

Hello,

KDT 웹 개발자 양성 프로젝트

5기!

with



HTML 클래스 속성의 작성법

BEM (Block Element Modifier)

요소_일부분 Underscore(Lodash) 기호로 요소의 일부분을 표시
요소--상태 Hyphen(Dash) 기호로 요소의 상태를 표시



```
<div class="container">  
  <div class="name"></div>  
  <div class="item">  
    <div class="name"></div>  
  </div>  
</div>
```

```
<div class="container">  
  <div class="container__name"></div>  
  <div class="item">  
    <div class="item__name"></div>  
  </div>  
</div>
```

```
<div class="btn primary"></div>  
<div class="btn success"></div>  
<div class="btn error"></div>
```

```
<div class="btn btn--primary"></div>  
<div class="btn btn--success"></div>  
<div class="btn btn--error"></div>
```



HTML 작업하기

```
<header>
  <div class="inner">
    <div class="header__logo">
      <a href="#"><i class="fa-solid fa-leaf"></i></a>
      <a href="#">Inflearn</a>
    </div>

    <ul class="header__menu">
      <li><a href="#">강의</a></li>
      <li><a href="#">로드맵</a></li>
      <li><a href="#">메토링</a></li>
      <li><a href="#">커뮤니티</a></li>
      <li><a href="#">인프런</a></li>
    </ul>

    <div class="header__space"></div>

    <div class="header__search">
      <input type="text" />
      <i class="fa-solid fa-magnifying-glass"></i>
    </div>

    <div class="header__login-join">
      <div class="btn btn--login">로그인</div>
      <div class="btn btn--register">회원가입</div>
    </div>
  </div>
</header>
```



The screenshot shows the Font Awesome website with the Twitter icon selected. The 'Start Using This Icon' button is highlighted with a red arrow. The interface includes navigation links for HTML, React, Vue, and CSS, along with version information (2.0.0 and 3.0.0).



```
body {}
a {}
header {}
header .inner {}
header .inner .header__logo {}
header .inner .header__logo a {}
header .inner .header__logo a i {}
header .inner .header__menu {}
header .inner .header__menu li {}
header .inner .header__menu li a {}
header .inner .header__search {}
header .inner .header__search input {}
header .inner .header__search i {}
header .inner .header__space {}
header .inner .header__participation a:hover {}
header .inner .header__button {}
header .inner .header__button .header__button__login {}
header .inner .header__button .header__button__register {}
```



```
a {
  text-decoration: none;
  color: #4a4a4a;
}

header {
  background-color: skyblue;
  box-shadow: 0 2px 4px 0 rgba(207, 207, 207, 0.5);
  display: flex;
  justify-content: center;
}

.inner {
  background-color: orange;
  width: 1200px;
  height: 64px;
  padding: 0 32px;
  display: flex;
  justify-content: space-between;
  align-items: center;
}
```





로고 파트



Selected family: Indie Flower

Review

Use on the web

To embed a font, copy the code into the `<head>` of your HTML

<link> @import

```
<link rel="preload" href="https://fonts.googleapis.com" type="font-variant-ligatures">
<link rel="preload" href="https://fonts.googleapis.com/css2?family=Indie+Flower&display=swap" rel="stylesheet">
```

CS rules to specify families

```
font-family: 'Indie Flower', cursive;
```

Read our FAQs

```
header .inner .header__logo {  
    height: 32px;  
}  
  
header .inner .header__logo a {  
    font-family: "Indie Flower", cursive;  
    font-size: 32px;  
    color: green;  
}  
  
header .inner .header__logo a i {  
    font-size: 32px;  
    color: green;  
}
```

메뉴 파트



실습, 메뉴 파트 작성하기!

A screenshot of a website header. On the left is the Inlearn logo. To its right is a horizontal menu bar with Korean text: '로그인', '멘토링', '커뮤니티', and '인프런'. Below this is another horizontal bar with the Inlearn logo on the left and menu items '강의', '로드맵', '멘토링', '커뮤니티', and '인프런' on the right. A large red arrow points downwards from the top menu bar towards the second one.



```
header .inner .header__menu {  
  display: flex;  
  font-size: 16px;  
  height: 32px;  
  padding: 0 20px;  
}  
  
header .inner .header__menu li {  
  padding: 8px 16px;  
}  
  
header .inner .header__menu li a:hover {  
  color: #1dc078;  
}
```



검색 파트

실습, 둘보기 아이콘 위치 정하기!

- 둘보기 아이콘을 input 요소 안쪽으로 집어 넣어 봅시다!
- 저는 position: absolute 를 사용해서 구현 했습니다. 하지만 방법은 다양하니 각자의 방법으로 구현해 보세요!



```
header .inner .header__search {  
  height: 36px;  
  margin-right: 20px;  
}  
header .inner .header__search input {  
  box-sizing: border-box;  
  width: 148px;  
  height: 36px;  
  padding: 5px 9px;  
  font-size: 16px;  
  color: gray;  
  background-color: #f6f6f6;  
  border-radius: 3px;  
  border: none;  
}  
header .inner .header__search input:hover {  
  border: 1px solid #c8c8c8;  
}  
header .inner .header__search input:focus {  
  border: 1px solid #c8c8c8;  
  outline: none;  
  box-shadow: inset 0px 0px 0 2px #1dc078;  
}
```



```
header .inner .header__search i {  
  position: absolute;  
  top: 0;  
  bottom: 0;  
  margin: auto;  
  font-size: 16px;  
  height: 16px;  
  right: 9px;  
  /* transform: translateX(-30px); */
```



로그인, 조인 파트



헤더 > 버튼

The screenshot shows a design tool interface with a toolbar at the top. In the center, there is a button labeled "회원가입" (Sign In). To the left of the button, there is a preview area showing the button's appearance. On the right side, there is a detailed properties panel for the button. The properties listed include:

Property	Value
Color	#363636
Font	16px Pretendard, -apple-system, BlinkMa...
Background	#FFFFFF
Margin	0px 8px 0px 0px
Padding	0px 8px 0px 8px
ACCESSIBILITY	
Contrast	Aa 12.08
Name	로그인
Role	button
Keyboard-focusable	(unchecked)

- 인프런 페이지를 참고해서 버튼 기본 틀을
잡아 봅시다!
- 높이: 34px / padding: 8px

```
header .inner .header__login-join .btn {  
  height: 34px;  
  border: 1px solid #dbdbdb;  
  box-sizing: border-box;  
  padding: 8px;  
  border-radius: 3px;  
  cursor: pointer;  
}
```



헤더 > 버튼

- BEM, Modifier 적용!

```
header .inner .header__login-join .btn.btn--login {  
  color: #4a4a4a;  
  background-color: white;  
}  
  
header .inner .header__login-join .btn.btn--register {  
  color: white;  
  background-color: #f77867;  
}
```



디테일 설정

모양이 좀 다르네요?



A screenshot of the Inflearn website header. It shows two different button styles side-by-side. The top row has a dark blue button with white text and a light blue button with dark blue text. The bottom row has a light blue button with dark blue text and a dark blue button with white text. This visual comparison illustrates the difference in appearance when applying BEM and modifier classes.

헤더 > 스페이스

- header__space에 flex-grow 속성 부여!

```
header .inner .header__space {  
  flex-grow: 1;  
}
```

A screenshot of the Inflearn website header demonstrating the effect of adding the 'flex-grow' CSS property to the 'header__space' class. The space between the navigation links and the search bar is significantly larger than in the previous screenshot, making the layout appear more balanced and spacious.

하지만



- 각각 부분에 flex-shrink: 0 값 부여!

```
flex-shrink: 0;
```



페이지 클론!

Clone?



<https://github.com/xenosign/star-bucks>

main 1 branch 0 tags

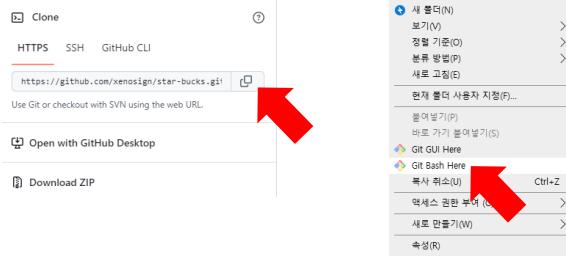
Code

Author	Commit Message	Date	Commits
xenosign	add images	4 minutes ago	2 commits
	images	4 minutes ago	
	.gitattributes	Initial commit	
	favicon.ico	add images	
	favicon.png	add images	

Help people interested in this repository understand your project by adding a README.

Add a README

CLI(Command Line Interface)



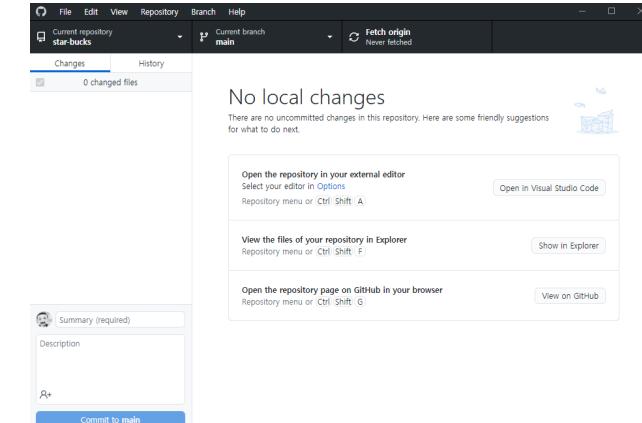
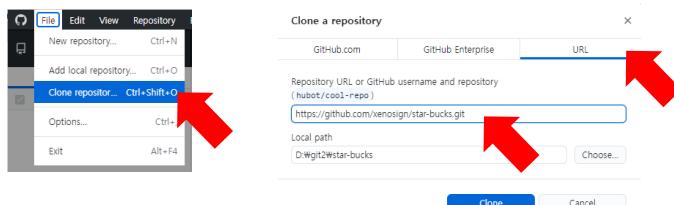
CLI(Command Line Interface)

- 명령어 : [git clone](#) 복사해온 주소
- [git clone https://github.com/xenosign/star-bucks.git](https://github.com/xenosign/star-bucks.git)

```
git clone https://github.com/xenosign/star-bucks.git
Cloning into 'star-bucks'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 10 (delta 0), reused 10 (delta 0), pack-reused 0
Receiving objects: 100% (10/10), 98.49 KiB | 19.70 MiB/s, done.
```



GUI(Graphic User Interface)



Pull!



GUI(Graphic User Interface)

The screenshots show the GitHub Desktop application interface. The top one shows the 'Changes' tab with a red arrow pointing to the 'Fetch origin' button in the toolbar. The bottom one shows the 'Pull origin' button highlighted with a red arrow.



CLI(Command Line Interface)

MINGW64:/d/git/star-bucks

```
tetz@DESKTOP-P7Q40LL MINGW64 /d/git/star-bucks (main)
$ git pull --all
```

- 명령어 : **git fetch origin**(상태 체크, 상태만 로컬에 가져옴, 반영 X)
- **git pull --all**(로컬에 반영 O)

```
tetz@DESKTOP-P7Q40LL MINGW64 /d/git/star-bucks (main)
$ git fetch origin
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 640 bytes 1024 bytes/s, done.
From https://github.com/kenosign/starbucks
  f1af881..cd7ddaa main > origin/main
```

```
tetz@DESKTOP-P7Q40LL MINGW64 /d/git/star-bucks (main)
$ git pull --all
Fetching origin
Updating f1af881..cd7ddaa
Fast-forward
  README.md | 1 +
  1 file changed, 1 insertion(+)
```





페이지 클론!

[https://radiant-maamoul-
80f681.netlify.app/](https://radiant-maamoul-80f681.netlify.app/)



기초 작업 시작!

- Index.html 파일 생성!
- Images / css / js 폴더 각각 생성



HTML



문자 인코딩 설정!

- `<meta charset="UTF-8">`
 - UTF-8 : 초성, 중성, 종성으로 문자를 구분하여 작성(권장)
 - Ex : ㄱ, ㅏ, ㅇ
 - EUC-KR : 하나의 완성 된 글자를 인식
 - Ex : 강



뷰포트 렌더링 방식 설정!

- `<meta name="viewport" content="width=device-width, initial-scale=1.0">`
 - width=device-width : 화면의 가로 너비를 각 디바이스(Device)의 가로 너비와 동일하게 적용
 - initial-scale=1.0 : 화면의 초기 화면 배율(확대 정도)을 설정
 - user-scalable=no : 사용자가 디바이스 화면을 확대(yes)/축소(no)할 수 있는지 설정
 - maximum-scale=1 : 사용자가 화면을 확대할 수 있는 최댓값
 - minimum-scale=1 : 사용자가 화면을 축소할 수 있는 최솟값

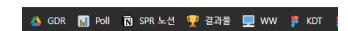


Favicon



Favicon(favorite icon) 설정!

- Favicon이 뭐죠?



- Favorite Icon의 약자입니다!



Favicon(favorite icon) 설정!



- Starbucks 폴더에 favicon.ico 와 favicon.png 적용
- Favicon.ico 는 자동 적용
 - 단, Live Server 로 켜야만 됩니다!
- Favicon.png 는 별도의 link 로 설정
 - 보통 favicon.ico 는 화질이 안 좋기 때문에 상황에 따라 쓸 수 있도록 설정

```
<link rel="icon" href="./favicon.png" />
```

Reset CSS 설정!



- 브라우저에 적용 된 기본 CSS를 초기화
- <https://www.jsdelivr.com/package/npm/reset-css>
- Copy to html 클릭
- 코드에 삽입!

```
<link  
      rel="stylesheet"  
      href="https://cdn.jsdelivr.net/npm/reset-css@5.0.1/reset.min.css"  
/>
```

main CSS 설정!



- 프로젝트 폴더에 css 폴더 생성
- Main.css 파일 생성 후 링크!
- 정상적으로 링크가 되었는지 확인
 - Body 태그에 글자 색을 설정 후
 - Hello, starbucks 출력 후 확인!

상대 경로
절대 경로



절대 경로

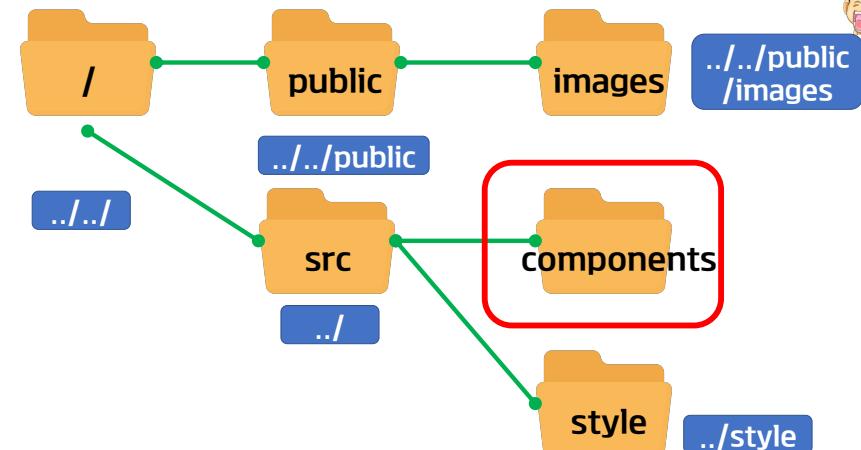
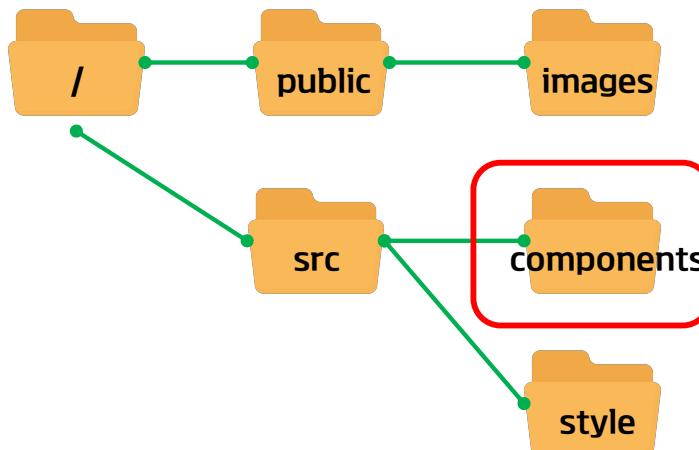
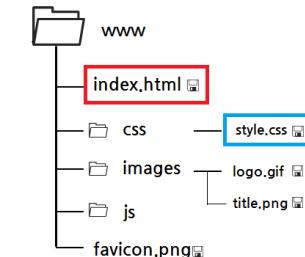


- 최상위 디렉토리부터 해당 파일까지 경유한 모든 경로를 전부 기입하는 방식
- 하나의 파일당 하나의 고유한 절대 경로를 보유
- Ex
 - C:\Users\tetz\Desktop\KDT-5th\수업 자료\정규 수업\9\starbucks-clone\index.html

상대 경로



- 현재 파일이 존재하는 디렉토리를 기준으로 해당 파일까지의 위치를 작성
- 상대 경로 접근 방법
 - 최상위 디렉토리 : /
 - 현재 디렉토리 : ./
 - 상위 디렉토리 : ../



Open Graph (+ Twitter card)



```
<meta property="og:title" content="네이버">
<meta property="og:url" content="https://www.naver.com/">
<meta property="og:image" content="https://s.pstatic.net/static/www/mobile/edit/2016/0705/mobile_212852414260.png">
<meta property="og:description" content="네이버 메인에서 다양한 정보와 유용한 컨텐츠를 만나 보세요">
<meta name="twitter:card" content="summary">
<meta name="twitter:title" content="네이버">
<meta name="twitter:url" content="https://www.naver.com/">
<meta name="twitter:image" content="https://s.pstatic.net/static/www/mobile/edit/2016/0705/mobile_212852414260.png">
<meta name="twitter:description" content="네이버 메인에서 다양한 정보와 유용한 컨텐츠를 만나 보세요">
```



웹 페이지가 SNS로 공유 될 때
우선적으로 활용되는 정보를 지정



Open Graph 설정!

- **og:type** : 페이지의 유형(E.g, website, video.movie)
- **og:site_name** : 속한 사이트의 이름
- **og:title** : 페이지의 이름(제목)
- **og:description** : 페이지의 간단한 설명
- **og:image** : 페이지의 대표 이미지 주소(URL)
- **og:url** : 페이지 주소(URL)



Twitter Card 설정!

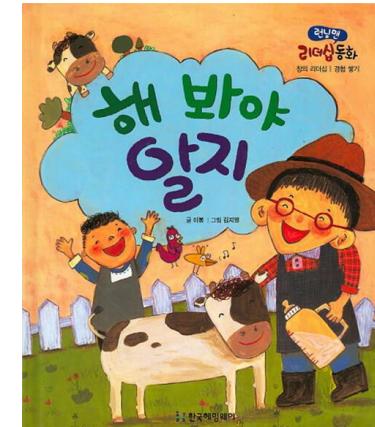
- **twitter:card** : 페이지의 유형(E.g, website, video.movie)
- **twitter:name** : 속한 사이트의 이름
- **twitter:title** : 페이지의 이름(제목)
- **twitter:description** : 페이지의 간단한 설명
- **twitter:image** : 페이지의 대표 이미지 주소(URL)
- **twitter:url** : 페이지 주소(URL)

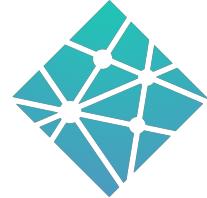


```
<!-- Opengraph -->
<meta property="og:type" content="website" />
<meta property="og:site_name" content="Starbucks" />
<meta property="og:title" content="Starbucks Coffee Korea" />
<meta
  property="og:description"
  content="스타벅스는 세계에서 가장 큰 다국적 커피 전문점으로, 64개국에서 총 23,187개의 매점을 운영하고 있습니다." />
<meta property="og:image" content=".//images/starbucks_seo.jpg" />
<meta
  property="og:url"
  content="https://incandescent-custard-f663b9.netlify.app/" />
```



```
<!-- Twittercard -->
<meta property="twitter:card" content="summary" />
<meta property="twitter:site" content="Starbucks" />
<meta property="twitter:title" content="Starbucks Coffee Korea" />
<meta
  property="twitter:description"
  content="스타벅스는 세계에서 가장 큰 다국적 커피 전문점으로, 64개국에서 총 23,187개의 매점을 운영하고 있습니다." />
<meta property="twitter:image" content=".//images/starbucks_seo.jpg" />
<meta
  property="twitter:url"
  content="https://incandescent-custard-f663b9.netlify.app/" />
```





netlify

<https://www.netlify.com/>



Welcome to Netlify

Sign up with one of the following:

- GitHub
- GitLab
- Bitbucket
- Email

Already have an account? [Log in](#)

Authorize Netlify Auth

Netlify Auth by netlify wants to access your GitHub account

Personal user data

Cancel Authorize netlify

Authorizing will redirect to <https://app.netlify.com>

Not owned or operated by GitHub Created 8 years ago More than 1K GitHub users



Get started

Tell us about yourself

Your responses will help us improve your experience.

I'm using Netlify for

Work Client Personal School

• Required field

I'm a:

Select...

My first project will be:

Select...

Personalize your team space

Make your team your own.

Name your team

Company or team name

People who use Netlify for work often use a company or team name.



Get started

Tell us about yourself
Your responses will help us improve your experience.

I'm using Netlify for
 Work Client Personal School
Required field

I'm a:

My first project will be:

Personalize your team space
Make your team your own.

Name your team

People who use Netlify for personal work often use a project name or their own name.

Set up and continue

Deploy your first project

Import an existing project from a Git repository or pick a starter template.
Not ready yet? [Skip this step for now](#).

Import an existing project
Start from a template
[Import from Git](#) 

...or deploy manually
Drag and drop your site output folder here
Or [browse to upload](#)

Import an existing project from a Git repository

Connect your Git repo, push your code... and that's it! Or [go back](#) to choose a different deploy method.

Step 1 of 3

Connect to Git provider
Choose the Git provider where your site's source code is hosted. When you push to Git, we run your build tool of choice on our servers and deploy the result.

You can [unlock options for self-hosted GitHub/GitLab](#) by upgrading to the Business plan.

GitHub **GitLab** Bitbucket Azure DevOps 

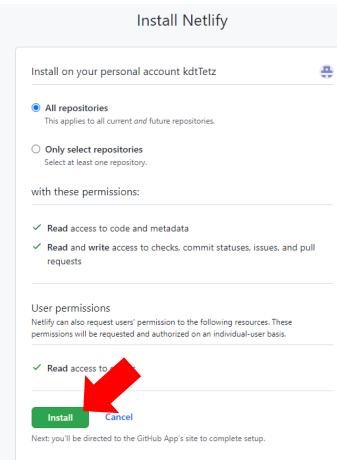
Netlify by Netlify would like permission to:

Verify your GitHub identity (kdtTetz)
Know which resources you can access
Act on your behalf
Resources on your account
Email addresses (read)
View your email addresses

[Learn more about Netlify](#)

[Cancel](#) **Authorize Netlify** 

Authorizing will redirect to
<https://api.netlify.com>



Install Netlify

Install on your personal account kdtetz

All repositories
This applies to all current and future repositories.

Only select repositories
Select at least one repository.

with these permissions:

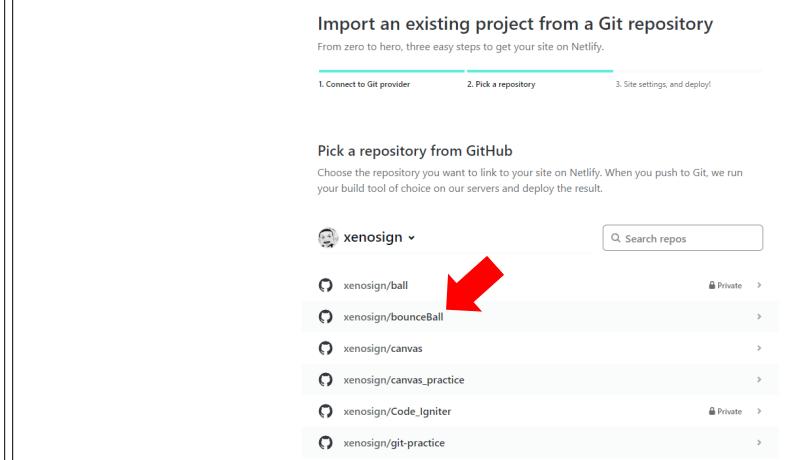
- Read access to code and metadata
- Read and write access to checks, commit statuses, issues, and pull requests

User permissions
Netlify can also request users' permission to the following resources. These permissions will be requested and authorized on an individual-user basis.

- Read access to

Install **Cancel**

Next: you'll be directed to the GitHub App's site to complete setup.



Import an existing project from a Git repository

From zero to hero, three easy steps to get your site on Netlify.

1. Connect to Git provider
2. Pick a repository
3. Site settings, and deploy!

Pick a repository from GitHub

Choose the repository you want to link to your site on Netlify. When you push to Git, we run your build tool of choice on our servers and deploy the result.

xenosign ▾

xenosign/ball

xenosign/bounceBall **arrow pointing to this item**

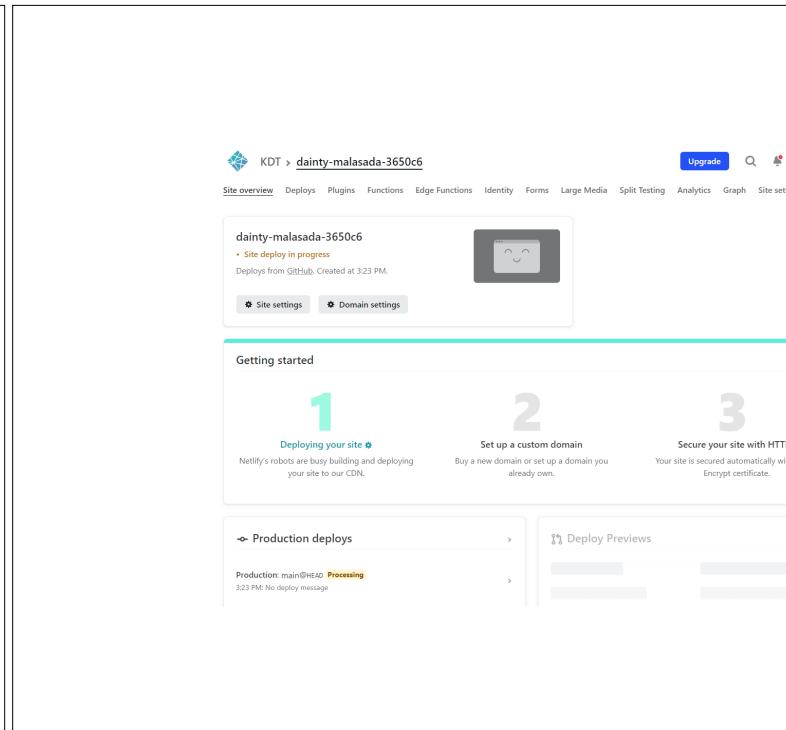
xenosign/canvas

xenosign/canvas_practice

xenosign/Code_Igniter

xenosign/git-practice

xenosign/js-study



Import an existing project from a Git repository

From zero to hero, three easy steps to get your site on Netlify.

1. Connect to Git provider
2. Pick a repository
3. Site settings, and deploy!

Site settings for xenosign/canvas

Get more control over how Netlify builds and deploys your site with these settings.

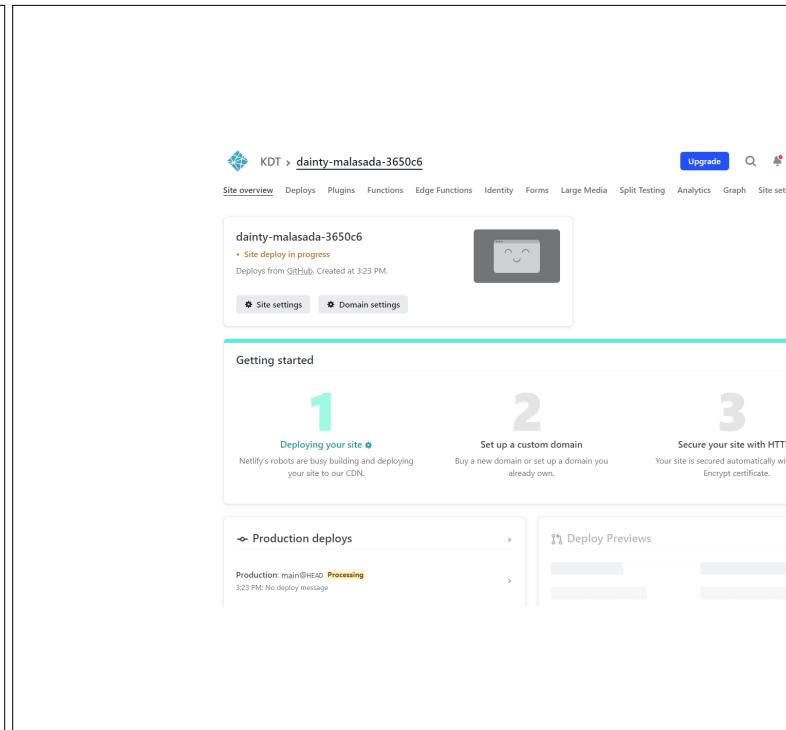
Owner: KDT

Branch to deploy: main

Basic build settings
If you're using a static site generator or build tool, we'll need these settings to build your site.
[Learn more in the docs](#)

Show advanced

Deploy site **arrow pointing to this button**



KDT > dainty-malasada-3650c6

Upgrade

Site overview Deploys Plugins Functions Edge Functions Identity Forms Large Media Split Testing Analytics Graph Site settings

dainty-malasada-3650c6

Site deploy in progress
Deploys from GitHub. Created at 3:23 PM.

Site settings **Domain settings**

Getting started

1 Deploying your site
Netlify's robots are busy building and deploying your site to our CDN.

2 Set up a custom domain
Buy a new domain or set up a domain you already own.

3 Secure your site with HTTPS
Your site is secured automatically with a Let's Encrypt certificate.

Production deploys

Production: main@HEAD Processing
3:23 PM: No deploy message

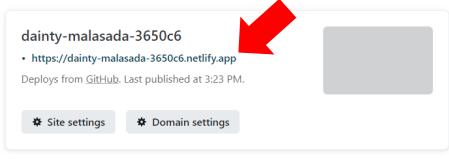
Deploy Previews



Netlify 주의점!

- 메인 페이지는 반드시 **index.html**
- 자 그럼 정상적으로 **Open Graph** 가 작동을 하는지 확인해 볼까요?

tetz 오후 3:28
<https://shiny-strudel-103512.netlify.app/>



Starbucks

Starbucks Coffee Korea

스타벅스는 세계에서 가장 큰 ...

shiny-strudel-103512.netlify.app

Font Family

설정





Hello, Starbucks



Hello, Starbucks!



나눔 고딕을 적용하자!

- <https://fonts.googleapis.com/specimen/Nanum+Gothic?subset=korean&query=nanum#standard-styles>

- 폰트 두께 선택!

나눔 고딕을 적용하자!



- Reset CSS 다음, Main CSS 이전!

```
<!-- Font -->
<link rel="preconnect" href="https://fonts.googleapis.com" />
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />
<link
  href="https://fonts.googleapis.com/css2?family=Nanum+Gothic:wght@400;700;800&display=swap"
  rel="stylesheet"
/>
```

CSS 에도 적용!

```
body {
  color: orange;
  font-size: 100px;
  font-family: "Nanum Gothic", sans-serif;
}
```



Hello, Starbucks
Hello, Starbucks



Hello, Starbucks!
Hello, Starbucks



Icon

가져다 쓰기

이번엔 구글 아이콘을 가져다 써요!



- 이런 아이콘들이 존재!
- 갓 구글 것을 한번 써보죠!
- <https://fonts.google.com/icons>

The screenshot shows the Google Fonts Material Symbols page. At the top, there's a navigation bar with 'Icons' selected. Below it is a search bar and filter options for 'Outline', 'Rounded', and 'Sharp' styles. A section titled 'Introducing Material Symbols' explains the font's features. On the right, there are links for 'Design guide', 'Figma Plugin', 'GitHub repo', 'Apache license', and 'Developer guide'. The main area displays a grid of icons under 'UI actions' such as Search, Home, Menu, Close, Settings, Done, Expand More, Check Circle, Favorite, Add, Delete, Arrow Back, Chevron Right, Star, Logout, Arrow Forward, and Help. To the right of the grid is a 'Customization' sidebar with sliders for 'Fill' (blue), 'Weight' (blue), and 'Grade' (blue).

원하는 스타일로 커스터마이징!



공간 채우기 여부

큰 범주의 두께

작은 범주의 두께(세부 두께 조정)

광학 크기 기본값 설정(크기에 따라 weight, Grade 값 자동 조절)
- 광학적으로 똑같아 보이도록 설정



원하는 아이콘 클릭 후, 가져오기!

The screenshot shows the 'UI actions' section of the Material Icons website. A red arrow points to the search icon, which is highlighted with a blue border. Below the search icon, there is some explanatory text and code snippets for variable and static icon fonts.



크기는 폰트 Size 로 조절!!



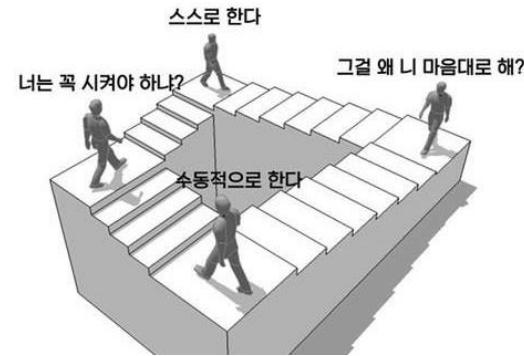
- 왜냐!?
- 폰트니까!

```
<!-- Icons -->
<link
  rel="stylesheet"
  href="https://fonts.googleapis.com/css2?family=Material+Symbols+Outlined:opsz
 ,wght,FILL,GRAD@20..48,100..700,0..1,-50..200"
 />

/* MATERIAL */
.material-symbols-outlined {
  font-variation-settings: "FILL" 0, "wght" 700, "GRAD" 0, "opsz" 48;
}
```

Header 만들기!





<https://www.starbucks.co.kr/index.do>



COMMON CSS



COMMON CSS 설정

- Body 태그 설정
 - 폰트 관련 기본 설정
 - 크기, 두께, 줄 높이는 기본 값 설정
- .inner 클래스 설정
 - 전역에서 사용할 Block 요소 설정
 - 넓이 1100px / 중앙 정렬 사용
- 그 외 태그 관련 설정

```
/* COMMON */
body {
  color: #333;
  font-size: 16px;
  font-weight: 400;
  line-height: 1.4;
  font-family: "Nanum Gothic", sans-serif;
}

.inner {
  width: 1100px;
  margin: 0 auto;
}

a {
  text-decoration: none;
}
```



Header

HTML



```
<header>
  <div class="inner">
    <a href="/" class="logo">
      
    </a>

    <!-- sub menu -->
    <div class="sub-menu">
      <ul class="menu">
        <li>
          <a href="#">Sign In</a>
        </li>
        <li>
          <a href="#">My Starbucks</a>
        </li>
        <li>
          <a href="#">Customer Service & Ideas</a>
        </li>
        <li>
          <a href="#">Find a Store</a>
        </li>
      </ul>
      <div class="search">
        <input type="text" />
        <span class="material-symbols-outlined"> search </span>
      </div>
    </div>
  </div>
```



Header

CSS



기본 CSS 미리 선언하기

```
/* HEADER */
header {}
header .inner {}
header .inner .logo {}
header .sub-menu {}
header .sub-menu ul.menu {}
header .sub-menu ul.menu li {}
header .sub-menu ul.menu li a {}
header .sub-menu ul.menu li a:hover {}
header .sub-menu .search {}
header .sub-menu .search input {}
header .sub-menu .search input:focus {}
header .sub-menu .search .material-symbols-outlined {}
```



전체 Header 스타일 결정

- 넓이는 화면 100% 사용
- 스크롤을 따라오네요!? → fixed
- 배경색: #f6f5ef
- 위쪽 검은 테두리 → 2px solid black
- 아래쪽 회색 테두리 → 1px solid #eae7e2
- Z-index: 1
- 실제 컨텐츠는 .inner 가 가져 갑니다!
 - 높이 값 설정 / Flex 는 아닌것 같죠!?



```
/* HEADER */
header {
  width: 100%;
  position: fixed;
  top: 0;
  background-color: #f6f5ef;
  border-bottom: 1px solid #eae7e2;
  border-top: 2px solid black;
  z-index: 1;
}
```

```
header .inner {
  height: 120px;
}
```



전체 Header 스타일 결정

- 넓이는 화면 100% 사용
- 스크롤을 따라오네요!? → fixed
- 당연히 화면 제일 위에 고정
- 배경색: #f6f5ef
- 위쪽 검은 테두리 → 2px solid black
- 아래쪽 회색 테두리 → 1px solid #eae7e2
- Z-index: 1



로고!



- 로고 위치 조절!
 - Img 태그는 inline 속성입니다!
 - Inline 요소의 수직 정렬은 baseline이 되기 때문에 아래에서 살짝 뜹니다
→ 이미지 태그를 block 속성으로

로고!



- 로고가 inner 밖을 나갈 일이 있을까요?
 - Position: absolute;
- 수직 정렬을 위한 위, 아래 위치 설정
 - Top, bottom: 0;
- 수직 정렬
 - Margin: auto;
- 그런데 안가네요?
 - Margin 계산을 위한 요소의 크기가 없기 때문! → height: 75px;

로고!



- 그런데 이상한 곳으로 가네요!?
- Position: absolute 는 어디를 기준으로 위치를 잡죠?
- 혹시 모르니, .inner 왼쪽에 붙여 두기
 - Left: 0;
- 여기에 right: 0; 을 주면? 가로로도 중앙 정렬이 될까요?

서브 메뉴

실습, 스타벅스 헤더 서브메뉴 만들기!



The screenshot shows the top navigation bar of the Starbucks website. It includes links for 'Sign in', 'My Starbucks', 'Customer Service & Ideas', and 'Find a Store'. Below these is a search bar with a magnifying glass icon. A large red arrow points from the text input field down to the search icon.

실습, 스타벅스 헤더 서브 메뉴 만들기!



- 서브 메뉴

- Top: 6px / right: 50px

- 서브 메뉴 아이템

- 글자색 : #656565 → hover 시, underline

- Padding: 10px 15px

- Font-size: 13px

실습, 스타벅스 헤더 서브 메뉴 만들기!



- 인풋 태그

```
header .sub-menu .search input {  
    width: 36px;  
    height: 34px;  
    padding: 4px 10px;  
    border: 1px solid #ccc;  
    box-sizing: border-box;  
    border-radius: 5px;  
    outline: none;  
    transition: width 0.4s;  
}
```

실습, 스타벅스 헤더 서브 메뉴 만들기!



- 둘보기 아이콘

```
header .sub-menu .search .material-symbols-outlined {  
    font-variation-settings: "FILL" 0, "wght" 700, "GRAD" 0, "opsz" 48;  
}
```

<a> 태그는 클릭을 위한 요소!



- 로고를 잠시 안보이도록 처리!
 - Display: none 또는 visibility: hidden
- <a> 태그 영역을 확실하게 보기위해 백그라운드 색상 설정
- 폰트 크기 설정
- 클릭을 위해서 padding 설정
- 글자 색 변경! / Hover 시 강조 효과
- 밑줄 없애기!

```
header .sub_menu ul.menu li a {  
    background-color: white;  
    font-size: 12px;  
    padding: 10px 15px;  
    color: #656565;  
    text-decoration: none;  
}
```

서브 메뉴 꾸미기



- 가상 요소 선택자를 사용해서 “|” 넣기!
 - ::after
 - Content: “|”
 - Font-size: 13px, color: #e5e5e5
- 뭔가 어색한 점이 느껴지시나요?

Sign in | My Starbucks | Customer Service & Ideas | Find a Store

서브 메뉴 꾸미기



- 세로로 쌓인 아이템을 가로로 펼치고 싶은데 어떻게 하면 될까요?
- 필요 없는 배경을 제거
- 무언가 하나 빠져 있습니다!
 - “|” 넣기
- 여기서 | 를 위한 li 를 따로 선언하는 선택이 맞을까요?

```
header .sub_menu ul.menu {  
    display: flex;  
}
```

서브 메뉴 꾸미기



- 일단 after 요소를 block 으로
- “|” 문자를 넣는 것이 아니라 block 요소의 백 그라운드 색상으로 표기
- 위치가 이상해 지죠!?
 - 포지션을 absolute
 - Top, bottom 값을 주기!
 - 중앙 정렬! → margin: auto;

```
header .sub_menu ul.menu li:not(:last-child)::after {  
    content: "";  
    display: block;  
    width: 1px;  
    height: 12px;  
    background-color: #656565;  
}
```

서브 메뉴 꾸미기

- 그런데도 이상하죠? 왜냐!?

- 포지션이 absolute 일 때, 좌표의 기준은 어디?
- 부모 Li 요소에 position: relative 값 주기

- 그런데 왼쪽에 붙어 있네요!?

- Right: 0;

```
header .sub_menu ul.menu li:not(:last-child)::after {  
    content: "";  
    display: block;  
    width: 1px;  
    height: 12px;  
    background-color: #656565;  
    position: absolute;  
    top: 0;  
    bottom: 0;  
    right: 0;  
    margin: auto;  
  
}  
  
header .sub_menu ul.menu li {  
    position: relative;  
}
```



검색 메뉴 작업!

- Input에 포커스가 가면?

```
outline: none;
```

- 이상한 라인이 생기는게 영 예쁘지 않네요 → outline 제거

- 실제 스타벅스는 검색 메뉴는 포커스를 하면 어찌 되죠?

- 길이가 늘어남 → 190px

- 글자색이 회색

- 폰트 크기도 맞추는 편이 좋겠죠?

- 저 포커스 효과는 맘에 안드니까!

- Input, border를 스타벅스 색(#007042)으로 변경!

- 너무 확확 변하니까 변화 속도를 조절!

```
header .sub_menu .search input:focus {  
    width: 190px;  
    color: #656565;  
    font-size: 12px;  
    border-color: #007042;  
}  
  
transition: width 0.4s;
```



검색 메뉴 작업!

- .search

- 먼저 검색 메뉴의 영역을 확실하게
- 높이 34px

```
header .sub_menu .search {  
    background-color: red;  
    height: 34px;  
}
```

- Input

- 넓이 36px, 높이 34px
- Padding: 4px 10px;
- Border: 1px solid #ccc;
- Box-sizing: border-box;
- Border-radius: 5px;

```
header .sub_menu .search input {  
    width: 36px;  
    height: 34px;  
    padding: 4px 10px;  
    border: 1px solid #ccc;  
    box-sizing: border-box;  
    border-radius: 5px;  
    outline: none;  
}
```



검색 메뉴 옮기기 작업!

- 검색 메뉴를 sub_menu 와 나란히!

- 돋보기를 input 안으로!

- Position: absolute;
- Absolute 를 주면 일단 top, bottom: 0 값 + margin: auto 주기!
- 위치의 기준(부모)이 될 요소에게 position: relative 주기
- 근데 변화가 없네요? → 돋보기의 height 값이 없어서 → 24px
- 그래도 변화가 없네요? 좌우 포지션 속성 값 주기!
- 백그라운드 컬러 제거

```
header .sub_menu {  
    display: flex;  
}  
  
header .sub_menu .search .material-symbols-outlined {  
    position: absolute;  
    height: 24px;  
    top: 0;  
    bottom: 0;  
    right: 3px;  
    margin: auto;  
}
```

화면 배치 작업!

- 전체 서브 메뉴 위치를 변경!
 - Position: absolute; (+ 부모 위치 속성 보기)
 - 오른쪽으로!
 - 위에서 10px 만큼만
- 로고 되돌리기!
- 헤더 배경 색 입히기
 - 배경 : #F6F5EF
 - 아랫줄 : 1px solid #c8c8c8;

```
header .sub_menu {  
    display: flex;  
    position: absolute;  
    right: 0;  
    top: 10px;  
}
```



Sign In | My Starbucks | Customer Service & Ideas | Find a Store



메인 메뉴



메인 메뉴 HTML

```
<ul class="main_menu">
  <li class="item">
    <div class="item__name">COFFEE</div>
  </li>
  <li class="item">
    <div class="item__name">MENU</div>
  </li>
  <li class="item">
    <div class="item__name">STORE</div>
  </li>
  <li class="item">
    <div class="item__name">RESPONSIBILITY</div>
  </li>
  <li class="item">
    <div class="item__name">STARBUCKS REWARDS</div>
  </li>
  <li class="item">
    <div class="item__name">CORPORATE SALES</div>
  </li>
  <li class="item">
    <div class="item__name">WHAT'S NEW</div>
  </li>
</ul>
```



Header CSS



기본 CSS 미리 선언하기

```
/* MAIN MENU */
header .main-menu {}
header .main-menu .item {}
header .main-menu .item .item__name {}
```



실습, 헤더 메인 메뉴 만들기!

The screenshot shows the official Starbucks website's header. It features the Starbucks logo on the left. To the right of the logo is a navigation bar with six items: COFFEE, MENU, STORE, RESPONSIBILITY, STARBUCKS REWARDS, and WHAT'S NEW. The 'MENU' item is highlighted with a green background and white text. At the top right of the header, there are links for Sign In, My Starbucks, Customer Service & Ideas, and Find a Store, along with a search icon.

This screenshot shows the same Starbucks website header as above, but with a different visual style. The 'MENU' item has been styled with a solid green background and white text, while the other menu items remain in their original grey state. The rest of the header elements, including the logo, links, and search icon, appear identical to the first screenshot.



실습, 헤더 메인 메뉴 만들기!



- 메인메뉴 아이템

- Padding: 10px 20px 34px 20px
- Font-size: 13px

- Hover 시

- Background-color: #2c2a29;
- Color: #669900;

메인 메뉴 컨텐츠



The screenshot shows the Starbucks website's header. At the top left is the Starbucks logo. To its right is a dark navigation bar containing the word "COFFEE" in yellow, followed by "MENU", "STORE", "RESPONSIBILITY", "STARBUCKS REWARDS", "CORPORATE SALES", and "WHAT'S NEW". Above the "WHAT'S NEW" link is a search bar with a magnifying glass icon. Below the header, there are five columns of menu items: "커피" (Coffee), "커피 이야기" (Story of Coffee), "스타벅스 리거브" (Starbucks Reserve), "에스프레소 음료" (Espresso Drinks), and "최상의 커피를 즐기는 법" (How to Enjoy the Best Coffee). Each column lists specific items like "스타벅스 원두" (Starbucks Beans) or "도포" (Dopio).



메인 메뉴 컨텐츠, 뼈대를 세우기



- Item__name 의 형제 요소로 item__contents div 요소 추가
- Item__contents 의 자식 요소로 item_contents__menu 추가

```
<li class="item">
  <div class="item__name">WHAT'S NEW</div>
  <div class="item__contents">
    <div class="item__contents_menu">
```

메인 메뉴 컨텐츠, 배치!



- 각각의 영역을 확인하기 위해 background-color 값 지정
- 높이 값을 지정해야 보이므로 각각 200px, 100px 지정
- 어? 이상한데 위치하고 있네요?
- item__contents 의 position 을 지정
 - 기준에 하던 것 처럼, absolute!
 - 어 사라지는데요?
 - 넓이 값을 줘서 위치를 찾아 보죠!

메인 메뉴 컨텐츠, 배치!



- 우리는 화면 넓이 전체를 사용할 것이므로 → width: 100%;
- 오른쪽으로 치우친 것 처럼 보이니 → left: 0;
- 그대로네요!?
- 넓이를 좀 줄여보죠 → width: 10%;
- absolute는 부모 요소를 기준으로 하므로, item의 위치를 따르게 되고, width: 100% 를 해주어도 main-menu 이상 커질 수가 없네요!
- 이럴 땐!?

메인 메뉴 컨텐츠, 배치!



- Fixed 는 뷰포트를 기준으로 하기에, 드디어 찐 100% 적용!
- Top, bottom 값을 지정해 주지 않으면 아이템의 원래 위치를 사용하기 때문에 자연스럽게 들어갔어요!
- 하지만 우리는 헤더의 크기를 알기 때문에 따로 지정을 해줄 수 있습니다!
→ top: 120px;
- 해당 영역에 마우스를 올려 놓으니, WHAT'S NEW 에 hover 효과가 일어나네요!? → 모든 요소가 겹친 상태, 제일 마지막에 올라간 WHAT'S NEW의 contents 가 제일 위에 있기 때문!

```
header .main-menu .item .item__contents {  
  position: fixed;  
  width: 100%;  
  left: 0;  
  top: 120px;  
}
```



메인 메뉴 컨텐츠, Hover 처리



- 메인 메뉴 아이템에 Hover 가 일어나면 Contents 가 보여야 하므로 → 일단 모든 contents 를 안보이게 display: none; 처리
- 아이템에 hover 가 일어나면 다시 보여야 겠죠? → display: block;
- 그럼 컨텐츠를 한번 꾸며 봅시다!
 - 일단 글자를 삽입!
 - 뷰포트가 기준이기 때문에 저~~ 멀리 가있죠?
 - 이럴 때를 위해 사용했었습니다! → .inner

메인 메뉴 컨텐츠, Common Class 처리



- Header 의 직계 .inner 만 선택이 되도록 선택자를 직계로 수정!

```
header > .inner {  
    height: 120px;  
}
```

```
header .main-menu .item .item__contents {  
    position: fixed;  
    width: 100%;  
    left: 0;  
    display: none;  
    top: 120px;  
}  
  
header .main-menu .item:hover .item__contents {  
    display: block;  
}
```



메인 메뉴 컨텐츠, 상단 컨텐츠 작업!



- 시작은 HTML 이겠죠?
- 스타벅스 홈페이지를 봅시다!
 - 기존에 우리가 만들었던 헤더 메뉴 구조 아래에 요소가 붙어 있는게 보이시나요?
 - flex-wrap + 정렬도 보이시죠?
 - 일단 HTML 뼈대를 세워 봅시다!

메인 메뉴 컨텐츠, 상단 컨텐츠 작업!



- .inner 클래스의 경우는 이번에는 ul 만 사용하면 되므로, div → ul
 - 계층 구조는 가능한 단순화 하는게 좋습니다
- 제목은 링크이므로 <a> 태그로 구현
- 의 자식 요소로 를 주기

```
<li class="item">
  <div class="item__name">COFFE</div>
  <div class="item__contents">
    <div class="item__contents__menu">
      <ul class="inner">
        <li>
          <a href="#">커피</a>
        <ul>
          <li>스타벅스 원두</li>
          <li>스타벅스 비아</li>
          <li>스타벅스앳홈 by 캡슐</li>
          <li>나와 어울리는 커피</li>
        </ul>
      </li>
    </ul>
  </div>
</div>
</li>
```



메인 메뉴 컨텐츠, 상단 컨텐츠 CSS



- CSS 기초부터 미리 선언하기!
- 태그 자식으로 , 태그가 중첩 되는 요소 이므로 직계 자손 선택자를 사용 하는게 좋습니다!

```
header .main-menu .item .item__contents .item__contents__menu {}
header .main-menu .item .item__contents .item__contents__menu .inner {}
header .main-menu .item .item__contents .item__contents__menu .inner > li {}
header .main-menu .item .item__contents .item__contents__menu .inner > li > a {}
header .main-menu .item .item__contents .item__contents__menu .inner > li > ul {}
header .main-menu .item .item__contents .item__contents__menu .inner > li > ul > li {}
```



메인 메뉴 컨텐츠, 상단 컨텐츠 CSS



- 먼저 가로 배치를 해야겠죠?
- 넓이는 .inner 1100px 의 공간에 5개의 메뉴가 들어가므로? →
220px 씩 사용이 가능
 - Width: 200px
 - Padding: 20px 20px 0 0;
- 제목 <a> 태그
 - 흰색 / font-size: 15px / bottom 간격 주기

```
header .main-menu .item .item__contents .item__contents__menu {  
    background-color: #2c2a29;  
    padding: 20px 0;  
}  
  
header .main-menu .item .item__contents .item__contents__menu .inner {  
    display: flex;  
    flex-wrap: wrap;  
}  
  
header .main-menu .item .item__contents .item__contents__menu .inner > li {  
    width: 200px;  
    padding-right: 20px 20px 0 0;  
}  
  
header .main-menu .item .item__contents .item__contents__menu .inner > li >  
a {  
    display: block;  
    margin-bottom: 15px;  
    color: white;  
    font-size: 14px;  
}
```

메인 메뉴 컨텐츠, 상단 컨텐츠 CSS



- <a> 태그 hover
 - 밑줄 / 커서 변경
- 하단의 리스트 작업
 - 글자 색상 : #999 / 글자 크기 12px
 - 각각의 태그 상하 간격 설정
 - 태그도 hover 처리

```
Header .main-menu .item .item__contents .item__contents__menu .inner > li > a:hover {  
    text-decoration: underline;  
}  
  
header .main-menu .item .item__contents .item__contents__menu .inner > li > ul {}  
  
header .main-menu .item .item__contents .item__contents__menu .inner > li > ul > li {  
    color: #999;  
    font-size: 12px;  
    margin: 5px 0;  
}  
  
header .main-menu .item .item__contents .item__contents__menu .inner > li > ul > li:hover {  
    cursor: pointer;  
    text-decoration: underline;  
}
```

메인 메뉴 컨텐츠, 나머지 채우기!



메뉴

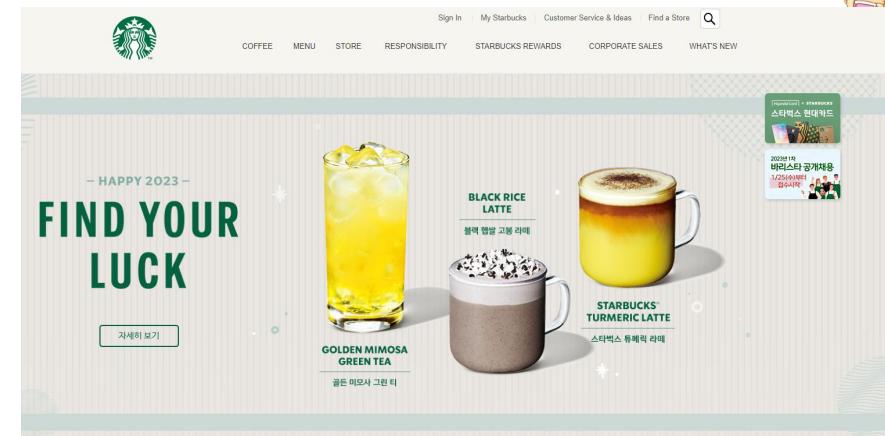
Animation?



```
header .main-menu .item .item__contents {  
  position: fixed;  
  width: 100%;  
  left: 0;  
  top: 120px;  
  transform: scaleY(0);  
  transform-origin: center top 0;  
}  
  
header .main-menu .item:hover .item__contents {  
  transition: 0.5s;  
  transform: scaleY(1);  
}
```



Badges



Badges, 움직임 관찰하기



- 화면 넓이 + 스크롤을 따라오네요? 그렇다면 포지션은?
- 크기도 화면 상태에 따라 변하네요 + 반응형
- 반응형은 다음에 JS 배우고 다시 할 예정입니다!
- 크기 변화도 일단은 컴퓨터 화면을 기준으로 하겠습니다!
- 화면 크기에 따라 변하게 하고 싶으시면 img 크기를 vw 또는 100% 비율로 주시면 됩니다!

Badges, HTML 구축 + CSS 선언



- Header 를 따라오는 구조를 가지고 있으니, header 안에 넣어서 처리하는 방법을 추천드려요
- Badges > badge > img 구조로 고고고!

```
<!-- BADGES -->
<div class="badges">
  <div class="badge">
    
  </div>
  <div class="badge">
    
  </div>
</div>
```

```
/* BADGES */
header .badges {}
header .badges .badge {}
header .badges .badge img {}
```



Badges, CSS 작업

- Header 포지션을 먼저 처리하죠!
 - Position: fixed;
 - 하지만 fixed로 변경하면 .inner의 크기만 남기 때문에 넓이가 이상해집니다 → width: 100%
- Badges는 fixed로 처리해도 되지만, header 속해 있고 header를 기준으로 위치를 잡으면 header를 따라 다니겠죠?
 - Absolute 처리
 - Top: 150px
 - 움직임은 .inner를 기준으로 움직이는군요! 잘 생각해 봅시다!
- 커서 처리!



```
/* BADGES */
header .badges {
  position: absolute;
  top: 150px;
  left: calc(1100px + (100vw - 1100px) / 2);
}

header .badges .badge {
  cursor: pointer;
}

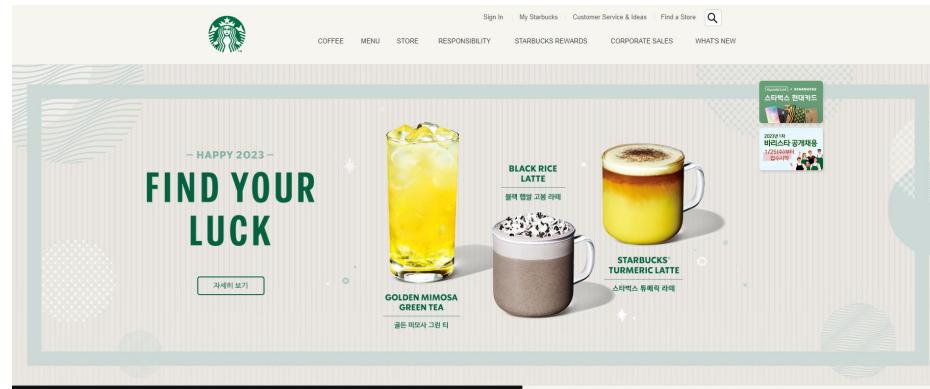
header .badges .badge img {
  width: 140px;
}

@media screen and (max-width: 1400px) {
  header .badges {
    left: calc(100vw - 150px);
  }
}
```



Visual





Badges, HTML 작업

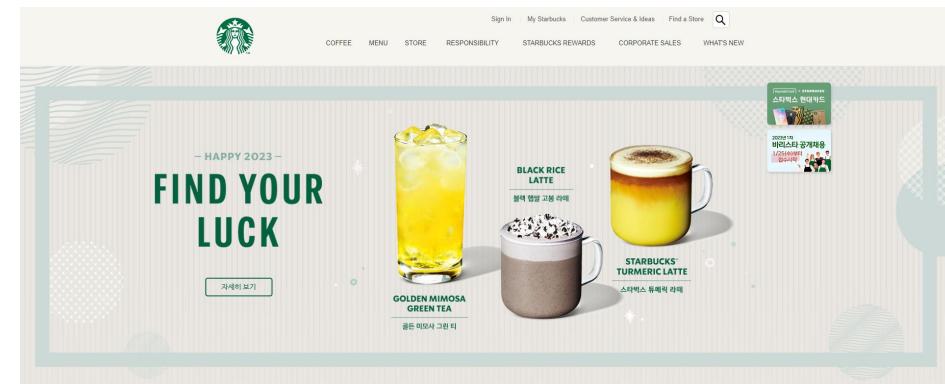


- 중요 아이템을 처리하는 HTML 태그 → <section>
- .inner 클래스 적용
- 타이틀 파트
- 이미지 3개

```
<!-- VISUAL -->
<section class="visual">
  <div class="inner">
    <div class="slogan">
      
      <a href="#" class="btn">자세히 보기</a>
    </div>
    
    
    
  </div>
</section>
```



실습, Visual section 만들기



실습, Visual section 만들기

- Visual 파트 높이 : 646px;
- Background-position : center;
- Slogan 이미지 : 넓이 340px / top 180px / left -130px
- Item-1 : 넓이 290px / top 120px / left 340px
- Item-2 : 넓이 310px / top 208px / left 538px
- Item-3 : 넓이 310px / top 162px / left 762px



실습, Visual section 만들기

- 버튼 스타일

```
.visual .inner .slogan .btn {  
    display: block;  
    width: 130px;  
    text-align: center;  
    font-size: 15px;  
    color: #076443;  
    transition: 0.6s;  
    opacity: 0;  
}  
  
.visual .inner .slogan .btn:hover {  
    background-color: #076443;  
    color: white;  
    font-weight: bold;  
    text-decoration: underline;  
}
```

