

강의 계획

목표 : 이탈자 Zero

1일차

웹 상식, HTML, CSS

정적 웹 사이트 만들기

2일차

자바스크립트 기초

3일차

동적 웹 사이트 만들기

웹 상식, HTML,
CSS
정적 웹 사이트
만들기

김은찬

웹을 배워야 하는 이유

1. 웹에서 구현 할 수 있는 프로그램
 - a. Google의 Docs, Sheet, Map, Slide
2. 모바일 애플리케이션
 - a. 저전력 CPU 개발로 하드웨어 성능 향상
 - i. 모바일 웹 앱
3. 데스크탑 애플리케이션
 - a. Node.js와 Chromium 조합한 **Electron**
 - i. Visual Studio Code, Atom, MongoDB Manager
4. Web Assembly
 - a. 브라우저 성능 향상
 - b. 설치가 필요 없는 온라인 게임
 - c. OpenCV를 WebAssembly로 컴파일하여 사용 ⇒ 이미지 처리 성능 향상

인터넷과 웹

인터넷(International Network)



웹(World Wide Web)

월드 와이드 웹(WWW)

유럽 입자 물리학 연구소에서 소프트웨어 엔지니어로 취업한 **팀 버너스리**

전세계 과학자들이 입자 가속기를 사용하기 위한 **문서**를 **공유**하는 데 **문제 多**

- 로그인-로그아웃 (**1980년대**) ⇒ 컴퓨터 사용해 본적 없음
- 서로 다른 프로그램 (**1980년대**) ⇒ 사용법을 모름

⇒ **커피** 마시면서 **이야기**하는 게, 정보 공유하기 더 **쉬움**

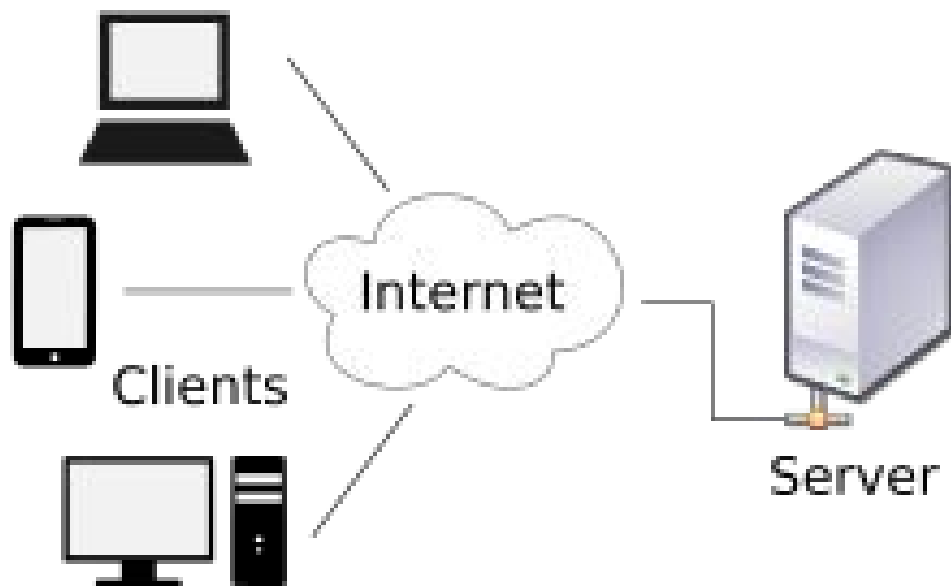
⇒ ⇒ 1990년 9월, 비공식 프로젝트로 시간 부여

WWW의 기초 설계

팀 버너스리는 지금까지도 웹의 기초가 되는 세 가지를 설계

1. HTML(HyperText Markup Language)
2. HTTP (Hypertext Transfer Protocol)
3. URI(Uniform Resource Identifier)

웹은 클라이언트-서버 모델



HTML과 CSS는 쉽다. 하지만 웹은 쉽지 않다.

1990년 말, 웹마스터라는 직업이 있었습니다. ⇒ 지금은 없습니다.

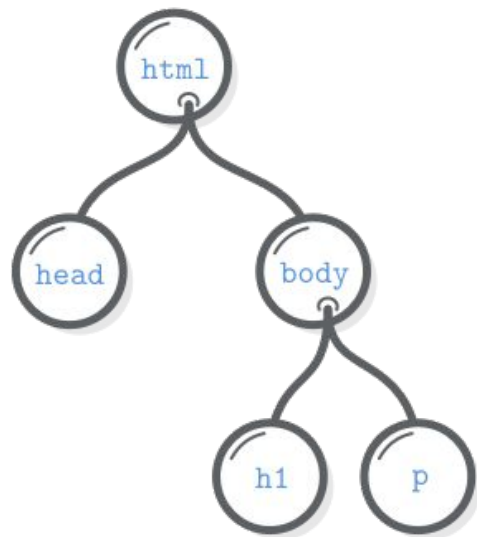
1. 프론트 엔드

- a. 기획자
- b. 디자이너
- c. 웹 퍼블리셔

2. 백 엔드

- a. 서버 전문가
 - i. 운영체제
 - ii. 검색엔진
 - iii. 각종 서버(웹 서버, GIS 서버, 검색 엔진 서버)
- b. 데이터베이스 전문가
 - i. SQL 서버
 - ii. NoSQL 서버
- c. 보안 전문가

웹 페이지 구성 요소



HTML



CSS



JAVASCRIPT

실습 1일차 - 정적 웹 사이트 (블로그) 만들기

1. Visual Studio Code 설치
 - a. <https://code.visualstudio.com/>
2. Git 설치
 - a. <https://git-scm.com/>
3. Github 페이지 만들기
 - a. <https://github.com>
 - b. 회원 가입
 - c. 프로젝트 생성
 - i. 프로젝트 명 : 아이디.github.io
4. Hello World 출력하기
 - a. 디렉토리 생성
 - i. C:\workspace\github page
 - b. index.html 파일 생성
 - i. Hello World 입력 & 저장

실습-직접 확인 하기

```
1 <!DOCTYPE html>
2 <html lang="ko">
3 <title>Hello World!</title>
4 <style>
5   #guide {
6     color : aquamarine;
7   }
8 </style>
9 <h1 id="header1">Hello World!</h1>
10 <script>
11   let h1 = document.getElementById("header1");
12   h1.addEventListener('click', (e)=>{
13     e.target.innerHTML = "Break Time";
14     e.target.id = "guide";
15   })
16 </script>
17 </html>
```

쉬는 시간! (질문하세요!)



<h1>HTML</h1>

Hyper Text?

⇒ 문서 이곳 저곳을 넘나들 수 있게하는 좋은 것! (카카오)

<h2>HTML?</h2>

⇒ 문서에 구조와 의미를 부여하는 마크업 언어에 Hyper Text만 가져다 넣은 것!

우리는 HTML로 할 일은?

⇒ 문서에 구조와 의미를 부여하기!

구조와 의미는 태그라고 불리는 것으로 부여한다.

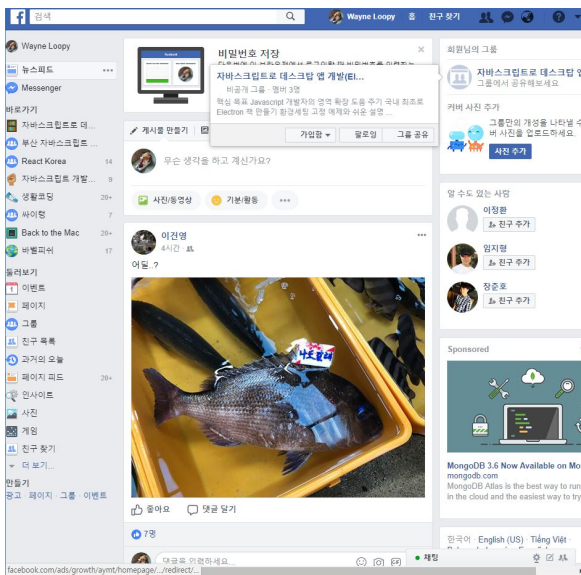
태그의 모양

- <시작>값</종료> 태그
- <단일 value="값"/> 태그

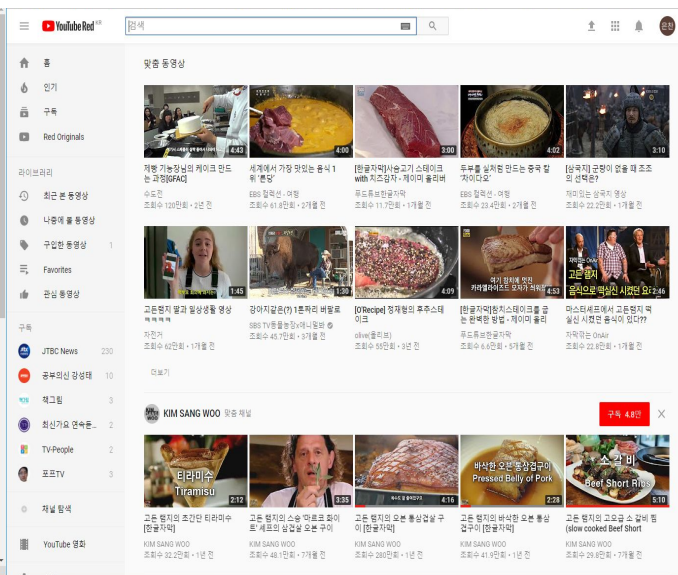
태그의 속성과 이벤트 핸들러

- 속성 : 태그에 부여된 속성
- 이벤트 핸들러 : 태그가 해당 이벤트를 어떻게 처리할 것인가
 - 함수 중에서 이벤트를 처리하는 함수를 이벤트 핸들러라고 함

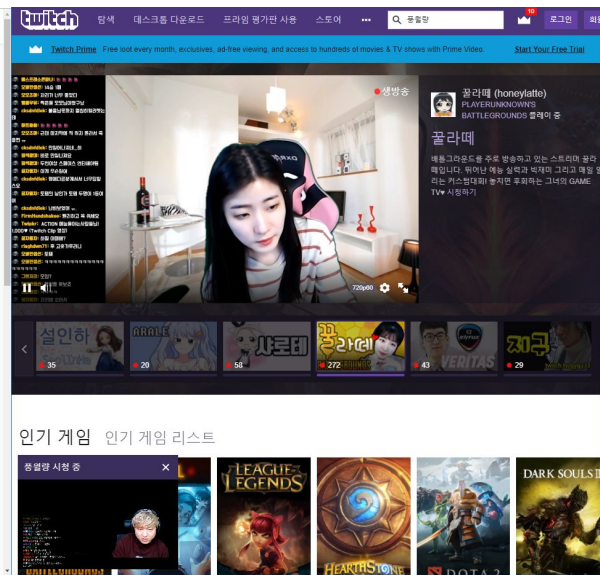
레이아웃 부터 만들어 봅시다



페이스북

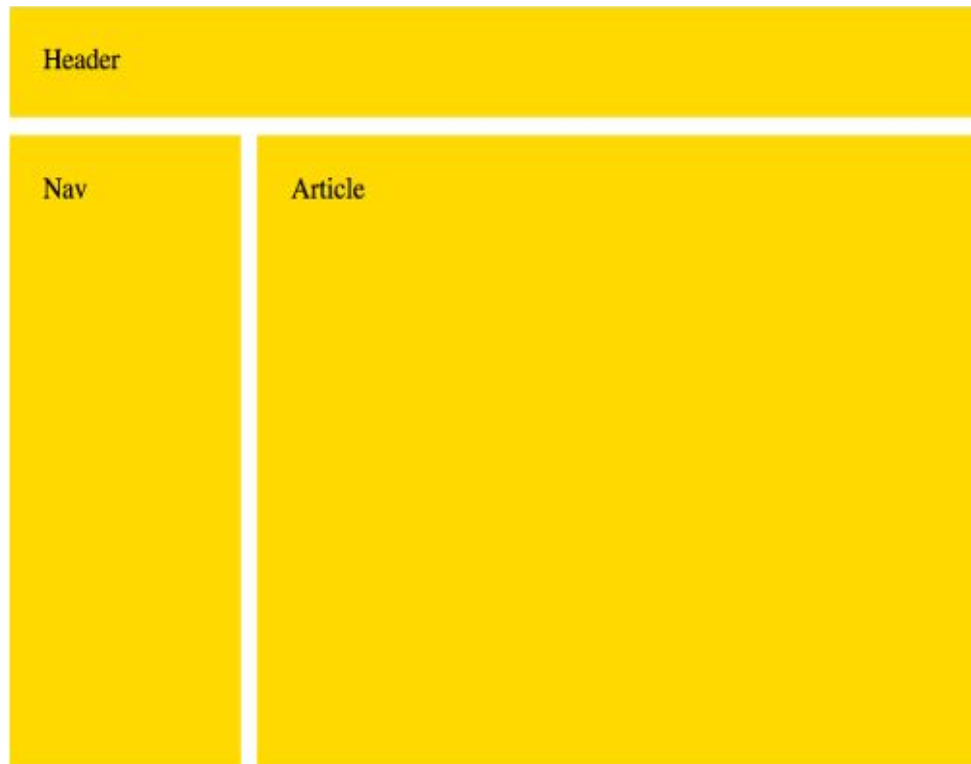


유튜브



트위치

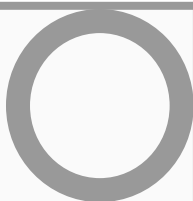
영역 나누기




실습-영역 나누기

포인트: 의미있는 태그 사용 직관성  && XHTML은 버리자! ⇒ 생산성 

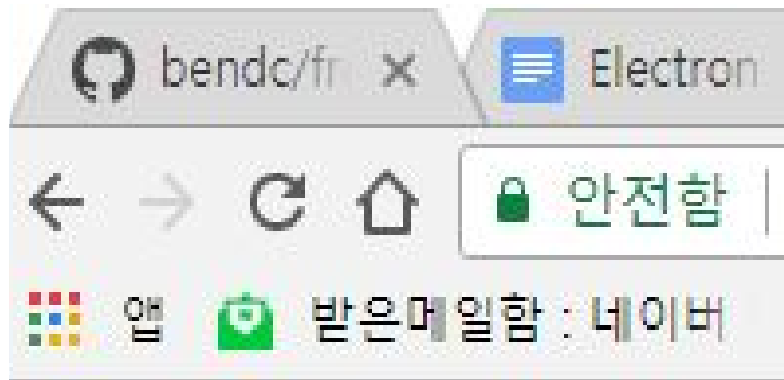
1	<!DOCTYPE html>
2	<html lang="ko">
3	<main>
4	<header>Header</header>
5	<article>Article</article>
6	<nav>Nav</nav>
7	</main>
8	</html>



1	<!DOCTYPE html>
2	<html lang="ko">
3	<head>
4	</head>
5	<body>
6	<div>Header</div>
7	<div>Article</div>
8	<div>Nav</div>
9	</body>
10	</html>



실습-HTML 작성 후 확인하기



Header
Article
Nav



CSS(Cascading Style Sheets)

영혼 가출한 HTML에 영혼(스타일) 넣어주기



CSS 구문

CSS 사용법은 아주 간단하다.

1. 스타일을 적용하려는 태그(==요소)를 생각한다.
2. 적용할 스타일을 생각한다.
3. 그리고 적는다.

다음과 같이 적으면 된다. ⇒

1	선택자 {
2	속성 : 값;
3	[속성 : 값;]
4	}

실습-영역 태그에 외부 스타일 시트 추가하기

```
1 <!DOCTYPE html>
2 <title>GitHub Page of Wayne-Kim</title>
3 <link rel="stylesheet" href="./style.css">
4 <main id="body">
5   <header id="pageHeader">Header</header>
6   <article id="mainArticle">Article</article>
7   <nav id="mainNav">Nav</nav>
8 </main>
```

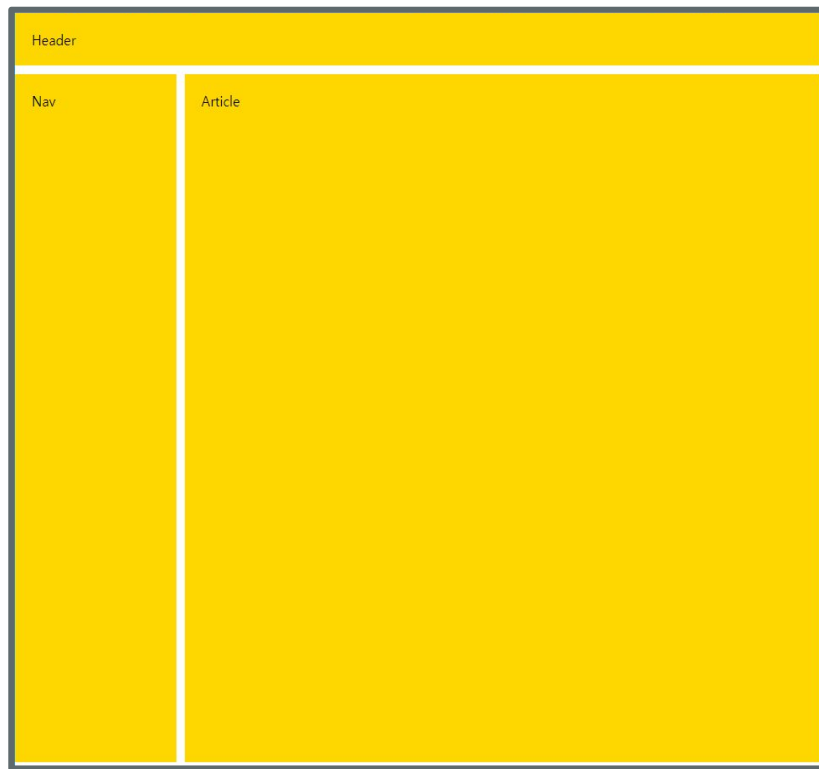
실습-외부 스타일 시트에 스타일 정의하기

포인트 : grid-template-areas 속성으로 쉽게 레이아웃 지정

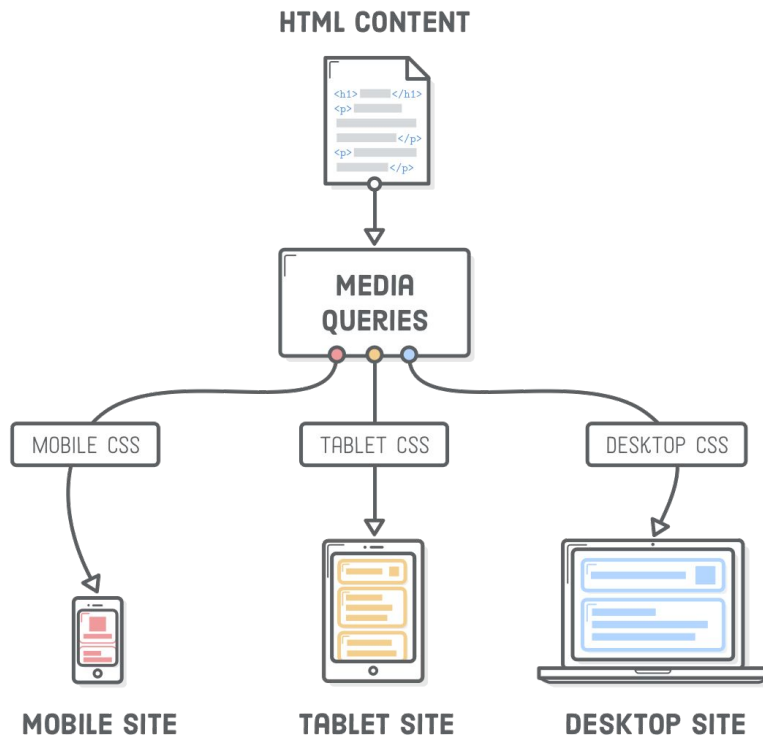
```
1  #body{
2    display: grid;
3    grid-template-areas: "header header" "nav article";
4    grid-template-rows: 60px 1fr;
5    grid-template-columns: 20% 1fr;
6    grid-gap: 10px;
7    height: 100vh;
8    margin: 0;
9  }
10 #pageHeader{
11   grid-area: header;
12 }
13 #pageFooter{
14   grid-area: footer;
15 }
```

```
16 #mainArticle{
17   grid-area: article;
18 }
19 #mainNav{
20   grid-area: nav;
21 }
22 header, footer, article, nav, div{
23   padding: 8px;
24   background: gold;
25 }
```

실습-외부 스타일 시트 추가 결과 확인



반응형 레이아웃



실습-반응형 레이아웃 스타일 정의하기

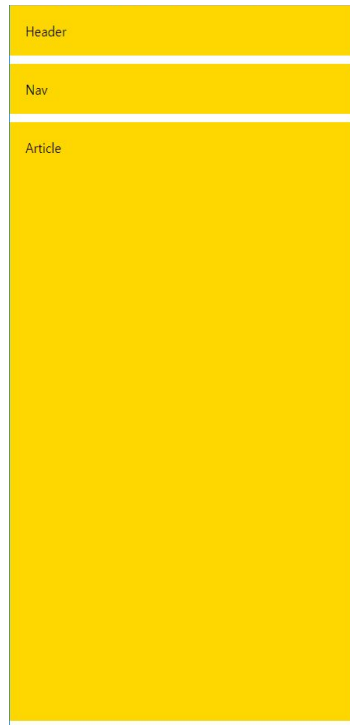
```
31 @media all and (max-width: 575px) {  
32   #body {  
33     grid-template-areas: "header" "nav" "article";  
34     grid-template-rows: 60px 60px 1fr;  
35     grid-template-columns: 1fr;  
36   }  
37 }
```

style.css

```
3 <meta name="viewport" content="width=device-width, user-scalable=no">
```

index.html

실습-반응형 레이아웃 스타일 결과 확인



헤더에 필요한 HTML 태그는 ?

- Home
- 검색
- 로그인

⇒ 이 태그들 한테 필요한 부가적으로 필요한 것은? 이미지!

검색 이미지 : <https://www.w3schools.com/howto/searchicon.png>

로그 이미지 : 여러분이 쓰고 싶은 걸로! 35x35px로 편집!



실습-헤더에 필요한 태그 추가하기

```
1 <!DOCTYPE html>
2 <title>GitHub Page of Wayne-Kim</title>
3 <meta name="viewport" content="width=device-width, user-scalable=no">
4 <link rel="stylesheet" href="/style.css">
5 <main id="body">
6   <header id="pageHeader">
7     <a id="logo" href="/index.html"></a>
8     <input type="text" name="검색" placeholder="검색.." id="search-box" />
9     <input type="button" name "로그인" value="로그인" / id="login-btn">
10  </header>
11  <article id="mainArticle">Article</article>
12  <nav id="mainNav">Nav</nav>
13  <footer id="pageFooter">Footer</footer>
14 </main>
```


실습-헤더에 추가한 태그에 스타일 추가하기

```
42  /* 로고 */
```

```
43  #logo {
```

```
44    border: 2px solid #ccc;
```

```
45    border-radius: 4px;
```

```
46    margin-right: 10px;
```

```
47    background-color: white;
```

```
48    background-image: url('./logo.png');
```

```
49    background-position: 2px 5px;
```

```
50    background-repeat: no-repeat;
```

```
51    padding: 10px 19px 11px;
```

```
52  }
```

```
54  /* 검색 창 */
```

```
55  #search-box {
```

```
56    width: 150px;
```

```
57    border: 2px solid #ccc;
```

```
58    border-radius: 4px;
```

```
59    font-size: 16px;
```

```
60    background-color: white;
```

```
61    background-image: url('./searchicon.png');
```

```
62    background-position: 10px 10px;
```

```
63    background-repeat: no-repeat;
```

```
64    padding: 13px 20px 11px 40px;
```

```
65  }
```

```
67  /* 로그인 버튼 */
```

```
68  #login-btn {
```

```
69    float : right;
```

```
70    clear : both;
```

```
71    border: 2px solid #ccc;
```

```
72    border-radius: 4px;
```

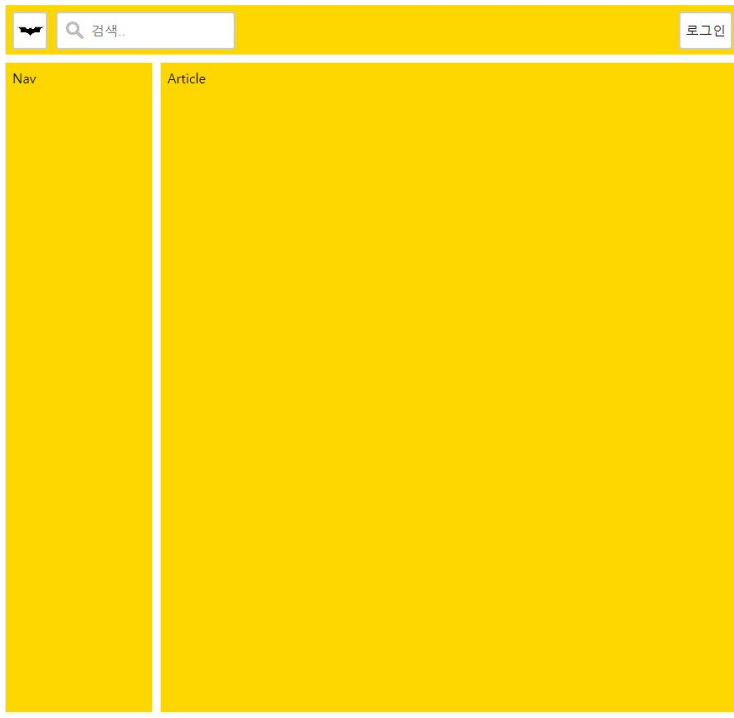
```
73    font-size: 16px;
```

```
74    background-color: white;
```

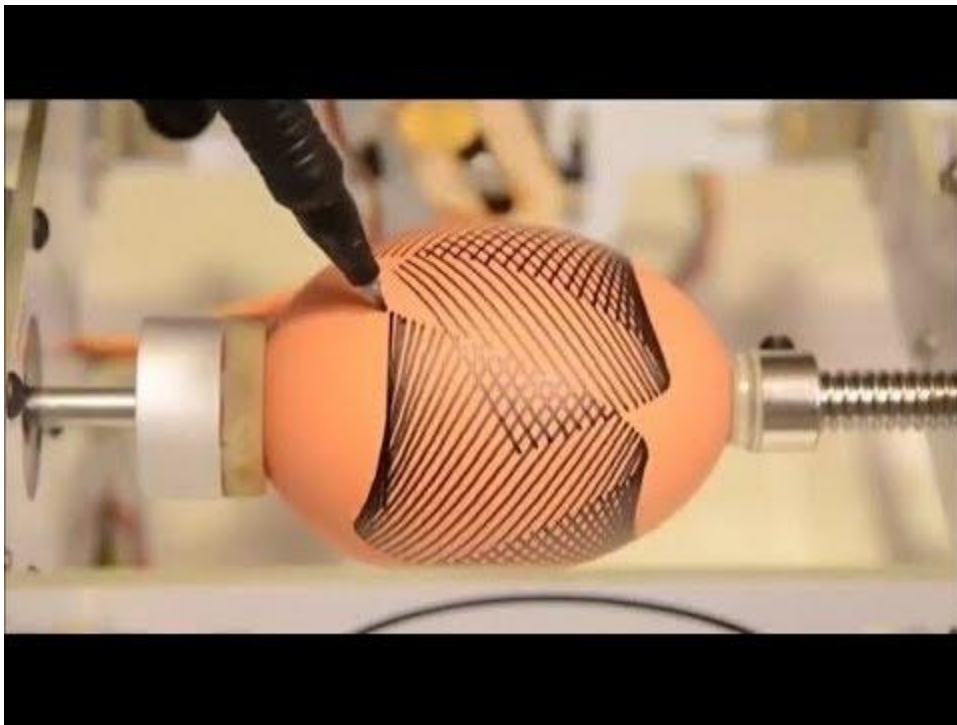
```
75    padding: 10px 6px 11px;
```

```
76  }
```

실습-헤더에 스타일 추가 후 결과 확인



쉬는 시간!(질문하세요!)



네비에 필요한 HTML 태그는 ?

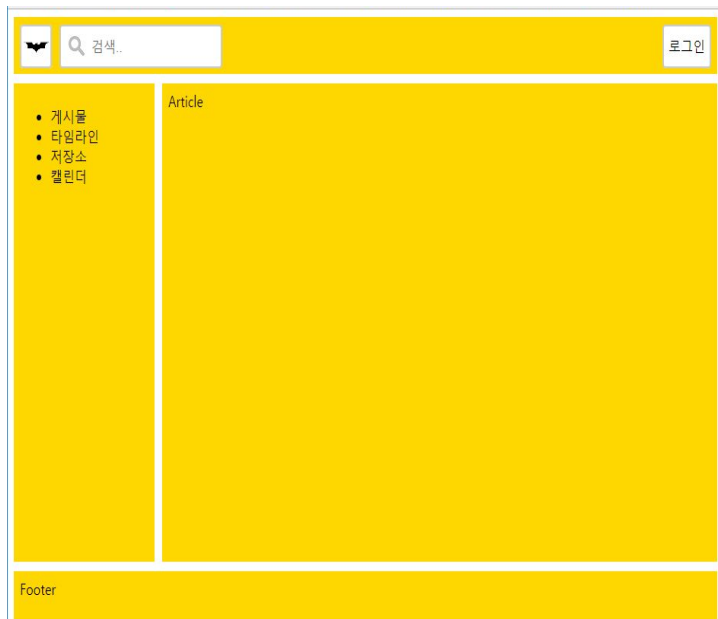
- 게시물
- 타임라인
- 저장소
- 캘린더
- To Do List

순서가 없는 네비임으로 태그가 적합하다!

실습-네비에 필요한 태그 추가하기

```
13 <nav id="mainNav">
14   <ul>
15     <li>게시물</li>
16     <li>타임라인</li>
17     <li>저장소</li>
18     <li>캘린더</li>
19   </ul>
20 </nav>
```

실습-네비에 추가한 태그 보기



반응형 웹을 고려하지 않을 수 없다.

네비에는 두 개의 스타일 시트가 필요하다.



DESKTOP SITE

```
1 /* 네비 틀 */
2 #mainNav ul {
3   list-style-type: none;
4   padding: 0;
5   margin: 0;
6 }
7 /* 네비 항목 */
8 #mainNav ul li {
9   border-bottom: 1px solid #efefef;
10  border: 2px solid #ccc;
11  border-radius: 4px;
12  background-color: white;
13  font-size: 16px;
14  padding: 10px 6px 11px;
15  margin-bottom: 8px;
16 }
```

```
17 /* 네비 틀 */
18
19 #mainNav ul li:before{
20   content: ">";
21   padding: 0px 5px 6px 0px;
22 }
23 #mainNav *::selection{
24   background: rgba(255, 255, 255, 0.1);
25 }
```

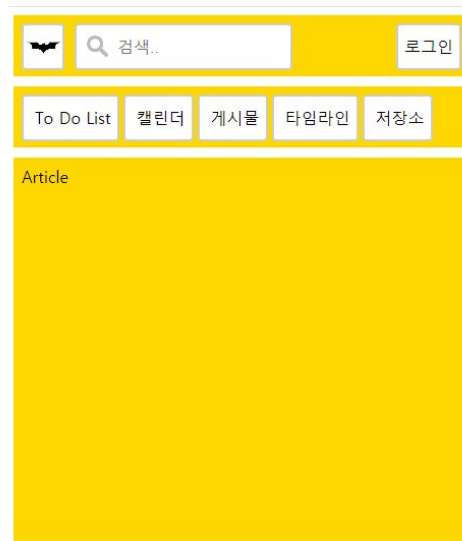
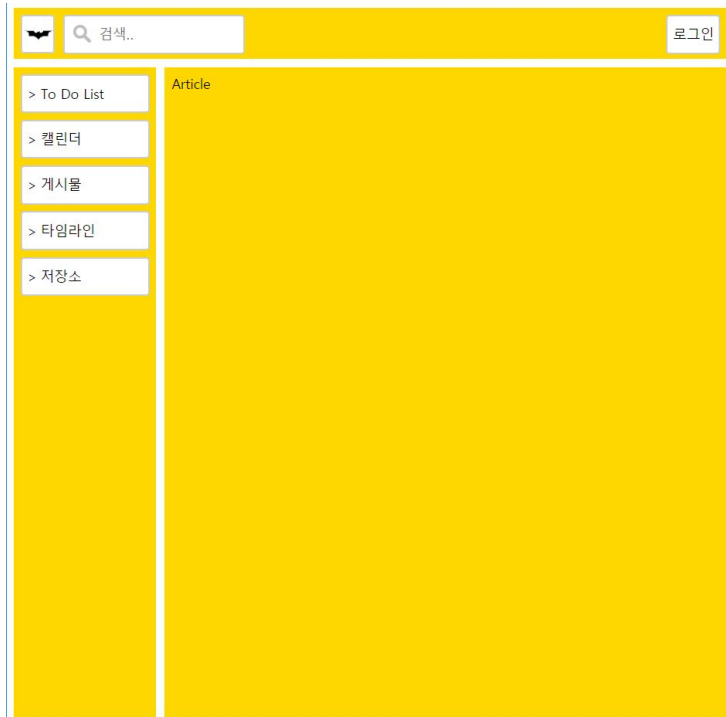


MOBILE SITE

```
1 @media all and (max-width: 520px) {
2   /* 중복 생략 */
3   #mainNav ul {
4     display: flex;
5     list-style-type: none;
6     padding: 0;
7     margin: 0;
8   }
9   #mainNav ul li:before{
10    content: "";
11    padding: 0px 5px 6px 0px;
12  }
```

```
13 #mainNav li{
14   border: 2px solid #ccc;
15   border-radius: 4px;
16   background-color: white;
17   font-size: 16px;
18   cursor: pointer;
19   margin-right: 4px;
20 }
21 }
```

실습-네비에 스타일 추가 후 확인



기사에 필요한 HTML 태그는 ?

우선 무엇을 넣을지 생각하자!

로그아웃 상태

- 공개된 게시물
 - 구현하자.

로그인 상태

- **To Do List**, 캘린더, 게시물, 타임라인, 저장소 요약본
 - 서버가 없음으로 2일차 때 자바스크립트를 배우고 3일차 때 **Node**를 배우고 구현한다.


실습-기사에 필요한 태그 추가하기

```
24 <article id="mainArticle">
25   <!-- 로그아웃 상태, 공개 게시물 보기 -->
26   <article class="board">
27     <article class="post">
28       <header>게시물 제목</header>
29       <article>게시물 내용</article>
30       <footer>게시물 댓글</footer>
31     </article>
32     <article class="post">
33       <header>게시물 제목</header>
34       <article>게시물 내용</article>
35       <footer>게시물 댓글</footer>
36     </article>
37   </article>
38 </article>
```

실습-바로 CSS 추가

```
1  /* 게시판 */
2  .board {
3      border-bottom: 1px solid #efefef;
4      border: 2px solid #ccc;
5      border-radius: 4px;
6      background-color: white;
7  }
8  .post {
9      border: 2px solid #ccc;
10     border-radius: 4px;
11     background-color: white;
12     margin-bottom: 8px;
13 }
14 .post * {
15     border-bottom: 1px solid #efefef;
16 }
```


실습-다음과 같이 UI를 만드세요.



로그인

> To Do List

> 캘린더

> 게시물

> 타임라인

> 저장소

게시물 제목

게시물 내용

게시물 댓글

게시물 제목

게시물 내용

게시물 댓글

게시물 제목

게시물 내용

게시물 댓글

게시물 제목

게시물 내용

게시물 댓글



로그인

To Do List

캘린더

게시물

타임라인

저장소

게시물 제목

게시물 내용

게시물 댓글

게시물 제목

게시물 내용

게시물 댓글

게시물 제목

게시물 내용

게시물 댓글

게시물 제목

게시물 내용

게시물 댓글

GitHub Page에 코드 적용하기

1. `git config --global user.email "you@example.com"`
2. `git config --global user.name "Your Name"`
3. `git init`
4. `git add *`
5. `git commit -m "first commit"`
6. `git remote add origin https://github.com/wayne-kim/wayne-kim.github.io.git`
 - a. GitHub 로그인 요청이 옵니다.
 - b. 로그인 합니다.
7. `git push -u origin master`

내가 만든 페이지 인터넷으로 보기

"아이디.github.io" url로 접속

고생하셨습니다.

최고의 자습서

- <https://www.w3schools.com/>

최고의 연습

- https://www.w3schools.com/html/html5_syntax.asp
- <http://www.tothenew.com/blog/10-best-practices-in-css/>

최고의 가이드

- <https://github.com/bendc/frontend-guidelines>