

코드 라이언  
**HTML / CSS** 강의 수강은  
잘 하셨나요? 😎

개발을 처음 해보시는 분들에게는  
내용이 다소 낯설기도 하고  
어려웠을지도 몰라요 🥲

그래도, 강의에서 다룬 코드들을  
다시 작성해보면서  
주기적으로 복습하다 보면,  
금방 익숙해질 거예요 😊

그래서,  
오늘 다루게 될 내용은  
**HTML, CSS** 보충입니다 !

오늘의 세션 🖥️

1. Semantic Tags
2. 반응형 웹 / 부트스트랩



그렇다면, **Semantic Tag**(시맨틱 태그) 먼저  
알아보도록 할까요?? 🤠

# Semantic Tag



Semantic Tag (시맨틱 태그)? 🤔



말 그대로, **의미가 있는 태그** 라는 뜻이다.  
**고유의 이름**을 가지며, 자신만의 의미를 갖는 태그!

이렇게 HTML의 여러 태그들 중에서도,  
**자기 나름대로의 의미를 가지고 있는 태그**들을 바로  
**Semantic Tag** 라고 한다!

Q.

의미가 있는 태그? 음.. 아직 감은 완벽하게 잘 안 잡히지만..

아니 그런데, 시맨틱 태그는 우리가 왜 써야하는거죠? 😐

1. div 태그 혹은 span 태그로 감싸서
2. 클래스 지정해준 다음,
3. CSS 적용해주면..

끝! 아닌가요?! 😐  
애 없어도 개발 잘 되는데요?!

A.  
맞아요

사실 웹 브라우저에 보이는 화면 자체는  
시맨틱 태그를 사용할 때와 사용하지 않을 때  
별 차이를 보이지 않아요..

하지만  
우리가 시맨틱 태그를 사용해야하는 이유는  
크게 세 가지

1. SEO
2. 유지보수
3. 웹 접근성

SEO?  
웹  
접근성?

아마 몇몇 분들은 몇 번 들어보셨을 용어들이에요

하지만, 처음 들어보는 🦁님들을 위해

**차근차근** 설명해드릴게요!

그렇다고 너무 겁먹지 말아요 😓

외울 필요도 없고, **“그렇구나!”** 이해만 해도 충분하답니다 😊

“난 이미 알고있다!”

하시는 분들은

‘30페이지’로 넘어가시면 돼요 😊

1. SEO

2. 유지보수

3. 웹 접근성

# SEO (Search Engine Optimization) = 검색 엔진 최적화

특히나 개발자, 기획자들 사이에서는  
종종 등장하는 용어입니다!



# SEO (Search Engine Optimization) = 검색 엔진 최적화

이번 세션에서는 깊게 다루지 않고, 간단하게 설명만 할게요!  
SEO에 대해 더 깊이 알아보고 싶다면,

★구글링★

(참고로, 코딩할 때 구글링을 잘 하는 것도 매우 중요하답니다!  
구글링이 나쁜 것이 절대 아니에요!

이 많은 정보들을 우리는 천재가 아닌 이상, 절.대! 다 외울 수 없습니다  
처음 배울 때는 “이러한 이유로 이렇게 있구나”를 충분히 이해하고,  
그 이후에는 상황에 맞게 구글링을 할 줄 알면 충분해요 😎)

# SEO (Search Engine Optimization) = 검색 엔진 최적화

웹 문서를 **검색 엔진이 이해하기 쉽도록**  
구성함으로써,

우리가 만든 웹 서비스(=웹 페이지)가 검색 결과로  
최대한 **상위에 노출될 수 있도록**  
최적화하는 것을 의미

시맨틱 태그는  
이러한 **SEO**를 도울 수 있어요

즉,

★상단에 검색 노출★ 시킬 수 있는 확률을 높일 수 있겠죠?

이 **SEO**(검색 엔진 최적화)는  
우리가 만든 서비스를 운영할 때  
중요한 부분 중에 하나랍니다!

1. SEO

2. 유지보수

3. 웹 접근성

그런데, 우리가 웹 서비스를 한 번 완성하면 끝일까요? 🤔

아닙니다!

서비스를 운영하다 보면,  
기획 부분에 변화가 생길 수도 있고,  
디자인 부분에서도 수정해야 할 상황이 생겨요

따라서 서비스를 운영하는 중에도, 기획 / 디자인 팀의 요구에 따라  
코드를 수정해야하는 상황이 자주 발생해요 😊

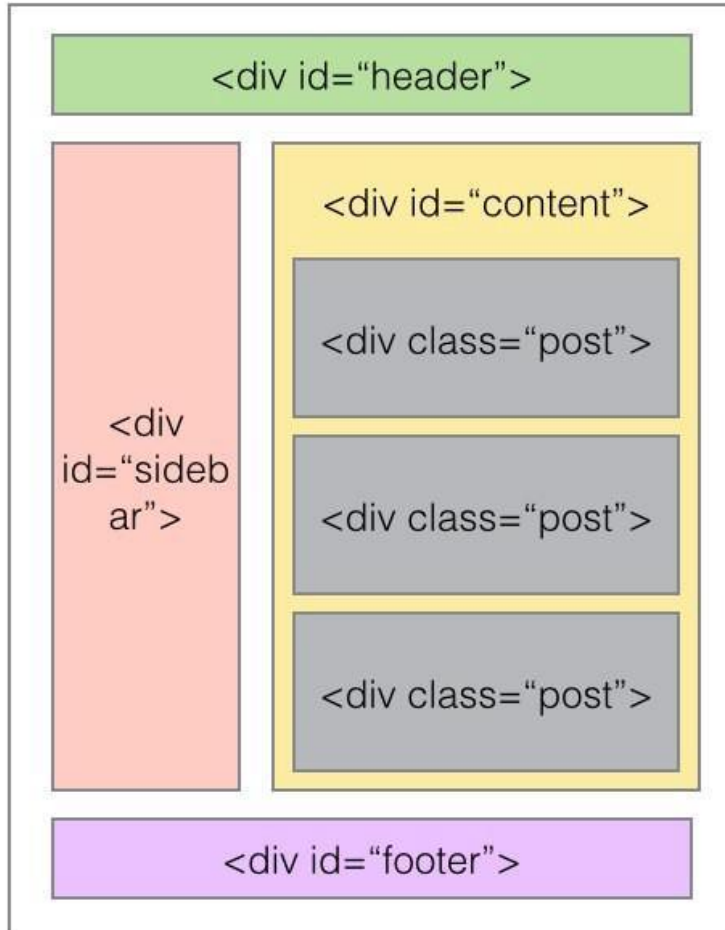
그래서  
처음 코드를 **지저분**하게 작성하게 되면  
나중에 해당 코드로 다시 작업할 때,

가독성이 떨어져서  
코드를 이해하기도 어렵고,  
수정하기도 힘들겠죠?😂

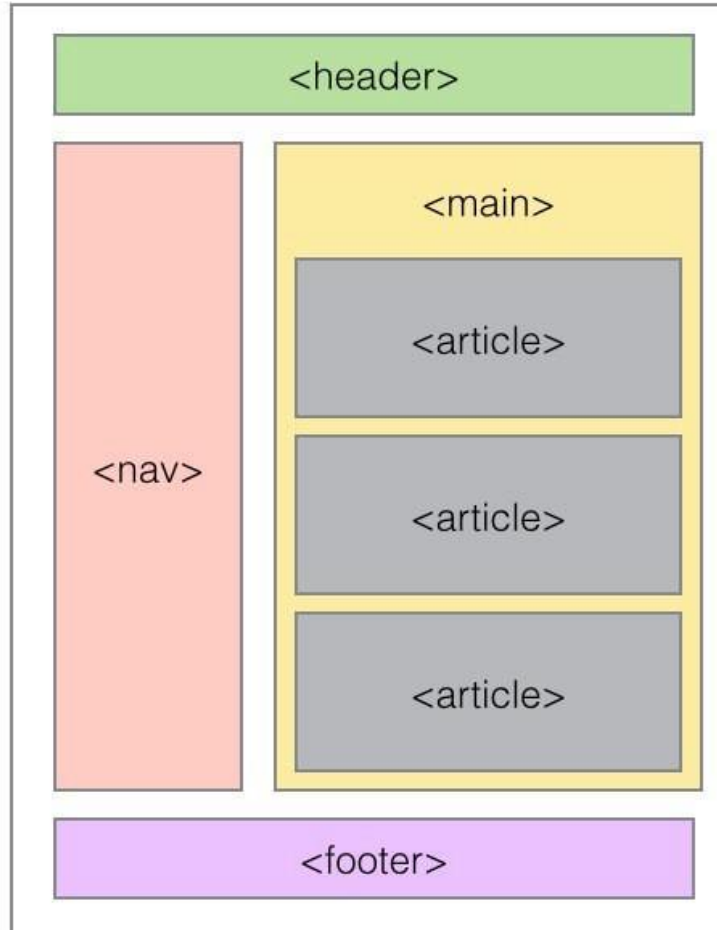
한번 예시를 볼까요?


## HTML4 vs HTML5 Page Structure on a Blog


### HTML4: Lots of Classes/IDs



### HTML5: Semantic Tags/Sections



 시맨틱 태그가 없을 때는  
div나 span으로 감싸고,  
각 태그에 id 혹은 class를 지정해주면서  
코드도 길어지고 복잡해지는 반면,

 시맨틱 태그를 사용했을 때는  
**class 지정 없이도**  
각 태그가 어떤 역할을 하는지  
한눈에 이해하기 쉬워졌고 코드도 깔끔해졌어요



따라서  
코드의 **가독성**이 높아지고  
그만큼 **유지보수**도 쉬워지겠죠? 😊

1. SEO
2. 유지보수
3. 웹 접근성

만약 사용자가 웹 사이트를 이용할 때,  
스크린 리더 혹은 키보드만을 사용하더라도  
시맨틱 태그를 활용한 웹 사이트에서는  
문제 없이 동작 시킬 수 있어요 😮

여기서 이야기하는 스크린 리더는 주로 시각 장애인이 사용하는데요,  
따라서 장애인 분들도 웹 서비스를 문제없이 이용할 수 있을거예요  
즉, 웹 접근성이 높아진다는 것이죠! 😊

시맨틱 태그를 왜 사용해야 하는지, 이제 이해가 되셨나요?! 😏

그렇다면 이제 대표적인 시맨틱 태그들에는 어떤 것이 있는지,  
또 각각 그 쓰임새는 무엇인지  
알아보도록 해요! 😊

1. header
2. nav
3. main
4. section / article
5. aside
6. footer

+ 더 많은 시맨틱 태그를 알아보고 싶다면,

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

1. **header**
2. **nav**
3. **main**
4. **section / article**
5. **aside**
6. **footer**

## 1. header

웹 사이트를 소개하는 콘텐츠를 주로 나타낸다.  
본래 제목(<h1> ~ <h6>)을 감싸는 용도이지만, 필수  
사항 아님

따라서, 페이지 상단에 웹 사이트 로고와 네비게이션  
링크를 감싸는 용도로 사용하기도 한다





1. header
2. nav
3. main
4. section / article
5. aside
6. footer

## 2. nav

사이트 내에 필요한 **링크를 모아둔 태그**  
보통 `<ul>`태그로 사이트 내부 링크들을 감싸서 사용  
(다음 페이지 예시 참고)



# <nav> 태그 예시 (부트스트랩)

코드에 해당하는 부분을  
서로 동일한 색상으로  
표시했어요

Navbar

Home

Features

Pricing

Disabled

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="#">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Features</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Pricing</a>
        </li>
        <li class="nav-item">
          <a class="nav-link disabled" href="#">Disabled</a>
        </li>
      </ul>
    </div>
  </div>
</nav>
```

이 코드는 지금 다 이해하실 필요는 없고,  
“대충 이런 식으로 사용되는구나”  
하고 넘어가시면 됩니다 😊

1. header
2. nav
3. **main**
4. section / article
5. aside
6. footer

### 3. main

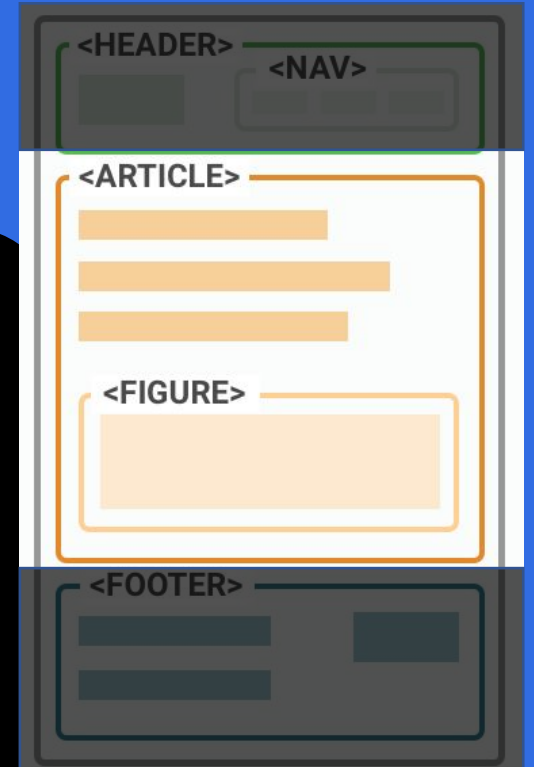
웹 사이트의 **본문**을 의미  
메인 콘텐츠를 가장 크게 감싸주는 역할을 하며,  
**하나의 HTML 문서당 한 개만 존재**



1. header
2. nav
3. main
4. **section / article**
5. aside
6. footer

#### 4. section / article

페이지 내에서 **각 콘텐츠의 영역을 구분**하는 데 사용



1. header
2. nav
3. main
4. section / article
5. **aside**
6. footer



## 5. aside

문서의 주요 내용과 **직접적인 연관이 없는 부분**을  
나타냄 (ex. 광고 등)  
<main>이나 <article> 내부에서 사용될 수 있음



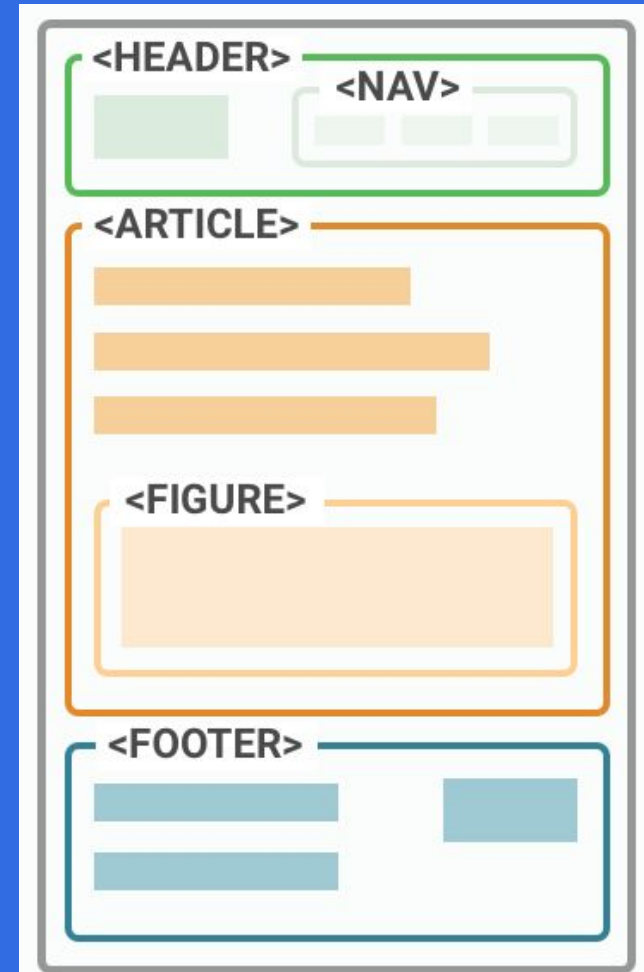
1. header
2. nav
3. main
4. section / article
5. aside
6. footer

## 6. footer

웹 페이지의 **가장 하단에 위치**  
회사 정보나 약관 정보 등의 콘텐츠를 표시하는 데  
사용됨



앞서 살펴본 시맨틱 태그들을 사용하면,  
다음과 같은 구조로 HTML 문서를 구성할 수 있어요!



## \* TO DO \*

이제 앞서 배운 시맨틱 태그를 이용해서  
간단한 웹 사이트의 뼈대 코드를 짜볼 거예요 🥰

앞으로 배울 자바스크립트, 부트스트랩 등을 이용해서  
디자인, 기능까지도 계속해서 하나씩 추가할  
것입니다 😊

먼저, 우리가 앞으로 만들게 될 최종 완성본을 한번 볼까요?

## 간단한 자기소개

이름❤️ 박민선

mbti❤️ ISTP

학교❤️ 한양대학교 ERICA캠퍼스

학과❤️ 소프트웨어학부 컴퓨터전공

취미❤️ 나들이 가기

좋아하는 음식❤️ 일식, 고기 등

## 나의 시간표

시간	일	월	화	수	목	금	토
9:00-10:00							
10:00-11:00							
11:00-12:00							
12:00-13:00							
13:00-14:00							
14:00-15:00							
15:00-16:00							
16:00-17:00							
17:00-18:00							

멋쟁이 사자처럼 10기

박민선

디자인은 부트스트랩을 배우고 나서,  
시간표 기능은 JS를 배우고 나서 추가할 것입니다!

우선 지금은,  
시맨틱 태그를 이용해서 **html 뼈대 코드**를 작성하는  
것에만 집중하면 된답니다 😊

앞 페이지에서 보여드린 예시를 보면서,  
**header, main, section, article, footer** 등의  
시맨틱 태그들을 **자율적으로 활용**해서  
뼈대 코드를 작성해 보아요 😊

기본 뼈대 구조 예시는 다음다음 페이지(51페이지)에 있으니 참고하셔도 좋습니다!

기본 뼈대 코드를 완성하셨다면,  
앞선 예시처럼 **본인을 소개하는 글도 작성**해서  
넣어주세요!



여기는 header

여기는 main

여기는 자기소개 article 혹은 section

여기는 시간표 article 혹은 section

여기는 footer

여기는 header

여기는 main

여기는 자기소개 article 혹은 section

여기는 시간표 article 혹은 section

여기는 footer

여기에 내용물을 넣으면

## 헤더

# 간단한 자기소개

---

이름 📄 박민선

mbti 📄 ISTP

학교 📄 한양대학교 ERICA캠퍼스

학과 📄 소프트웨어학부 컴퓨터전공

취미 📄 나들이 가기

좋아하는 음식 📄 일식, 고기 등

## 나의 시간표

---

푸터

헤더  
간단한 자기소개

이름 ❤ 박민선

mbti ❤ ISTP

학교 ❤ 한양대학교 ERICA캠퍼스

학과 ❤ 소프트웨어학부 컴퓨터전공

취미 ❤ 나들이 가기

좋아하는 음식 ❤ 일식, 고기 등

나의 시간표

푸터

참고로 각 태그의 배치는, 바로 다음에 이어지는  
부트스트랩 강의 이후에 진행할 것이기 때문에  
생김새가 최종 완성본과 다르다고  
당황하지 마세요! 😊

그리고 꼭 예시와 똑같이 안 만들어도 돼요!  
자신이 원하는 대로, 자율적으로 만들어보세요 😊

# 반응형 웹



## 반응형 웹

많은 분들이 익숙하게 들어봤을 용어인데요!  
그만큼 요즘 웹 개발에서 정말정말 많이 사용되고 있어요 😊

그렇기 때문에 반응형 웹은  
기획, 디자인, 개발 파트 모두에게 중요한 개념입니다 😮

## 기획자는

우리의 웹 서비스 특성에 맞게  
반응형 웹, 적응형 웹 등 여러 방식 중 하나를 선택하여  
서비스를 기획해야 하고,

참고로 적응형 웹은 세션에서 다루지 않아요! 궁금하다면 구글링!

## 디자이너 / 개발자는

반응형 웹 특성에 알맞는 작업물을 내야해요 😊

반응형 웹에 대한 자세한 특징 또한, 세션에서 다루지 않아요! 궁금하다면 구글링!

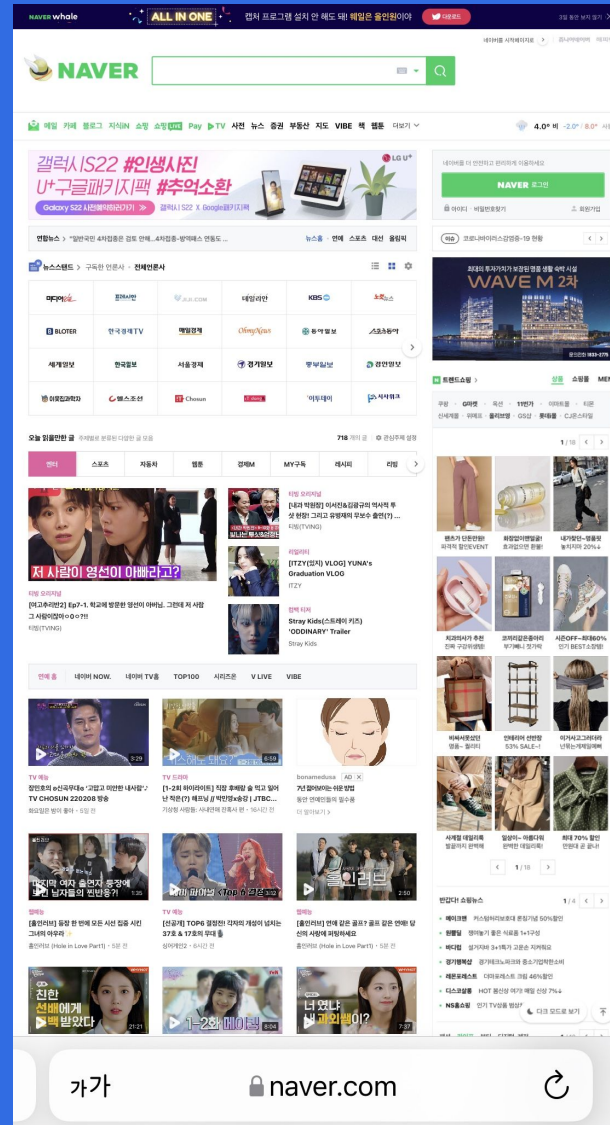
그렇다면 반응형 웹이란 것이 왜 탄생했을까요? 🤔

그것은 바로 모바일 기기의 등장!  
핸드폰, 태블릿 등과 같은 모바일이 등장하면서,  
디바이스의 크기도 그만큼 다양해졌기 때문이에요 🌟

디바이스가 다양해짐에 따라  
디바이스 크기에 따라 화면 구성(=HTML 구성)을 달리해야 했는데요,  
예시를 볼까요?



# 모바일 기기에서 접속한 www.naver.com (=pc버전 화면)



참고로 네이버에서는 반응형 웹이 아닌,  
적응형 웹을 사용하고 있어요

쉽게 풀어서 얘기하자면,  
네이버 사이트에 진입할 시에  
모바일 기기일 경우에는 m.naver.com으로,  
데스크탑일 경우에는 www.naver.com으로  
연결되도록 되어있답니다

적응형 웹에 대해서 더 궁금하시다면  
구글링을 통해 알아보세요!  
저희 세션에서는 자세히 다루지 않아요 😊

디바이스 크기에 비해  
화면에 담긴 정보가 너무 많아서  
사용자 경험(UX)이 많이 안 좋아졌어요 🥵

그래서  
모바일 기기에 알맞은 별도의 화면을 새롭게 구성함으로써  
해결책을 제시하게 됐어요

즉, 반응형 웹은  
간단히 말해서  
뷰포트(=사용자의 화면 크기)에 따라  
화면을 달리 구성한 것이라고 할 수 있어요

우선, 반응형 웹을 위해 탄생한  
**CSS 레이아웃 속성들**을 간단하게 알아볼 거예요!

반응형 웹에서 흔히 사용되는 CSS 레이아웃 속성에는 대표적으로  
**flex box, grid**가 있는데요,

저희 세션에서는 **flex box**만 간단하게 다루고,  
“Grid도 배우고 싶다!” 하시는 분들은 ★구글링★ 추천 드립니다!

## 🌻Flex Box🌻

우선 flex box는 **일직선 상**에 존재하는 태그들의 배치를  
편리하게 해주는 역할을 하는데요,

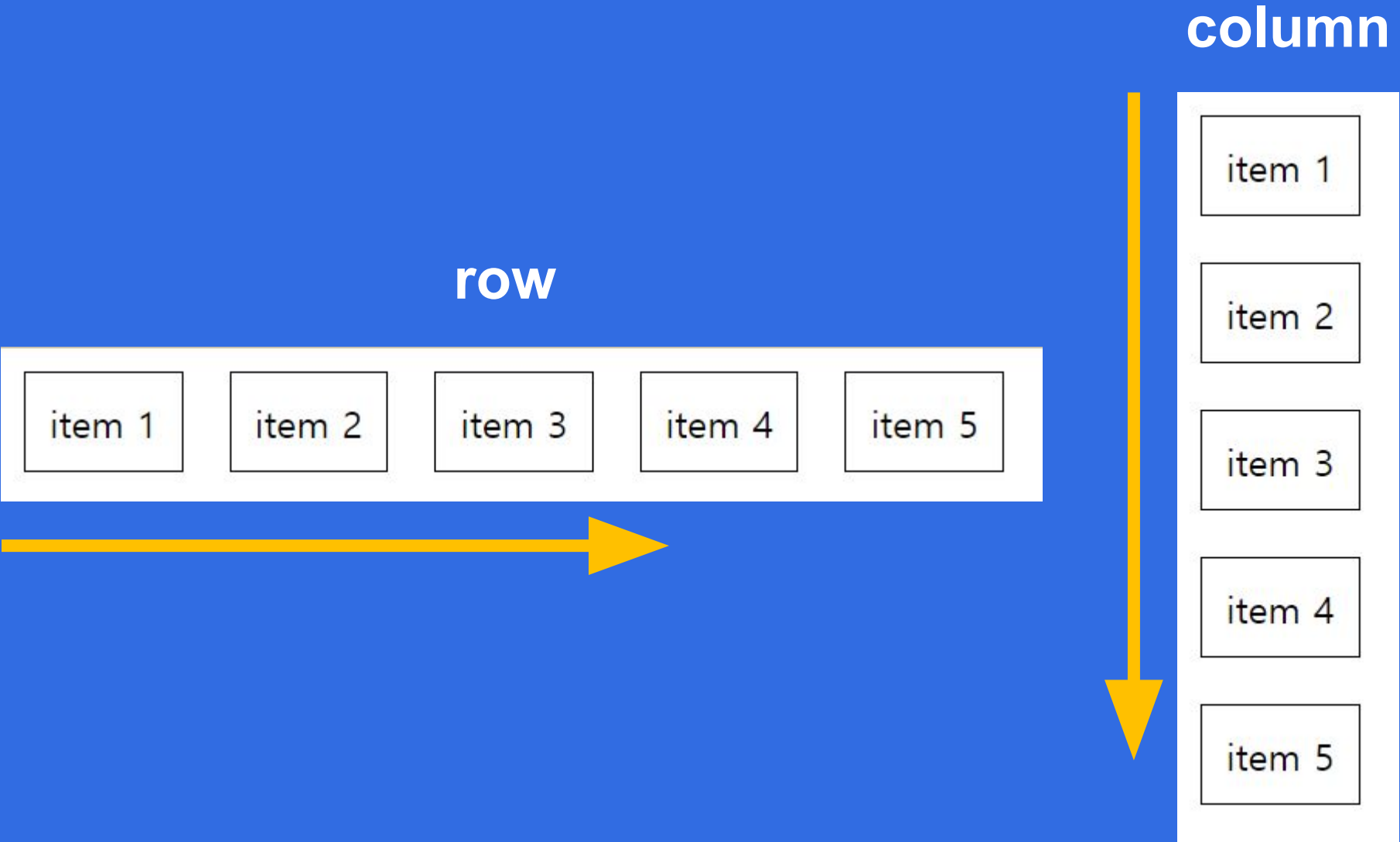
참고로 grid는 이차원(행+열) 배치를 다룹니다!

이 뿐만 아니라  
**화면이 변화할 때** 내부 태그들을 **어떻게 변형/배치**할 것인지  
지정할 수도 있습니다 😲

## 🌼Flex Box🌼

앞서 언급했듯, flex box는 일직선 상의 배치를 다룬다고 했습니다!  
그렇다면 먼저 어떤 방향으로 배치를 할 것인지 지정해줘야겠죠?

먼저 flex box에서의 방향은 가로(row), 세로(column) 두 가지가 있어요  
여기서 default 값은 가로(row) 방향입니다😊





## 🌸 Flex Box 사용하기 🌸

그렇다면 이 flex box를 어떻게 사용하는지도 알아야겠죠?  
먼저 flex box를 사용할 때는  
부모 태그에 **flex box**를 사용하겠다고! 라고 명시해야해요

```
<div id="container">  
  <div class="item">item 1</div>  
  <div class="item">item 2</div>  
  <div class="item">item 3</div>  
  <div class="item">item 4</div>  
  <div class="item">item 5</div>  
</div>
```

```
#container {  
  display: flex;  
}
```

## 🌸 Flex Box 사용하기 🌸

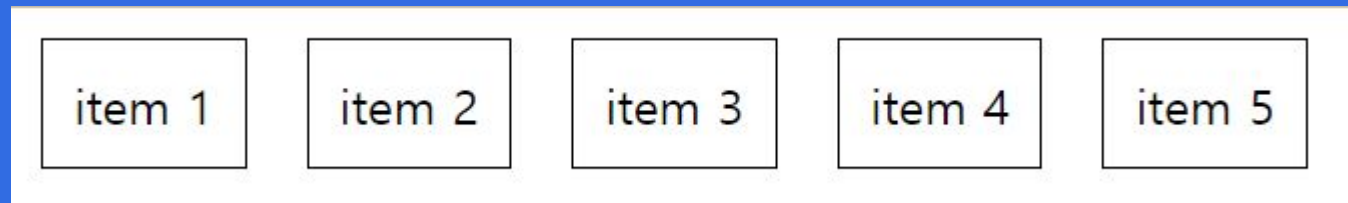
이렇게 레이아웃(display) 속성에 flex를 지정하면,  
해당 태그의 자식태그들을  
우리가 원하는 방식으로 배치할 수 있습니다

```
<div id="container">  
  <div class="item">item 1</div>  
  <div class="item">item 2</div>  
  <div class="item">item 3</div>  
  <div class="item">item 4</div>  
  <div class="item">item 5</div>  
</div>
```

```
#container {  
  display: flex;  
}
```

## 🌼 Flex Box 사용하기 🌼

아까 언급했듯이, 기본 방향은 가로방향(row)이기 때문에  
이대로 코드를 실행하면 다음과 같은 결과가 나와요



## 🌸 Flex Box 사용하기 🌸

배치 방향을 바꾸고 싶다면, **flex-direction** 속성값을 **column** 으로 지정하면 된답니다 (기본값: row)

**align-items** 속성의 기본 값(default 값)은 **stretch** 에요  
한번 **align-items** 속성값 지정 없이도 실행해보세요!

```
#container {  
  display: flex;  
  flex-direction: column;  
  align-items: flex-start;  
}
```

item 1

item 2

item 3

item 4

item 5

## Flex Box

더 자세한 설명은 아주 쉽게 설명 되어있는 1분코딩님의 게시글 추천!  
( <https://studiomeal.com/archives/197> )

## \* TO DO \*

아까 만들었던 페이지에서  
**flex**를 이용하여 자기소개 영역과 시간표 영역을  
가로로 배치해보세요!

## \* TO DO \*

그리고 화면의 가로 크기가 자기소개 영역 + 시간표  
영역의 가로 길이보다 작아질 경우,  
시간표 영역이 아래로 내려가게끔 해주는 flex 속성도  
직접 찾아서 적용해보세요!

## 적용 예시

헤더

### 간단한 자기소개

이름 ❤️ 박민선

mbti ❤️ ISTP

학교 ❤️ 한양대학교 ERICA캠퍼스

학과 ❤️ 소프트웨어학부 컴퓨터전공

취미 ❤️ 나들이 가기

좋아하는 음식 ❤️ 일식, 고기 등

### 나의 시간표

푸터

자기소개 영역은 border, box-shadow 속성들을 사용해서 꾸몄어요  
똑같이 적용해보고 싶으시다면 구글링 해보세요!



헤더

## 간단한 자기소개

이름❤️ 박민선

mbti❤️ ISTP

학교❤️ 한양대학교 ERICA캠퍼스

학과❤️ 소프트웨어학부 컴퓨터전공

취미❤️ 나들이 가기

좋아하는 음식❤️ 일식, 고기 등

푸터

## 나의 시간표

화면의 가로 크기가 줄어들었을 때



헤더

## 간단한 자기소개

이름❤️ 박민선

mbti❤️ ISTP

학교❤️ 한양대학교 ERICA캠퍼스

학과❤️ 소프트웨어학부 컴퓨터전공

취미❤️ 나들이 가기

좋아하는 음식❤️ 일식, 고기 등

## 나의 시간표

푸터

반응형 웹은  
앞에서 살펴보았던 CSS 레이아웃 속성,  
미디어 쿼리 등을 이용해 우리가 직접 구현할 수도 있지만,  
**부트스트랩**이라는 CSS 툴을 사용하여  
손쉽게 구현할 수도 있습니다!

# 부트스트 래

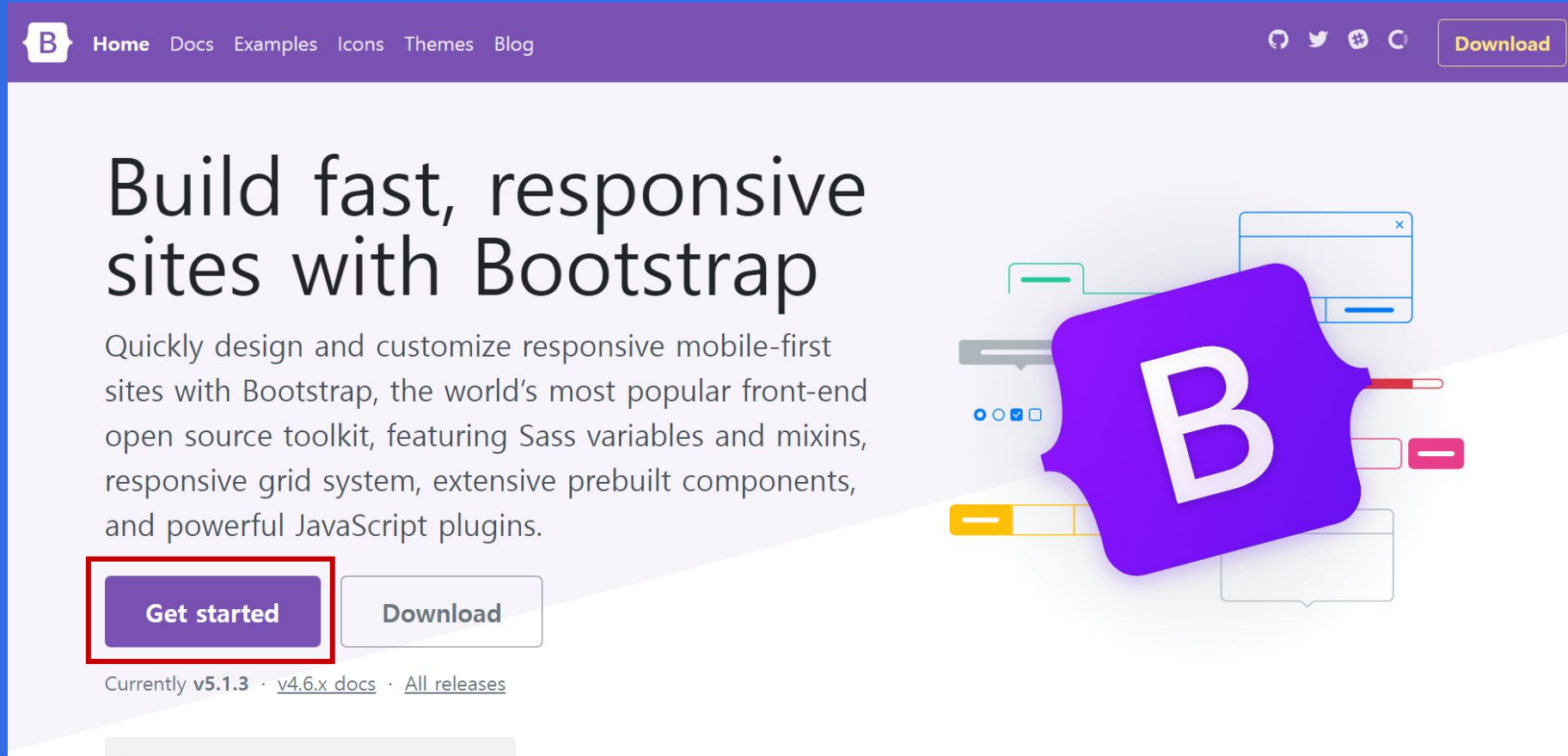


부트스트랩은 **미리 완성된 CSS 디자인**을 개발자들이  
사용할 수 있는 도구예요

부트스트랩 사이트에서  
이미 만들어진 **HTML, CSS, JS 코드**를 복사/붙여넣기만 하여  
손쉽게 사용할 수 있습니다

그러면 이제 부트스트랩을 어떻게 사용하는지,  
사용법을 알려드릴게요 🐾

먼저,  
부트스트랩( <https://getbootstrap.com/> )에 접속하셔서



**Get started** 버튼을 클릭

## 스크롤을 내리다보면 사진과 같이 CSS 섹션이 나타나는데요

### CSS

Copy-paste the stylesheet `<link>` into your `<head>` before all other stylesheets to load our CSS.

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
```

Copy

설명에 나온대로  
여기서 Copy 버튼을 클릭하시고,  
원하는 html 파일의 `<head>` 태그 안에 붙여넣기 해주세요

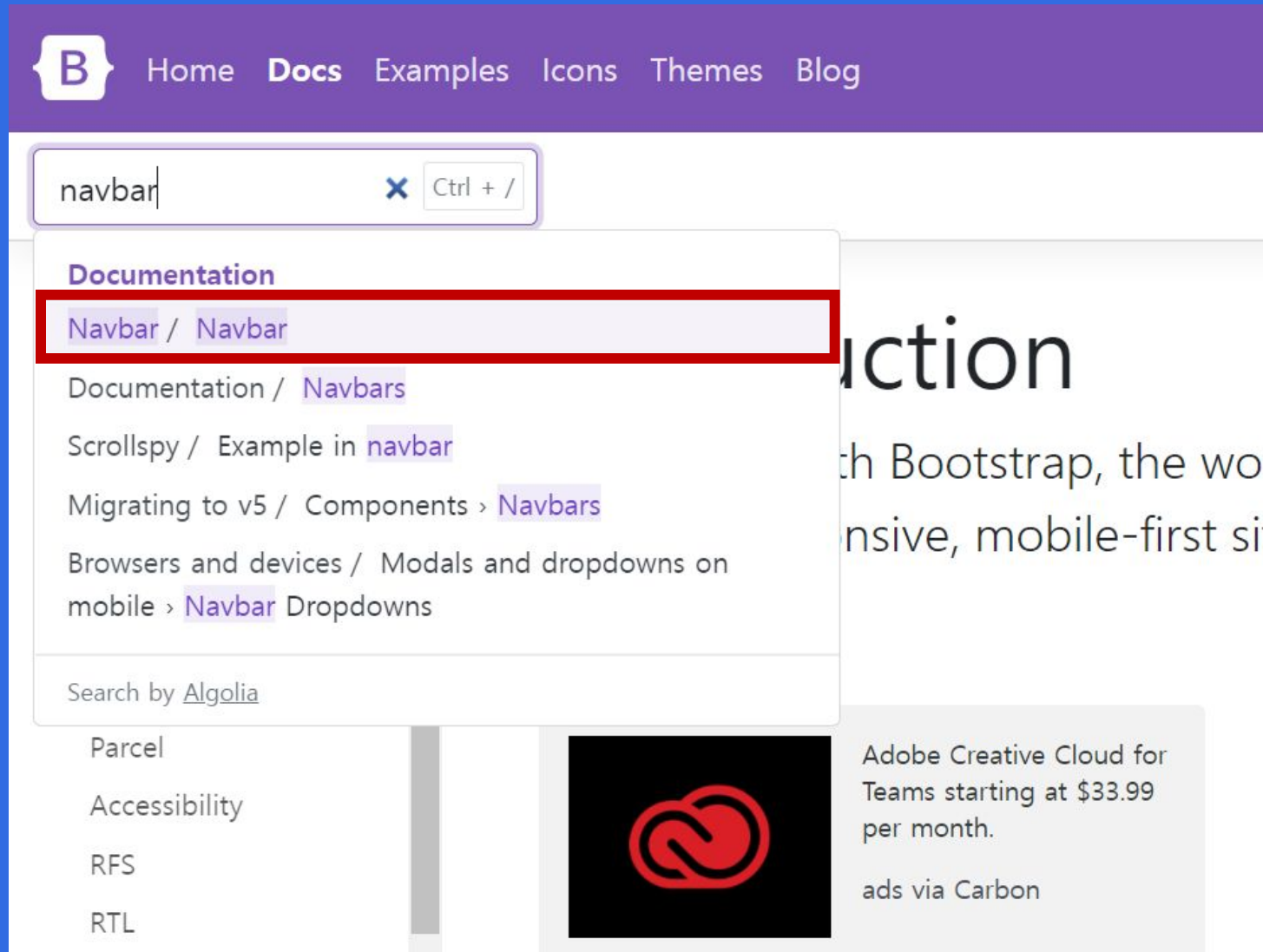
```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.cs
8      <title>Document</title>
9  </head>
10 <body>
11
12 </body>
13 </html>
```



여기까지 하셨다면,  
이제 부트스트랩에서 제공하는 코드를  
마음껏 적용해볼 수 있어요 😄

아까 잠깐 살펴보았던 **navbar**를 한번 적용해볼까요?

왼쪽 상단 검색창에 navbar를 입력해주시고,  
사진에 표시한 부분을 클릭해주세요



# 스크롤을 내리다보면, 다음과 같이 Nav 섹션이 보여요! 😲

## Nav

Navbar navigation links build on our `.nav` options with their own modifier class and require the use of [toggler classes](#) for proper responsive styling. **Navigation in navbars will also grow to occupy as much horizontal space as possible** to keep your navbar contents securely aligned.

Add the `.active` class on `.nav-link` to indicate the current page.

Please note that you should also add the `aria-current` attribute on the active `.nav-link`.

Navbar Home Features Pricing Disabled

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#n
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="#">Home</a>
```

Copy

# 빨갭게 테두리 친 부분은 해당 코드를 적용했을 때 브라우저에 어떻게 나타나는지, 그 예시 화면을 띄워준 것입니다

## Nav

Navbar navigation links build on our `.nav` options with their own modifier class and require the use of [toggler classes](#) for proper responsive styling. **Navigation in navbars will also grow to occupy as much horizontal space as possible** to keep your navbar contents securely aligned.

Add the `.active` class on `.nav-link` to indicate the current page.

Please note that you should also add the `aria-current` attribute on the active `.nav-link`.

Navbar Home Features Pricing Disabled

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#n
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="#">Home</a>
```

Copy

# 우리 html 코드에도 한번 적용시켜볼까요?

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#na
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="#">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Features</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Pricing</a>
        </li>
        <li class="nav-item">
          <a class="nav-link disabled">Disabled</a>
        </li>
      </ul>
    </div>
  </div>
</nav>
```

Copy

코드 블록 오른쪽 상단에 있는 **copy** 버튼을 눌러주세요

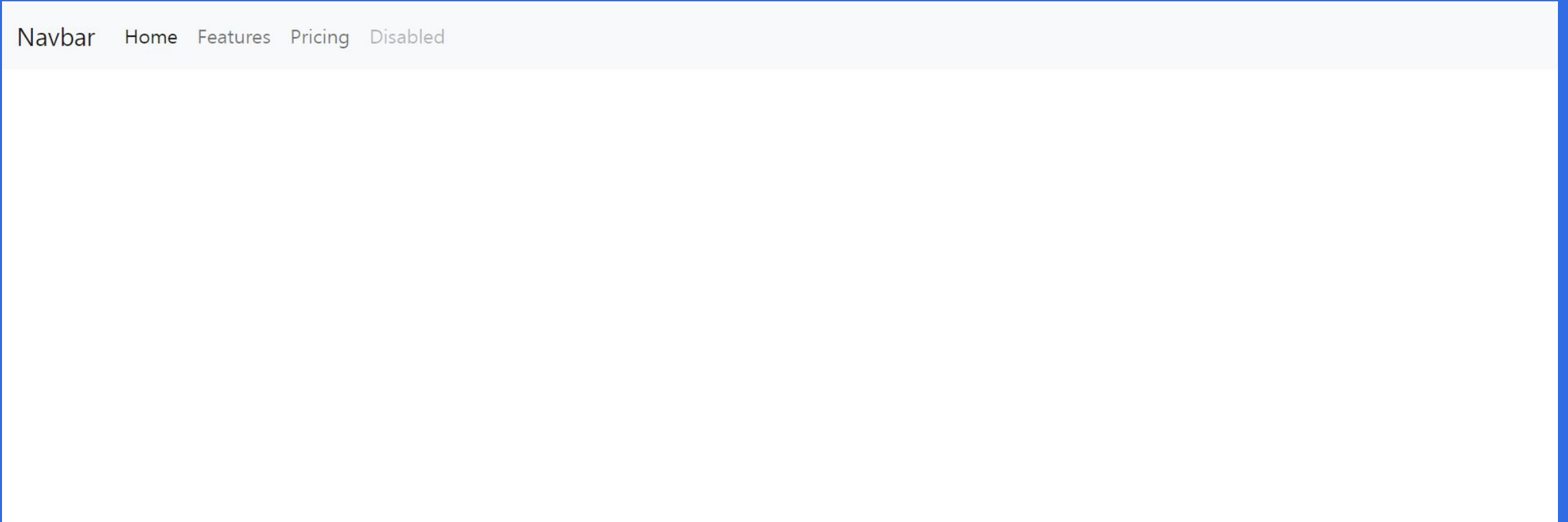


작업중인 html 파일을 열어주세요  
**<body>** 태그 안에서 navbar를 넣고 싶은 부분을 선택하고,  
 앞서 복사한 코드를 붙여넣기 해주세요

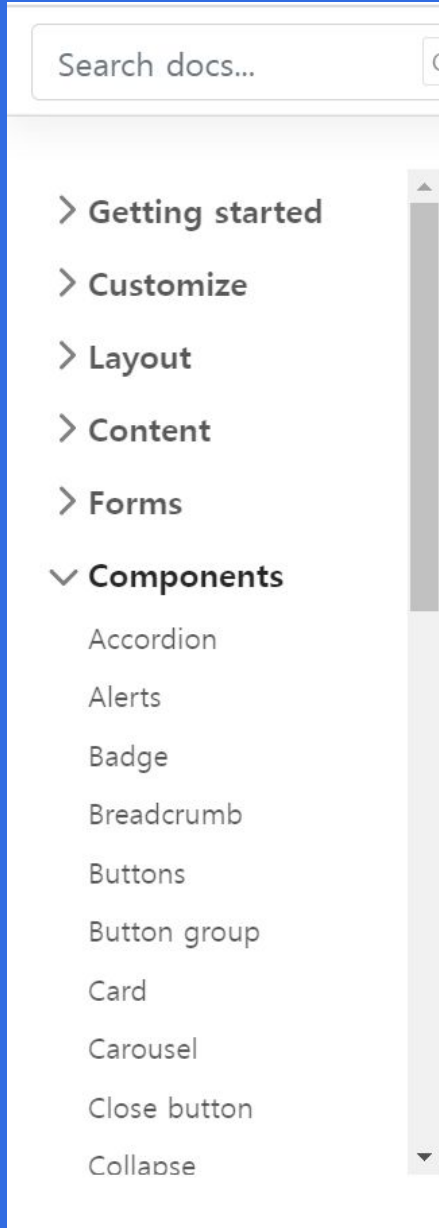
```

9     </head>
10    <body>
11        <nav class="navbar navbar-expand-lg navbar-light bg-light">
12            <div class="container-fluid">
13                <a class="navbar-brand" href="#">Navbar</a>
14                <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-b
15                    <span class="navbar-toggler-icon"></span>
16                </button>
17                <div class="collapse navbar-collapse" id="navbarNav">
18                    <ul class="navbar-nav">
19                        <li class="nav-item">
20                            <a class="nav-link active" aria-current="page" href="#">Home</a>
21                        </li>
22                        <li class="nav-item">
23                            <a class="nav-link" href="#">Features</a>
24                        </li>
25                        <li class="nav-item">
26                            <a class="nav-link" href="#">Pricing</a>
27                        </li>
28                        <li class="nav-item">
29                            <a class="nav-link disabled">Disabled</a>
30                        </li>
31                    </ul>
32                </div>
33            </div>
34        </nav>
35    </body>
36    </html>
    
```

## 파일 저장 후, Live Server로 실행하면



이렇게 부트스트랩에 나와있던 예시와  
똑같이 나오는 것을 볼 수 있습니다 😊



부트스트랩 사이트 좌측을 보시면,  
이렇게 정말 많은 디자인들을 제공하고 있습니다 😲

그때그때 필요한 디자인을 찾아서  
앞서 알려드린 방법대로 복사 / 붙여넣기만 한다면,  
본인이 직접 디자인하지 않고도  
깔끔한 디자인의 웹 사이트를 만들어낼 수 있어요 😊



## \* TO DO \*

이제 첫 번째 TO DO에서 진행한 html 코드를  
부트스트랩, CSS를 사용해서 꾸며볼까요?

참고로 앞서 보여드린 예시에서는  
<nav> 부분을 부트스트랩에서 가져오고,  
나머지 부분은 CSS로 직접 디자인을 입혔답니다!

# 예시

Navbar Home Link Dropdown ▾ Disabled

Search

Search

## 간단한 자기소개

이름❤️ 박민선

mbti❤️ ISTP

학교❤️ 한양대학교 ERICA캠퍼스

학과❤️ 소프트웨어학부 컴퓨터전공

취미❤️ 나들이 가기

좋아하는 음식❤️ 일식, 고기 등

## 나의 시간표

멋쟁이 사자처럼 10기

박민선

참고로 footer는 flex box를 적용했어요!  
예시와 똑같이 해도 괜찮지만  
자율적으로 구글링 하면서 직접 꾸며봐요!

## \* 시간표 넣기 \*

이제 시간표를 넣어 볼 건데요!  
HTML에서는 **<table>**이라는 태그를 사용해서  
표를 만들 수 있어요 😊

완성본을 보면서 **<table>** 태그 사용법을 간단하게만  
알아볼게요

시간	일	월	화	수	목	금	토
9:00-10:00							
10:00-11:00							
11:00-12:00							
12:00-13:00							
13:00-14:00							
14:00-15:00							
15:00-16:00							
16:00-17:00							
17:00-18:00							

시간	일	월	화	수	목	금	토
9:00-10:00							
10:00-11:00							
11:00-12:00							

table 태그의 내부에서 사용되는 태그는 여러가지 존재하는데요, (thead, tbody, tr, th, td .. )  
우선 table은 크게 **thead**와 **tbody** 으로 나눌 수 있습니다

16:00							
16:00-17:00							
17:00-18:00							

시간	일	월	화	수	목	금	토
9:00-10:00							
10:00-11:00							
11:00-							

말 그대로 **thead**는 <table> 에서의 **머리**부분,  
**tbody**는 <table> 에서의 **몸통**부분을 의미해요

16:00							
16:00-17:00							
17:00-18:00							

시간	일	월	화	수	목	금	토
9:00-10:00							
10:00-11:00							
11:00-12:00							
12:00-13:00							
13:00-14:00							
14:00-15:00							
15:00-16:00							
16:00-17:00							
17:00-18:00							

thead

tbody

시간	일	월	화	수	목	금	토
9:00-10:00							
10:00-11:00							
11:00-							

그리고 **<tr>** 태그는 하나의 **행**을 생성하고,  
**<td>** 태그는 하나의 **열**을 생성합니다  
**<th>** 태그는 해당 줄의 **의미가 무엇인지**를 나타낼 때  
주로 사용합니다

16:00							
16:00-17:00							
17:00-18:00							



시간	일	월	화	수	목	금	토
9:00-10:00							
10:00-11:00							
11:00-12:00							
12:00-13:00							
13:00-14:00							
14:00-15:00							
15:00-16:00							
16:00-17:00							
17:00-18:00							

<th>

<tr>

<td>

앞의 내용들을 종합하면  
이런 식으로 코드를 작성하면 되겠죠?

나머지는 여러분이 직접 작성해보세요 😊

```
<table>
  <thead>
    <tr>
      <th>시간</th>
      <th>일</th>
      <th>월</th>
      <th>화</th>
      <th>수</th>
      <th>목</th>
      <th>금</th>
      <th>토</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th>9:00-10:00</th>
      <td></td>
      <td></td>
      <td></td>
      <td></td>
      <td></td>
      <td></td>
      <td></td>
    </tr>
  </tbody>
</table>
```

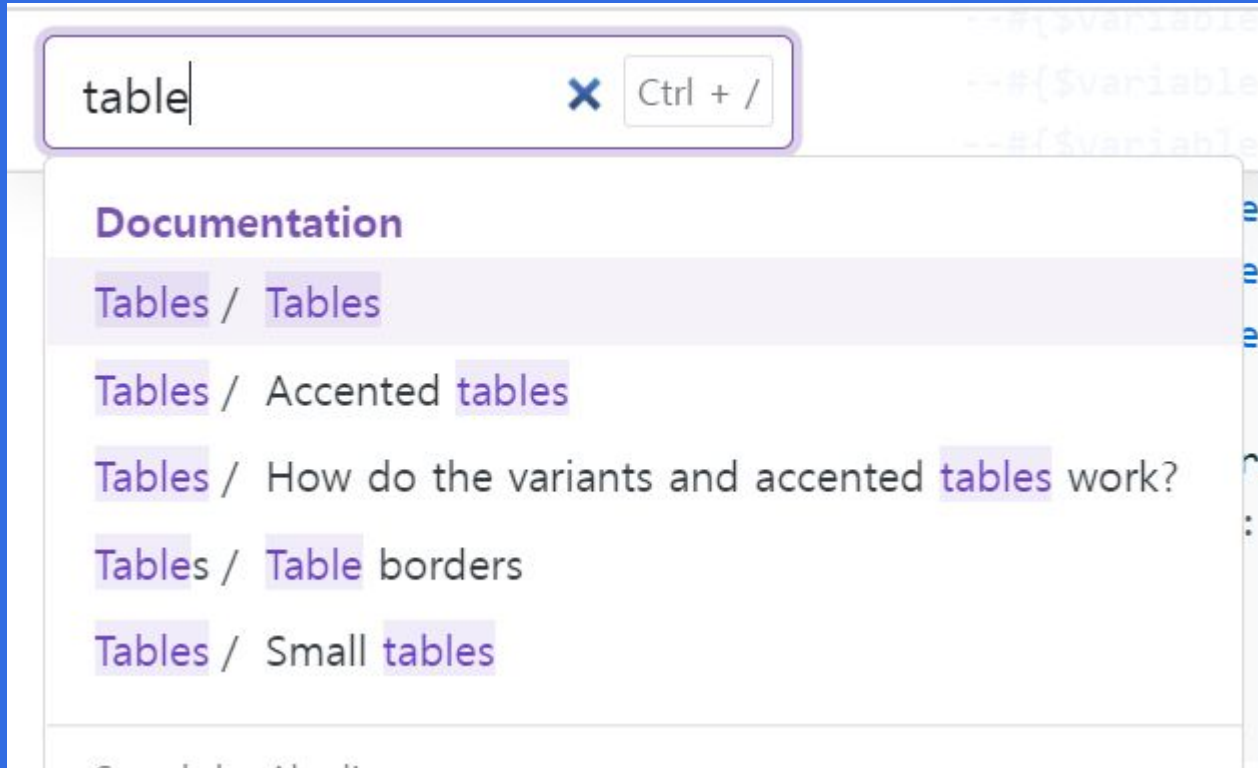
# 예시

HTML 코드를 다 작성하면 다음과 같이 나올 거예요

나의 시간표	
시간	일월화수목금토
9:00-10:00	
10:00-11:00	
11:00-12:00	
12:00-13:00	
13:00-14:00	
14:00-15:00	
15:00-16:00	
16:00-17:00	
17:00-18:00	

# 예시

이제 **부트스트랩**에서 **table**을  
검색해서 시간표를 꾸며 보아요! 😊



## 나의 시간표

시간	일	월	화	수	목	금	토
9:00-10:00							
10:00-11:00							
11:00-12:00							
12:00-13:00							
13:00-14:00							
14:00-15:00							
15:00-16:00							
16:00-17:00							
17:00-18:00							