고, 그 외에도 전역적으로 CSS custom property를 만들어서 사용할 수 있다.

```
:root {  
    margin: 0;  
    padding: 0;  
    color: #0000FF;  
    font-family: "Helvetica", "Arial", sans-serif;  
    line-height: 1.5;  
    --primary-color: #0000FF;  
    --body-fonts: "Helvetica", "Arial", sans-serif;  
    --line-height: 1.5;  
}  
  
p {  
    color: var(--primary-color);  
    font-family: var(--body-fonts);  
    line-height: var(--line-height);  
}
```

### ▷CSS custom property

css에서는 변수를 만들어 값을 저장하고 나중에 불러올 수가 있다. 보통은 .root selector안에 전역변수 형태로 저장해서 문서 전체에서 사용한다. 정의할 때는 --xxx으로 정의하고 사용할 때는 var()함수로 이용한다.

```
:root {  
    --funky-shadow: 3px 3px 10px brown;  
}  
  
.ugly-box {  
    box-shadow: var(--funky-shadow);  
}
```

## ◆class 내에 있는 class를 selector로 고를 때 주의

.class1.class2 selector는 html문서에서 tag에 class를 표시할 때 한 tag 안에 적는 attribute안에 두개의 class가 적혀있는 tag를 가리킨다.

```
<section class="banner" class="cover">
```

```
.banner.home
```

반면에 중간에 스페이스바가 있는 selector, .class1 .class2 의 경우 .class1으로 표시된 tag 안에 있는 자식 tag 중에 class2를 가리킨다.

```
<section class="banner" id="home">
  
  <div class = "contentBx">
    <h2>불편한 독서모임</h2>
    <h4>다양한 사람들과 다양한 책을</h4>
    <a href="#" class="btn">About us</a>
  </div>
</section>
```

```
.banner .home
```

## ◆position

[http://www.tcp school.com/css/css\\_position\\_position](http://www.tcp school.com/css/css_position_position)

position 속성은 HTML 요소가 위치를 결정하는 방식을 설정합니다.

CSS에서 요소의 위치를 결정하는 방식에는 다음과 같이 4가지 방식이 있습니다.

1. 정적 위치(static position) 지정 방식은 HTML이 단순히 웹 페이지의 흐름에 따라 차례대로 요소를 위치시키는 방식으로 default값이다. 단순히 쓸는 방법으로 element를 추가하기 때문에 top, right, bottom, left 속성값에 영향을 받지 않습니다.

```
position: static;
```

2. 상대 위치(relative position) 지정 방식: static을 기준으로 위치를 설정하는 방법으로 top, right, bottom, left 속성값으로 위치를 조정할 수 있다.

```
position: relative;
top: 0;
left: 30px;
```

3. 고정 위치(fixed position) 지정 방식은 뷰포트(viewport)를 기준으로 위치를 설정하는 방식으로 즉, 웹 페이지가 스크롤 되어도 고정 위치로 지정된 요소는 항상 같은 곳에 위치하게 됩니다. 따라서 navigation bar같은 곳이나 banner에 사용되기 좋다.

```
position: fixed;
top: 0;
right: 0;
```

4. 절대 위치(**absolute position**) 지정 방식은 위치가 설정된 조상(**ancestor**) 요소를 기준으로 위치를 설정된다. 하지만 위치가 설정된 조상(**ancestor**) 요소를 가지지 않는다면, HTML 문서의 body 요소를 기준으로 위치를 설정된다. 따라서 조상 요소를 기준으로 삼고 싶으면 조상 요소를 **relative**으로 설정해야 한다.

```
position: absolute;  
top: 50px;  
right: 0;
```

## ◆ 화면 중간에 글자 넣기

```
<section class="banner" id="home">  
      
    <div class = "contentBx">  
        <h2>불편한 독서모임</h2>  
        <h4>다양한 사람들과 다양한 책을</h4>  
        <a href="#" class="btn">About us</a>  
    </div>  
</section>
```

▷ 화면 중간에 글자를 넣고 싶으면 화면 중간에 넣고 싶으면 첫번째로는 화면에 배경 이미지를 추가한다.

```
.banner{  
    display: flex;  
    justify-content: center;  
    align-items: center;  
}
```

▷ .contentBx class의 position을 **relative**으로 설정하고 원하는 위치에 위치시킨다.

```
.banner .contentBx{  
    position: relative;  
    z-index: 1;  
    background: var(--white);  
    padding: 70px;  
    display: flex;  
    justify-content: center;  
    align-items: center;  
    flex-direction: column;  
}
```

▷ .contentBx class의 내용을 위로 올리고 싶으면 z-index값을 조정하면 된다.

## ◆ 버튼 만드는 법

```
.btn {
```

```

position: relative;
display: inline-block;
padding: 8px 20px;
margin-top: 30px;
border: 2px solid var(--black);
color: var(--black);
text-decoration: none;
font-size: 18px;
font-weight: 500;
letter-spacing: 2px;
transition: 0.25s;
}
.btn:hover {
  color: var(--white);
  background: var(--black);
}

```

#### ▶ 버튼 상태 스타일 바꾸기

<https://www.freecodecamp.org/korean/news/css-beotneun-seutail-hover-color-background/#:~:text=%60hover%60%20%EC%83%81%ED%83%9C%20%EA%BE%B8%EB%AF%B8%EA%B8%B0,%EC%84%A0%ED%83%9D%EC%9E%90%EB%A5%BC%20%EC%82%AC%EC%9A%A9%ED%95%A0%20%EC%88%98%20%EC%9E%88%EC%8A%B5%EB%8B%88%EB%8B%A4.>

버튼이 가질 수 있는 상태는 3가지이다.

##### 1. :hover

마우스가 올라가 있을 때 속성을 바꾸기 위해 사용되며 보통은 `transition`과 함께 사용되어 바뀌는 속도를 조절한다.

```

50
51  .navbar a {
52    color: var(--text-color);
53    font-size: 1.1rem;
54    font-weight: 500;
55    padding: 5px 0;
56    margin: 0px 30px;
57    transition: all .50s ease;
58  }
59
60
61
62
63
64
65
66  .navbar a:hover{
67    color: var(--main-color);
68  }
69
70  .navbar a.active{
71    color: var(--main-color);
72  }
73

```

##### 2. :focus

키보드를 이용해서 `tab`을 이용해 버튼이 `focus`되었을 때 속성이 바뀐다.

##### 3. :active

버튼을 눌렀을 때이다.

---

6월 18일 일

## ◆ 웹 개발의 전체 흐름

▶ frontend: 클라이언트

- 언어
  - html 뼈
  - css 디자인
  - javascript 움직이거나 동적인 부분
- SPA frontend 프레임워크
  - angular.js
  - react.js
  - vew.js
  - svelte

▶ backend: 서버

- 언어
  - js
  - java
  - go
  - php
  - python
  - kotlin
- 백엔드 웹 프레임워크: 정적 사이트 제작 가능
  - js: express
  - php: laravel
  - java: spring
  - python: django, flask

▶ 정적 사이트 vs SPA(single page application)

- 정적 사이트 : 일반적인 사이트 네이버에 들어가면 한 페이지가 보이고 다음 페이지가 보이고 등등
- SPA: 전체에서 가져오고 싶은 정보만 가져와서 바뀔 때마다 필요 정보만 바꿔고 새로고침 안 한다.

## ◆Front end roadmap

<https://roadmap.sh/frontend>

웹 개발은 그냥 JS

인공지능, 서버는 python

1. 개발 툴: VS, IntelliJ, 크롬
2. front end : html, css, JS
  - html: 태그와 페이지 구조
  - css: 스타일 레이아웃, 애니메이션
  - JS: 문법, browser API, event, DOM Manipulation
3. Github를 이용해서 버전 관리
4. 배포 전 test하는 것:
  - CI/CD
  - Test Pyramid: Jest, Cypress, Enzyme, react-testing library
5. 배포: Github page, netlify, heroku, aws

▶bootstrap이란

CSS framework으로 미이 만들어져 있는 여러 class를 가져오는 것만으로 빨리 만들 수 있다.

▶react, vue, angular, svelte

javascript framework이다. 가장 많이 쓰는 것은 react이고 요즘 뜨는 것은 svelte이다.

★SPA, SSG, SSR

ssr -> react, next.js

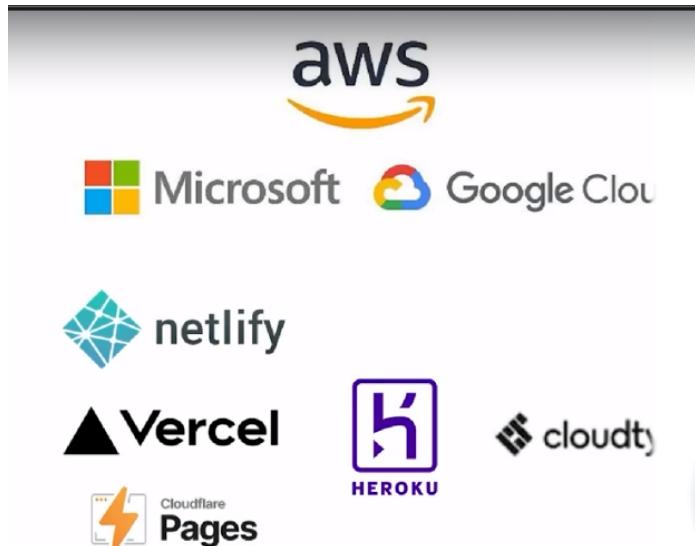
요즘 트렌드는 next.js 다른 공부하지 말고 next.js를 목표로 하자

## ◆Next.js

react + express.js + react-router-dom + server side rendering

6월 20일

## ◆ 클라우드 서버



일단 우리는 netlify를 이용하고 sites에서 내가 원하는 웹사이트를 올리고 수정을 해야 할 때는 deploy를 이용해서 수정한다.

## ◆ 도메인

.net .com .co.kr들이 도메인이다.

DNS (domain name system)이 도메인을 ip주소와 연결한다. 결국 각 ip에 대한 도메인을 DNS에 올려야 하는 것인데 개인이 하지는 않고 대행 서비스(netlify)같은 곳에서 유로로 팔고 관리도 해준다.

cafe24, gabia 한 550원이면 도메인을 이용할 수가 있다.

6월 21일

## ◆ favicon



ico converter 사용

## ◆CSS 연결하기

The screenshot shows a code editor interface with two tabs: 'index.html' and '# style.css'. The 'index.html' tab is active, displaying the following HTML code:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-
6   <title>Document</title>
7   <!-- <link rel="stylesheet" href="style.css"> -->
8   <!-- <style>
9     h1 {
10       color: red;
11     }
12   </style> -->
13 </head>
14 <body>
15   <h1 style="color: red;">사과</h1>
16   <h1>배</h2>
17   <h1>딸기</h1>
18 </body>
19 </html>
```

방법 1: style.css + link

방법 2: <style> tag를 <head>안에 넣기

방법 3: 각 태그에 속성으로 style 넣기

