

Hello,

KDT 웹 개발자 양성 프로젝트

5기!

with





2018 러시아 월드컵 조추첨식

POT 1

 RUS

 GER

 BRA

 POR

 ARG

 BEL

 POL

 FRA


FIFA WORLD CUP
RUSSIA 2018

GROUP A	GROUP B	GROUP C	GROUP D	GROUP E	GROUP F	GROUP G	GROUP H
A1	B1	C1	D1	E1	F1	G1	H1
A2	B2	C2	D2	E2	F2	G2	H2
A3	B3	C3	D3	E3	F3	G3	H3
A4	B4	C4	D4	E4	F4	G4	H4

SBS



mamihlapinatapai^[1]

마밀라피나타파이

칠레 남부 티에라델푸에고 지역의 "야간"(Yaghan)족 원주민들이 쓰던 명사 단어로, 이하의 특징으로 인해 1993년에 기네스 북에 등재되었다.

본격 세계에서 가장 뜻이 긴 단어. 그와 동시에 세상에서 가장 간명한 단어. 그와 동시에 세상에서 가장 타국어로 번역하기가 난감한 단어.^[2]

mamihlapinatapai : (명) 서로에게 꼭 필요한 것이면서도 자신은 굳이 하고 싶지 않은 어떤 일에 대해서 상대방이 자원하여 해 주기를 바라면서, 두 사람 사이에서 조용하면서도 긴급하게 오가는 미묘한 눈빛.

저기... 2016.01.05 14:59 ([REDACTED])

조장하실분?

↩ ㅋㅋ 2016.01.05 22:10 [REDACTED]

미친ㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋ

↩ 땡 터짐 2016.01.05 23:48 ([REDACTED])

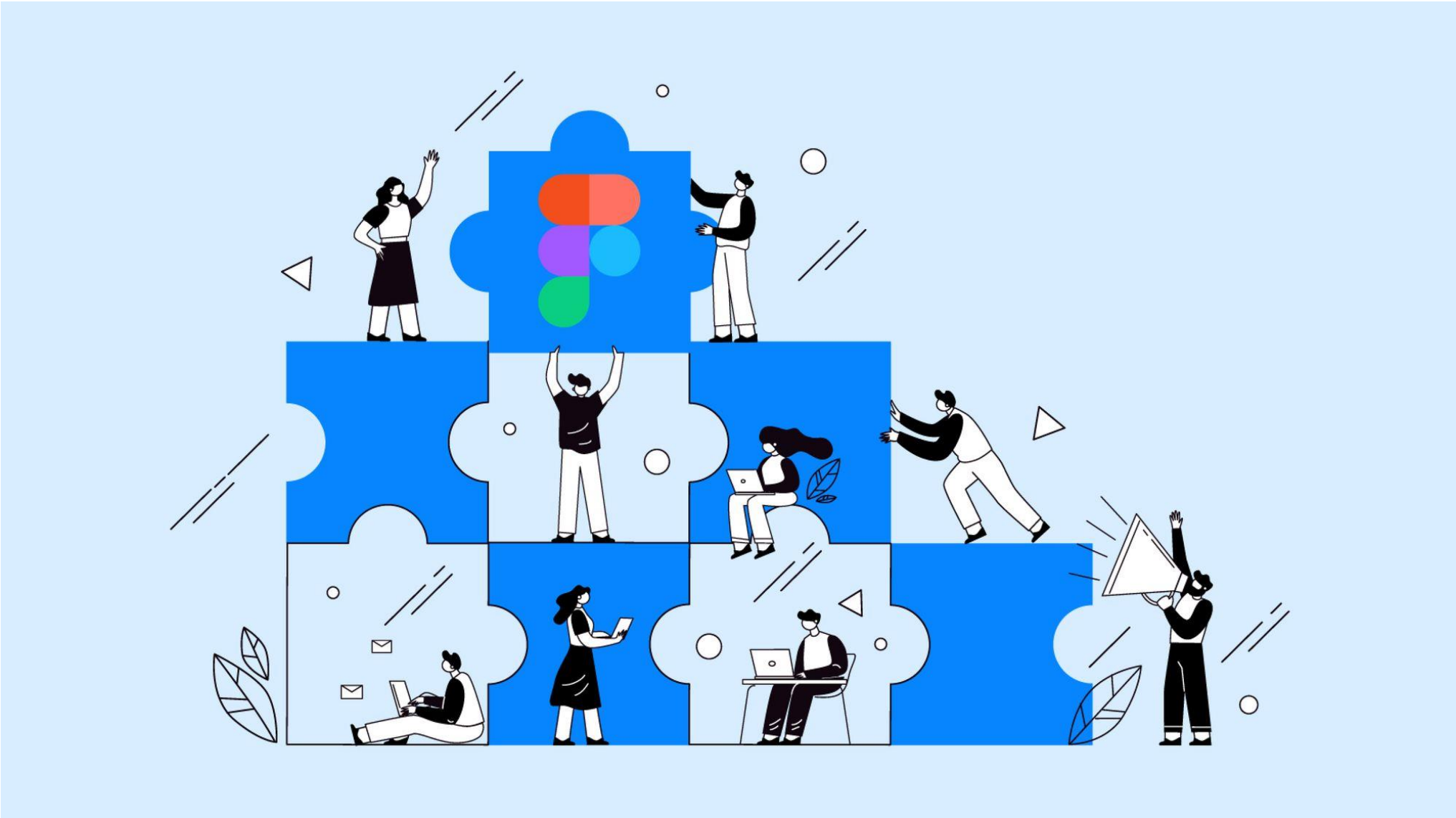
[REDACTED] 첫 댓글이 하드캐리

↩ ㅇㅇ 2016.01.08 04:13 ([REDACTED])

ㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋ



git



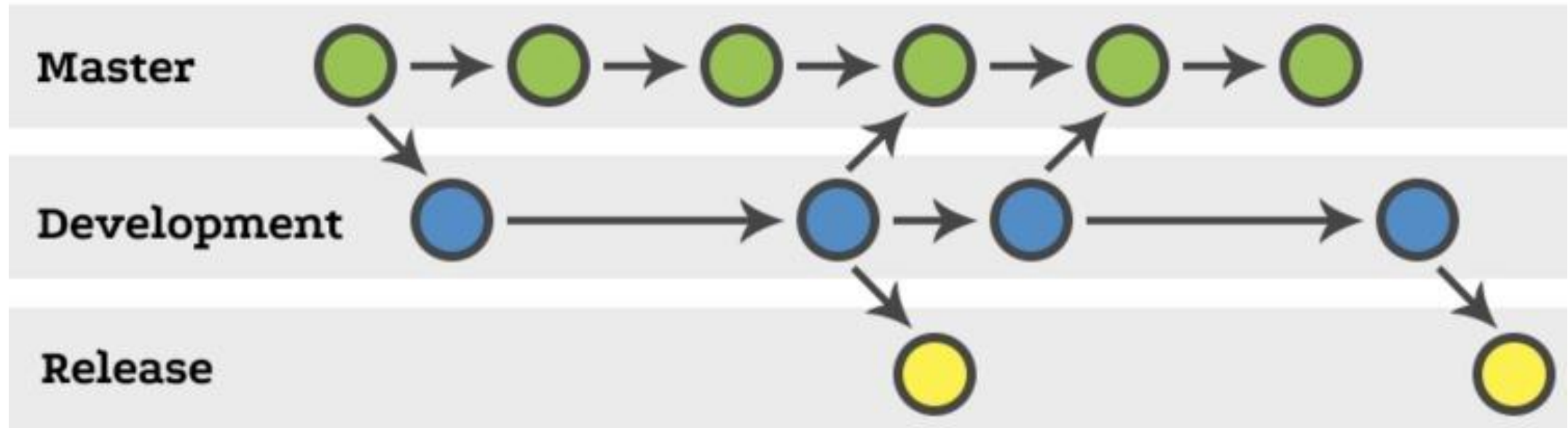


Git branch

차원!



Branch!



Branch!



- 기존 **master**의 내용을 유지하면서 새로운 작업 영역(차원)을 여는 방법입니다!
- **Branch** 를 파서 작업을 하면 해당 작업 사항은 **master** 에 전혀 영향을 끼치지 않습니다!
- **Branch**에서 원하는 기능을 개발 하다가, 완성이 되면 추후에 **Merge** 기능을 통해 **master** 로 합치면 됩니다!



확인 / 추가 / 이동 / 이름 변경 / 삭제

- 확인 : `git branch`
- 추가 : `git branch` 브랜치 이름
- 이동 : `git switch` 브랜치 이름 / `git checkout` 브랜치 이름
- 이름 변경 : `git branch -m` 기존브랜치이름 새로운브랜치이름
- 삭제 : `git branch -d` 브랜치 이름
- 만들면서 이동 : `git checkout -b` 브랜치이름



Merge



이제 내용을 합쳐야 겠죠?

- Merge 는 마스터에서 작업을 합니다!
- `git merge css`
- Css 브랜치의 내용이 master 로 합쳐진 것을 볼 수 있습니다!
- `Git branch`



Conflict



Conflict 란?



- Conflict는 merge를 하려고 할 때, master의 코드와 branch의 코드가 동일한 곳에서 다를 때 발생합니다!
- 다른 곳에서 다른 경우, 또는 삭제된 경우는 사용자가 Edit 한 것으로 판단하여 Conflict가 발생하지 않습니다.
- 동일한 곳에서 코드가 다르면 컴퓨터 입장에서는 어떤 코드가 맞는지 모르므로 해당 부분은 사용자가 처리해 줘야 합니다!

Conflict 상황 해결하기!



```
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
<<<<<< HEAD (Current Change)
  <title>Master</title>
=====
  <title>CSS</title>
>>>>>> css (Incoming Change)
  <link rel="stylesheet" href="./style.css">
</head>
```

- **Current Change**는 master 브랜치의 코드 상황을 보여 줍니다
- **Incoming Change**는 css 브랜치의 코드 상황을 보여 줍니다!
- 즉, 동일한 제목 부분에서 서로 코드가 달라서 conflict 가 발생!
- 이제 사용자는 어떤 코드를 사용할지 정해주면 됩니다!

Conflict 상황 해결하기!



```
tetz@DESKTOP-P7Q4OLL MINGW64 ~/Desktop/git (master|MERGING)
$
```

- 지금은 **merge** 중 이므로 커맨드 창에 **MERGING** 이라고 뜹니다!
- 코드 수정으로 끝이 아니라 **커밋**을 해야 **MERGING** 이 완료 됩니다!
- **Git add . → git commit -m "충돌 해결"**

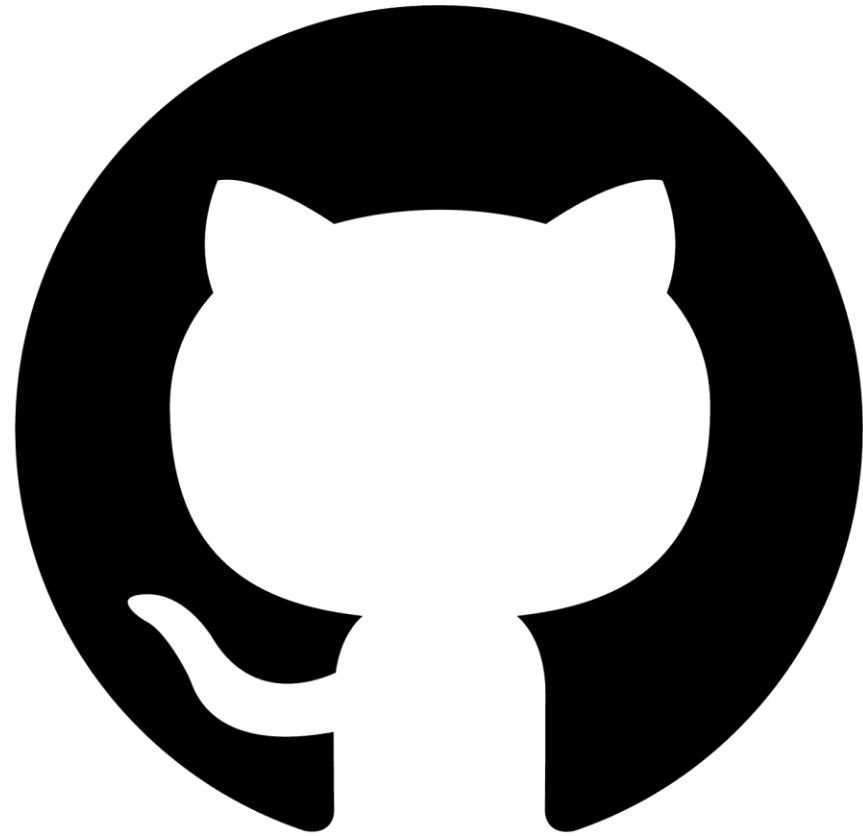
```
tetz@DESKTOP-P7Q4OLL MINGW64 ~/Desktop/git (master)
$
```

- Merge 가 완료되고 **MERGING** 이라는 문구도 사라집니다!

Conflict 상황 해결하기!



- Git branch
- 남은 css 브랜치는 삭제를 해봅시다!
- Git branch -d css
- Git branch




GitHub




online

조장 Part, 팀원 초대!




×


Add a collaborator to portfolio2




Tetsuro Takemoto
Tetz • Invite collaborator




Tetz
TetzSpreatics • Invite collaborator



tetzzz
Invite collaborator



tetziida
Invite collaborator



tetzerakg7
Invite collaborator

Select a collaborator above

Manage access

[Add people](#)☐ Select all Type ▾

☐  **Tetz**
Awaiting TetzSpreatics's response

Pending Invite 

Remove

팀원 Part, 초대 수락!



Signed in as
TetzSpreatics

😊 Set status

Your profile
Your repositories
Your codespaces
Your organizations
Your projects
Your stars
Your gists

Upgrade
Feature preview
Help
Settings

Sign out



Organizations


New organization

 **portfolio2** ⚠️ Invitation expires in 7 days. ●

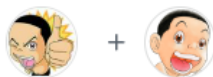
Join

Decline



 **spr-engineering** member and collaborator on 2 repositories

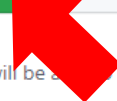
Leave



xenosign invited you to collaborate

Accept invitation

Decline



🔒 Owners of portfolio2 will be able to see:

- Your public profile information
- [Certain activity](#) within this repository
- Country of request origin
- Your access level for this repository
- Your IP address

Is this user sending spam or malicious content?

[Block xenosign](#)

팀원 Part, 다시 Push!



- Git push --all
- 그리고 깃허브로 가봅시다!

css had recent pushes less than a minute ago

Compare & pull request



팀원 Part, 깃헙에서 Merge!



Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base: main ← compare: css ✓ **Able to merge.** These branches can be automatically merged.

css 파일 추가

Write Preview

H B I ≡ <> 🔗 ≡ ≡ ☑ @ ↻ ↶

style.css 파일 추가 및 Reset 작업!

Attach files by dragging & dropping, selecting or pasting them.

Create pull request ▾

Reviewers

xenosign

Assignees

xenosign

Labels

None yet

Projects

None yet

Milestone

No milestone



css 파일 추가 #1

Edit <> Code

Open TetzSpreatics wants to merge 1 commit into main from css

Conversation 0 Commits 1 Checks 0 Files changed 2 +4 -0

TetzSpreatics commented now Collaborator

style.css 파일 추가 및 Reset 작업!

css 파일 추가 5242b11

TetzSpreatics requested a review from xenosign now

TetzSpreatics assigned xenosign now

Reviewers xenosign

Still in progress? Convert to draft

Assignees xenosign

Labels None yet

Projects None yet

Milestone No milestone

Development Successfully merging this pull request may close these issues.

Notifications Customize

Add more commits by pushing to the css branch on xenosign/portfolio2.

Review requested Review has been requested on this pull request. It is not required to merge. Learn more. Show all reviewers

1 pending reviewer

This branch has no conflicts with the base branch Merging can be performed automatically.

Merge pull request You can also open this in GitHub Desktop or view command line instructions.



팀원 Part, 깃헙에서 Merge!



- 현재는 룰을 정하지 않아서 바로 Merge가 가능한 상황입니다!

Add more commits by pushing to the `css` branch on `xenosign/portfolio2`.



Merge pull request #1 from `xenosign/css`

css 파일 추가

Confirm merge

Cancel



팀원 Part, 깃헙에서 Merge!




- Merge가 완료되어 main 브랜치에 style.css 파일이 생성

main 2 branches 0 tags Go to file Add file Code

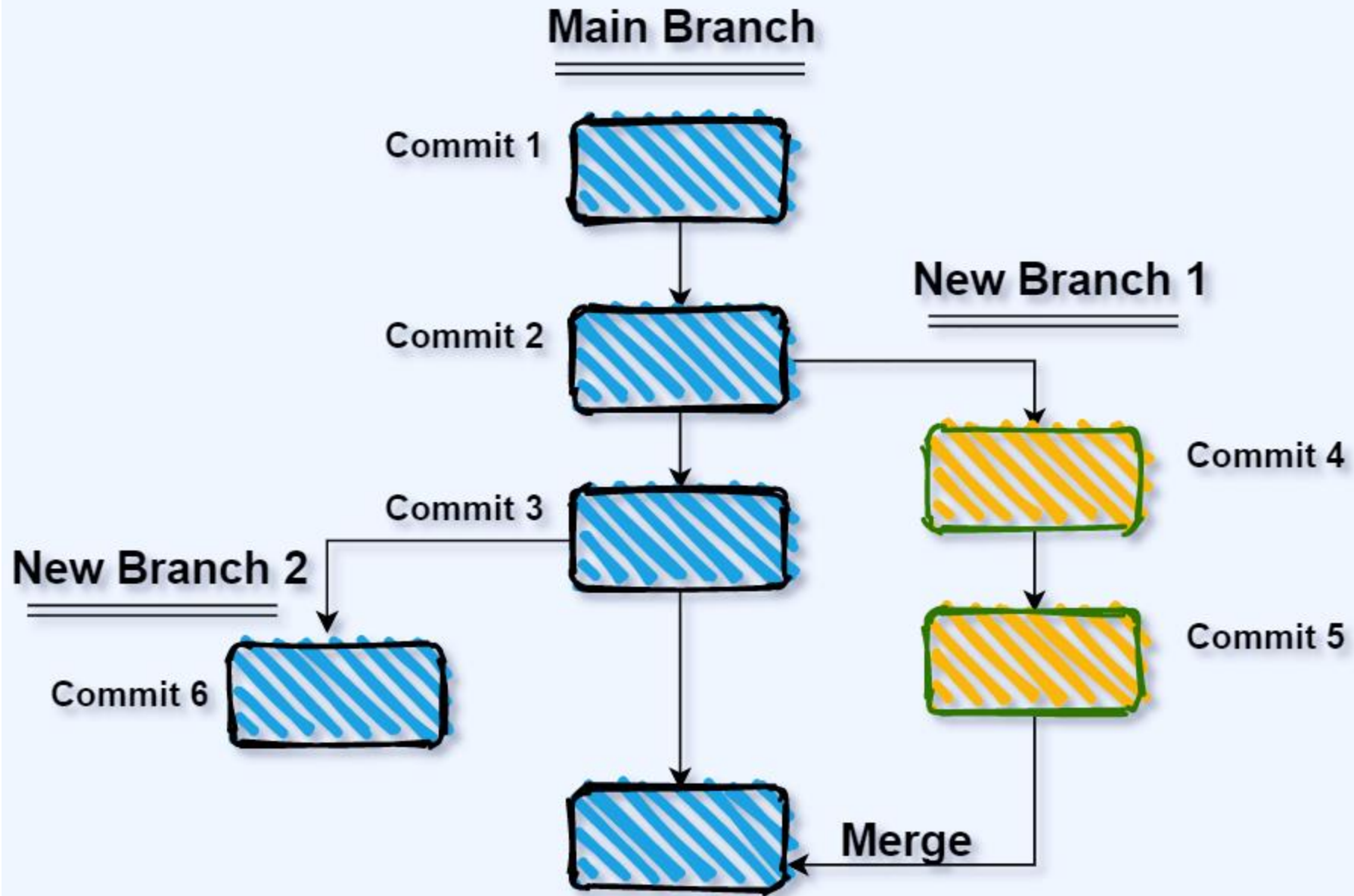
TetzSpreatics Merge pull request #1 from xenosign/css 6e6d956 19 seconds ago 3 commits

index.html	css 파일 추가	17 minutes ago
style.css	css 파일 추가	17 minutes ago

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



- 하지만 아직 우리 Local에는 main의 변경 사항이 반영이 안되어 있습니다!



팀원 Part, main 에서 pull 받기!



- Vscodero 돌아 갑니다!
- Git switch main / git checkout main
- Git pull --all
- 변경 사항 반영 완료!



Main 브랜치 보호하기

조장 Part, 브랜치 보호하기!



☰ xenosign / portfolio2 Public

<> Code

🕒 Issues

🔗 Pull requests

🎬 Actions

📁 Projects

📖 Wiki

🛡️ Security

📊 Insights

⚙️ Settings

⚙️ General

Access

👤 Collaborators

🗨️ Moderation options

Code and automation

🔗 Branches

🏷️ Tags

🎬 Actions

🔗 Webhooks

📁 Environments

📄 Pages

Security

Default branch

The default branch is considered the "base" branch in your repository, against which all pull requests and code commits are automatically made, unless you specify a different branch.

main

Branch protection rules

You haven't protected any of your branches

Define a protected branch rule to disable force pushing, prevent branches from being deleted, and optionally require status checks before merging. [Learn more about protected branches](#)

Add branch protection rule

조장 Part, 브랜치 보호하기!



Branches

Tags

Actions

Webhooks

Environments

Pages

Security

Code security and analysis

Deploy keys

Secrets

Integrations

GitHub apps

Email notifications

Your GitHub Free plan can only enforce rules on its public repository.

Branch name pattern *

main

Protect matching branches

☒ Require a pull request before merging

When enabled, all commits must be made to a non-protected branch that matches this rule.

☒ Require approvals

When enabled, pull requests targeting a matching branch cannot be merged.

Required number of approvals before merging: 1

✓ 1

2

3

4

5

6


- 최소 한명의 Reviewer가 Approve 리뷰를 남겨야만 머지가 가능하도록 설정!



Conflict

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).


base: **main**
←
compare: **conflict**
✖ Can't automatically merge. Don't worry, you can still create the pull request.



Update index.html

Write

Preview

H
B
I
 [List Icon]
 [Code Icon]
 [Link Icon]
 [List Icon]
 [List Icon]
 [Check Icon]
 @
 [Share Icon]
 ↩


Leave a comment

Attach files by dragging & dropping, selecting or pasting them.



Create pull request




 Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

Reviewers



No reviews—at least 1 approving review is required.

Assignees



No one—assign yourself

Labels



None yet

Projects



None yet

Milestone



No milestone


Development





Use [Closing keywords](#) in the description to automatically close issues

Update index.html #3




 Open

TetzSpreatics wants to merge 1 commit into `main` from `conflict` 

 Conversation 0

 Commits 1

 Checks 0

 Files changed 1



TetzSpreatics commented now

Collaborator



No description provided.



Update index.html

6f7bf49



TetzSpreatics requested a review from **xenosign** now



TetzSpreatics assigned **xenosign** now

Add more commits by pushing to the `conflict` branch on **xenosign/portfolio2**.



Review required

[Show all reviewers](#)

At least 1 approving review is required by reviewers with write access. [Learn more](#).



1 pending reviewer



This branch has conflicts that must be resolved

Resolve conflicts

Use the [web editor](#) or the [command line](#) to resolve conflicts.

Conflicting files

index.html





Update index.html #3

Resolving conflicts between `conflict` and `main` and committing changes → `conflict`

1 conflicting file



index.html
index.html

index.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8  <<<<<< conflict
9      <title>conflict</title>
10 |=====
11      <title>test</title>
12 >>>>>> main
13      <link rel="stylesheet" href="style.css">
14      <script src="main.js"></script>
15  </head>
16
17  <body>
18
19  </body>
20
21  </html>
22
```





Update index.html #3

Resolving conflicts between `conflict` and `main` and committing changes → `conflict`

1 conflicting file

index.html

index.html

index.html

1 <!DOCTYPE html>

2 <html lang="en">

3

4 <head>

5 <meta charset="UTF-8">

6 <meta http-equiv="X-UA-Compatible" content="IE=edge">

7 <meta name="viewport" content="width=device-width, initial-scale=1.0">

8 <title>conflict</title>

9 <link rel="stylesheet" href="style.css">

10 <script src="main.js"></script>

11 </head>

12

13 <body>

14

15 </body>

16

17 </html>

18

1 conflict

Prev ^

Next v

⚙

Mark as resolved

Commit merge

✓ Resolved



Sass

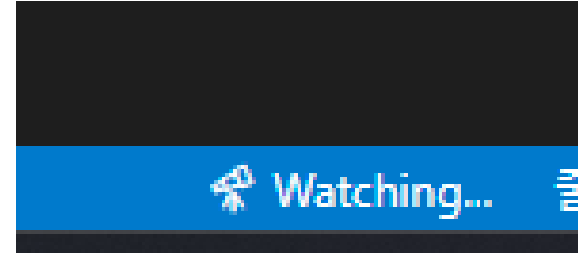
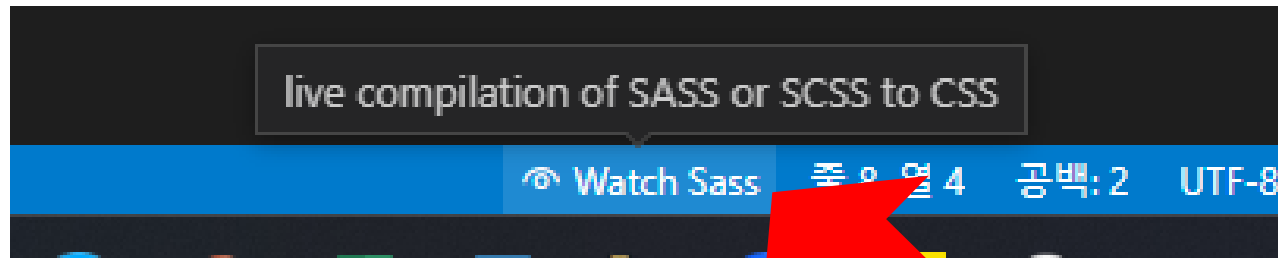
SCSS 사용 방법 1



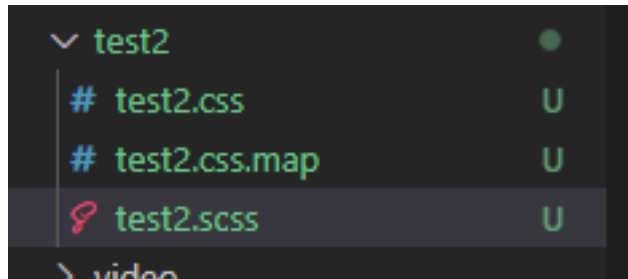
- `npx sass --watch SCSS파일.scss CSS파일.css`
- Npx 님! Sass 를 사용해서 scss 파일을 css 파일로 변경해 주세요!
- + 그리고 scss 파일이 보고 있다가(watch) 파일이 변경되면 CSS에 바로 적용해 주세요~!

```
lhs@DESKTOP-86MUCGC MINGW64 /d/git/movie_main/test (main)
$ npx sass --watch test.scss test.css
Compiled test.scss to test.css.
Sass is watching for changes. Press Ctrl-C to stop.
```

SCSS 사용 방법 2



- Watch Sass 클릭하면 하면 알아서 변환이 일어 납니다!
- Watching 상태면 변화가 저장되면 알아서 변환 됩니다!





Syntactically Awesome Style Sheets



Import



다른 scss 파일 가져오기, **import**

- @import "path/filename"
 - @import "scss/style.scss"
 - CSS 파일도 가능
- 방금 reset 처럼 확장자 없이 파일 이름만 치면
 - 해당파일이름.scss or sass
 - 또는 _해당파일이름.scss or sass 파일이 있으면 가져옴
 - 단, 같은 파일 이름이 2개 이상 존재하면 에러 발생

```
/* RESET */  
@import "reset";  
  
/* COMMON */  
@import "nav";  
@import "section";
```



Nesting

& 선택자 활용



- 치환 특성을 활용해서
- 공통 특성을 가진 modifier
- 생산 가능

```
.btn {  
  width: 100px;  
  height: 30px;  
  background-color: orange;  
  border-radius: 10px;  
  line-height: 30px;  
  
  &--small {  
    width: 50px;  
  }  
  
  &--large {  
    width: 200px;  
  }  
  
  &--red {  
    background-color: red;  
  }  
}
```



변수!(\$)



주요 색상을 변수에 지정하고 사용!

- \$ 를 사용해서 변수를 지정합니다!
- 보통 최 상단에 표기를 합니다!
- BEM 과 같이 쓰면 편하겠죠?

```
$body-bg: #eee;
$wrap-border: #ddd;
$wrap__item: #fff;
$wrap__item_h2: #333;
$wrap__item_p: #777;
$btn-bg: orange;
$btn-font: white;

body {
  background-color: $body-bg;
}
```



@mixin



기본값 설정도 가능합니다!

- 실수로 에러가 발생하는 것을 막기 위해 기본값 설정 가능

```
@mixin button($width: 100%, $bg-color: grey, $font-color: white) {  
  display: block;  
  width: $width;  
  height: 30px;  
  border-radius: 15px;  
  background-color: $bg-color;  
  color: $font-color;  
  text-align: center;  
  font: bold 16px/30px "arial";  
  margin-top: 20px;  
}
```



@for

@for 문을 통해 해결해 봅시다!



```
@for $i from 1 to 5 {  
  &:nth-child(#{ $i }) {  
    transition-delay: 0.2s * $i;  
  }  
}
```

- 또, 변경을 요청할 지 모르니 0.2s 도 변수로 지정하여 사용해 봅시다!

```
$interval: 0.3s;  
  
@for $i from 1 to 5 {  
  &:nth-child(#{ $i }) {  
    transition-delay: $interval * $i;  
  }  
}
```



@each

하는 김에 아까전의 @for 문도 처리



- \$index 값으로 처리가 가능하므로 하나로 묶어서 처리 합시다!

```
@each $el in $btn-bg-colors {  
  $index: index($btn-bg-colors, $el);  
  
  &:nth-child(#{ $index }) {  
    transition-delay: $interval * $index;  
    a {  
      @include button($bg-color: $el);  
    }  
  }  
}
```



@if

@if 문



- Item 의 백그라운드 색상에 따라 h2, p 태그의 글자 색상을 변경!

```
h2 {  
  font: bold 30px/1 "arial";  
  margin-bottom: 20px;  
  @if $wrap__item == #fff {  
    color: $el;  
  } @else {  
    color: white;  
  }  
}
```

```
p {  
  font: normal 16px/1.5 "arial";  
  @if $wrap__item == #fff {  
    color: $el;  
  } @else {  
    color: white;  
  }  
}
```



지금 시작합니다



git



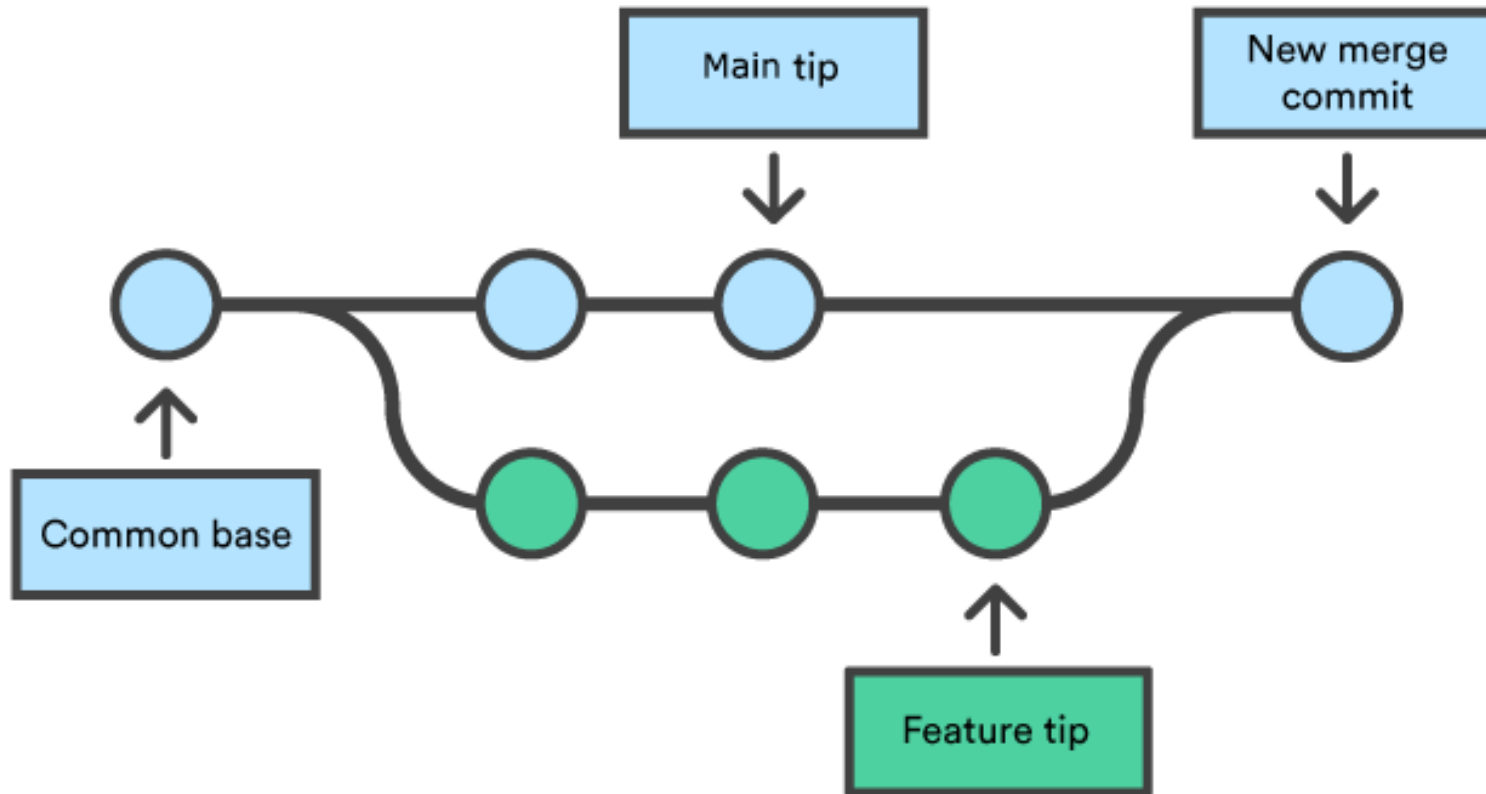
GitHub



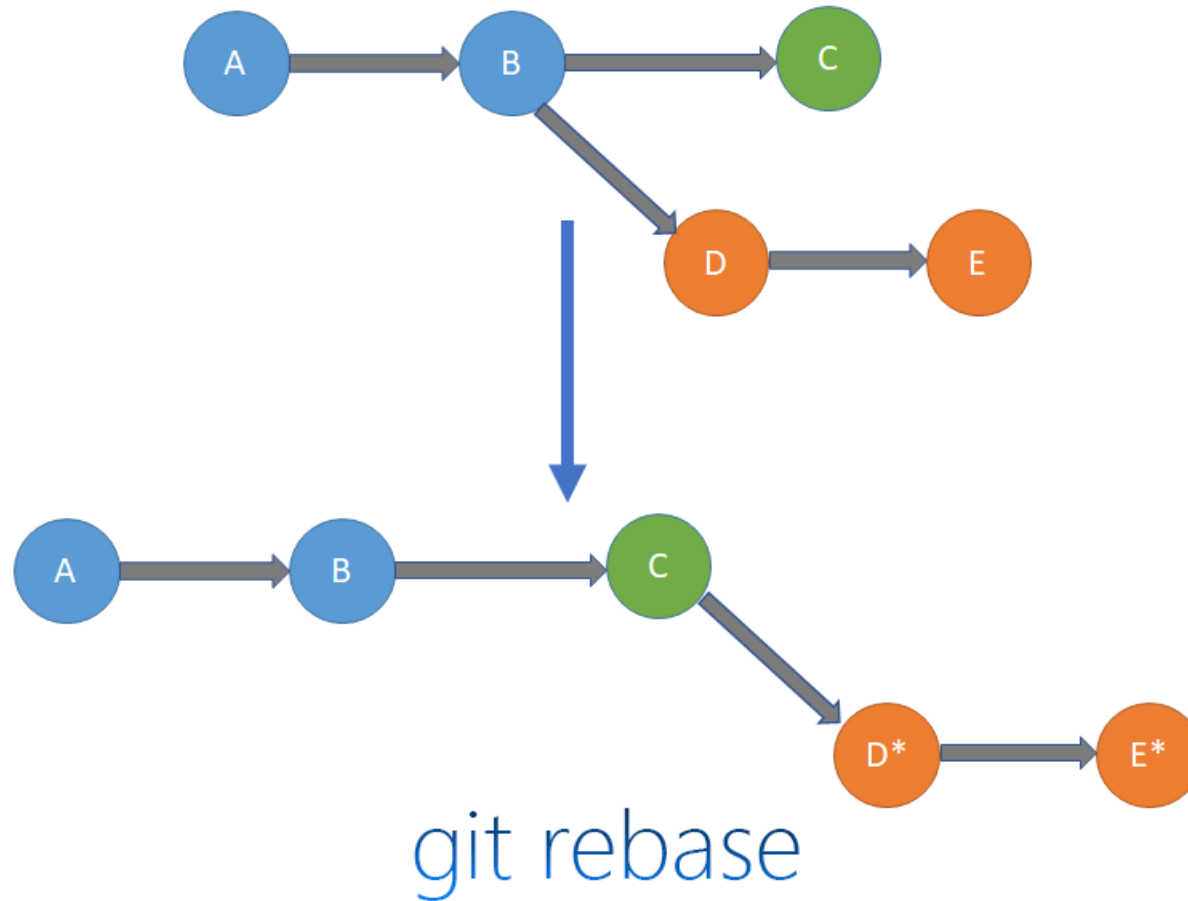
BRANCH를
유지하고
싶어요

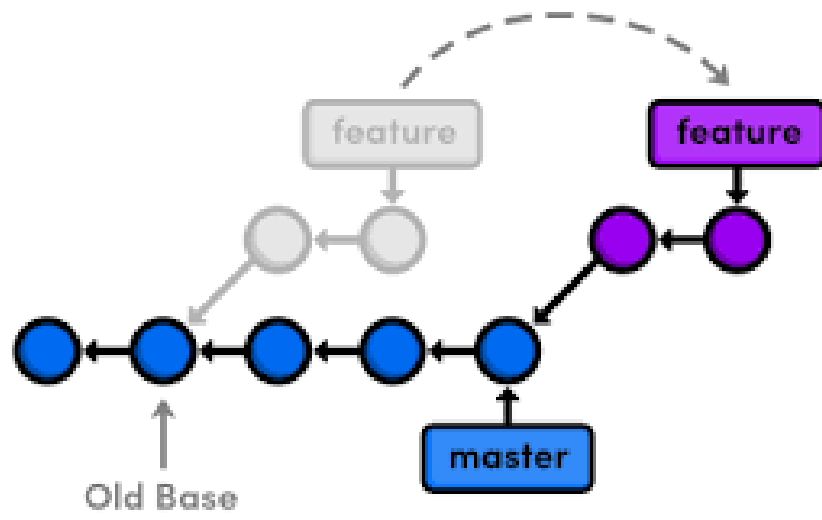


Git MERGING



Git REBASE





그래 이거야!

11-58









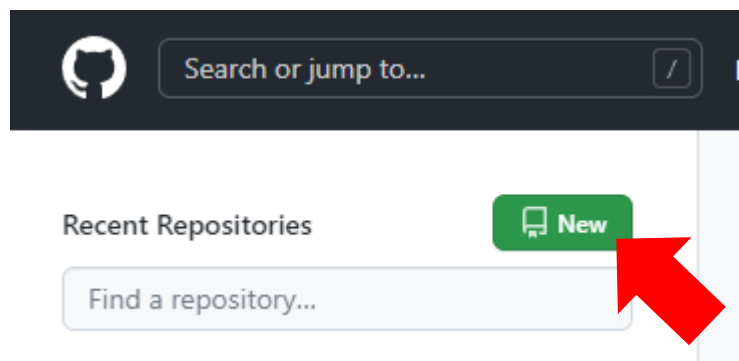
조장 Part



온라인 협업을 시작해 봅시다!



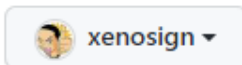
- 조장 분은 깃허브에 협업 연습을 할 Repo를 생성해 봅시다!



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner *



Repository name *



Great repository names are short and memorable. Need inspiration? How about [scaling-journey?](#)

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can commit.




Private

You choose who can see and commit to this repository.

온라인 협업을 시작해 봅시다!



Quick setup — if you've done this kind of thing before

 Set up in Desktop

or

HTTPS

SSH

<https://github.com/xenosign/portfolio2.git>



Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# portfolio2" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/xenosign/portfolio2.git
git push -u origin main
```



...or push an existing repository from the command line

```
git remote add origin https://github.com/xenosign/portfolio2.git
git branch -M main
git push -u origin main
```

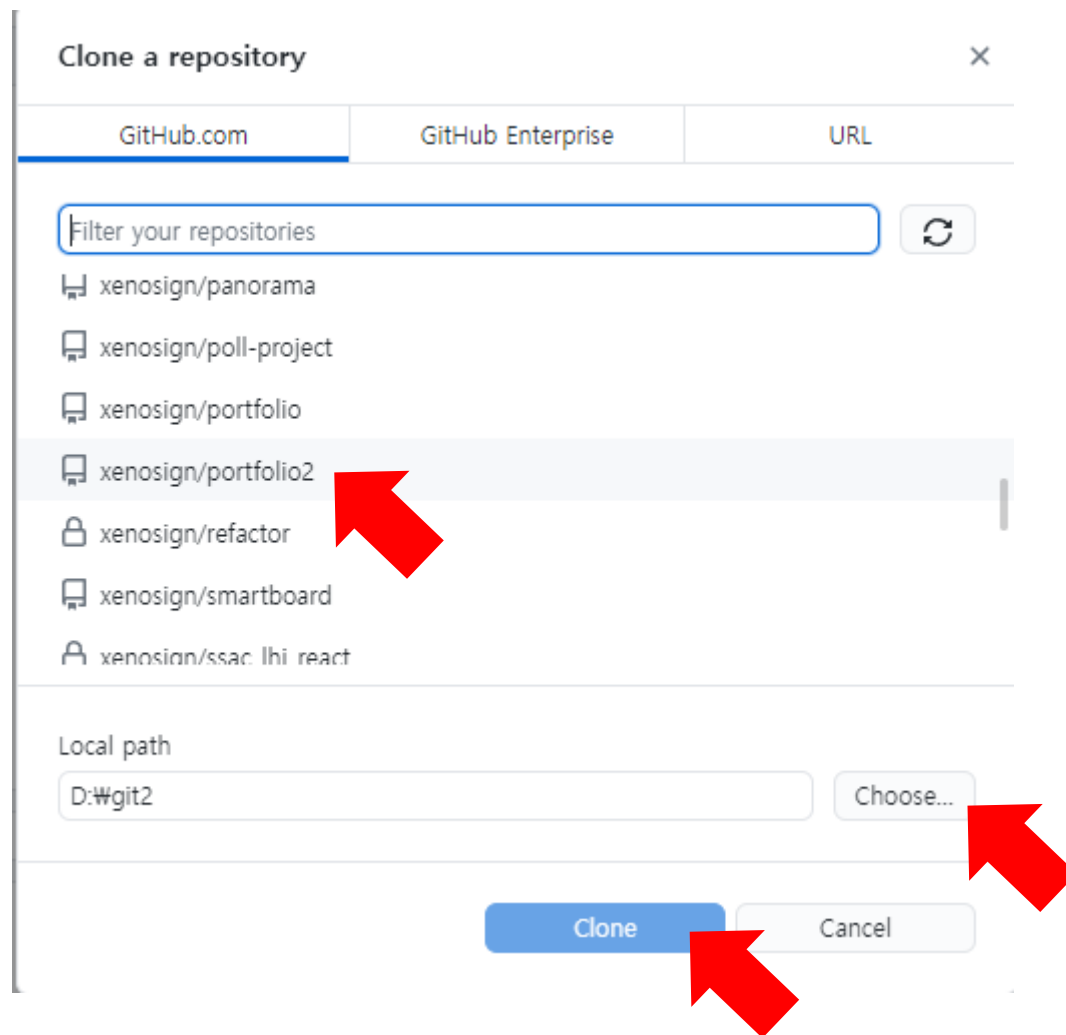
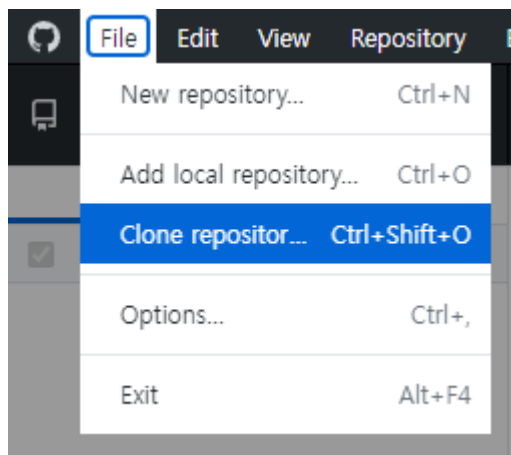


...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)

온라인 협업을 시작해 봅시다!



새로운 깃 저장소!



- Index.html 파일 생성 → 기본 코드 생성
- 커밋
- **Git push origin main / git push --all**
- 또는 Github Desktop으로 처리!

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.



Publish your branch

The current branch (main) hasn't been published to the remote yet. By publishing it to GitHub you can share it, open a pull request, and collaborate with others.

Always available in the toolbar or **Ctrl** **P**

Publish branch



조장 Part, 팀원 초대!



xensign / portfolio2 Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Actions

Webhooks

Environments

Pages

Manage access


You haven't invited any collaborators yet

Add people


A large red arrow pointing from the right towards the 'Settings' tab in the top navigation bar.A large red arrow pointing from the right towards the 'Collaborators' option in the left sidebar.A large red arrow pointing from the right towards the 'Add people' button in the 'Manage access' section.

조장 Part, 팀원 초대!




×


Add a collaborator to portfolio2




Tetsuro Takemoto
Tetz • Invite collaborator




Tetz
TetzSpreatics • Invite collaborator



tetzzz
Invite collaborator



tetziida
Invite collaborator



tetzerakg7
Invite collaborator

Select a collaborator above

Manage access

[Add people](#)☐ Select all Type ▾

☐  **Tetz**
Awaiting TetzSpreatics's response

Pending Invite 

Remove



팀원 Part

팀원 Part, 초대 수락!



Signed in as
TetzSpreatics

😊 Set status

Your profile
Your repositories
Your codespaces
Your organizations
Your projects
Your stars
Your gists

Upgrade
Feature preview
Help
Settings

Sign out



Organizations

New organization


 **portfolio2** ⚠ Invitation expires in 7 days. ●

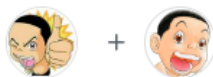
Join

Decline



Leave

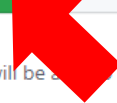
 **spr-engineering** member and collaborator on 2 repositories



xenosign invited you to collaborate

Accept invitation

Decline



🔒 Owners of portfolio2 will be able to see:

- Your public profile information
- [Certain activity](#) within this repository
- Country of request origin
- Your access level for this repository
- Your IP address

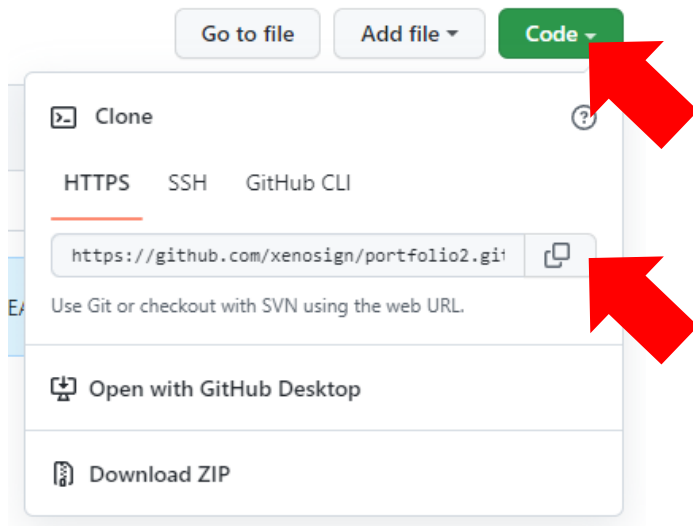
Is this user sending spam or malicious content?

[Block xenosign](#)

팀원들은, 해당 repo를 clone 합시다!



- 조장이 만든 git 주소를 받으세요! 해당 주소로 접속!



- **Git clone** 복사한 주소



전원 Part

전원 Part, 클론한 Repo에서!




- 새로운 브랜치 생성!
- Git branch 본인이름 이니셜!
- Git checkout -b lhs
- Style-이니셜.css 파일 생성!
- 아무 코드 추가 후 커밋!
- Git push origin 브랜치명

전원 Part, 다시 Push!



- `Git push --all`
- 그리고 깃허브로 가봅시다!

 css had recent pushes less than a minute ago

Compare & pull request




전원 Part, 깃헙에서 Merge!



Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base: main < <compare: css >> <div>✓ Able to merge. These branches can be automatically merged.</div></div><div><div><div><div><div>css 파일 추가</div><div><div>Write</div><div>Preview</div><div><div>H</div><div>B</div><div>I</div><div>≡</div><div><></div><div>🔗</div><div>≡</div><div>≡</div><div>☑</div><div>@</div><div>🗨</div><div>↩</div></div><div><div>style.css 파일 추가 및 Reset 작업!</div><div>Attach files by dragging & dropping, selecting or pasting them.</div><div><div>Create pull request</div><div>▼</div></div></div><div><div>Reviewers</div><div><div>xenosign</div><div>Assignees</div><div><div>xenosign</div><div>Labels</div><div>None yet</div><div>Projects</div><div>None yet</div><div>Milestone</div><div>No milestone</div></div></div></div></div>



css 파일 추가 #1

Edit <> Code

Open TetzSpreatics wants to merge 1 commit into main from css

Conversation 0 Commits 1 Checks 0 Files changed 2

+4 -0



TetzSpreatics commented now

Collaborator

style.css 파일 추가 및 Reset 작업!



css 파일 추가

5242b11



TetzSpreatics requested a review from xenosign now



TetzSpreatics assigned xenosign now

Add more commits by pushing to the css branch on xenosign/portfolio2.



Review requested

Show all reviews

Review has been requested on this pull request. It is not required to merge. [Learn more.](#)



1 pending reviewer



This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request



You can also open this in [GitHub Desktop](#) or view [command line instructions](#).



Reviewers



xenosign



Still in progress? Convert to draft

Assignees



xenosign

Labels



None yet

Projects



None yet

Milestone



No milestone

Development



Successfully merging this pull request may close these issues.

None yet

Notifications

Customize

전원 Part, 깃헙에서 Merge!



- 현재는 룰을 정하지 않아서 바로 Merge가 가능한 상황입니다!

Add more commits by pushing to the `css` branch on `xenosign/portfolio2`.



Merge pull request #1 from `xenosign/css`

css 파일 추가

Confirm merge

Cancel



전원 Part, 깃헙에서 Merge!




- Merge가 완료되어 main 브랜치에 style.css 파일이 생성

main 2 branches 0 tags Go to file Add file Code

TetzSpreatics Merge pull request #1 from xenosign/css 6e6d956 19 seconds ago 3 commits

index.html	css 파일 추가	17 minutes ago
style.css	css 파일 추가	17 minutes ago

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



- 하지만 아직 우리 Local에는 main의 변경 사항이 반영이 안되어 있습니다!



전원 Part

전원 Part, pull 받기!



- Vscode로 돌아 갑니다!
- Git switch main / git checkout main
- Git pull
- 변경 사항 반영 완료!



전원 Part, rebase 로 메인 코드 땡겨오기



- 자신이 작업 중인 브랜치로 갑니다
- Git switch / checkout 브랜치명
- Git rebase main

```
lhs@DESKTOP-86MUCGC MINGW64 /d/git/g-t (js)
$ git rebase main
Successfully rebased and updated refs/heads/js.
```

전원 Part, rebase 로 메인 코드 땡겨오기

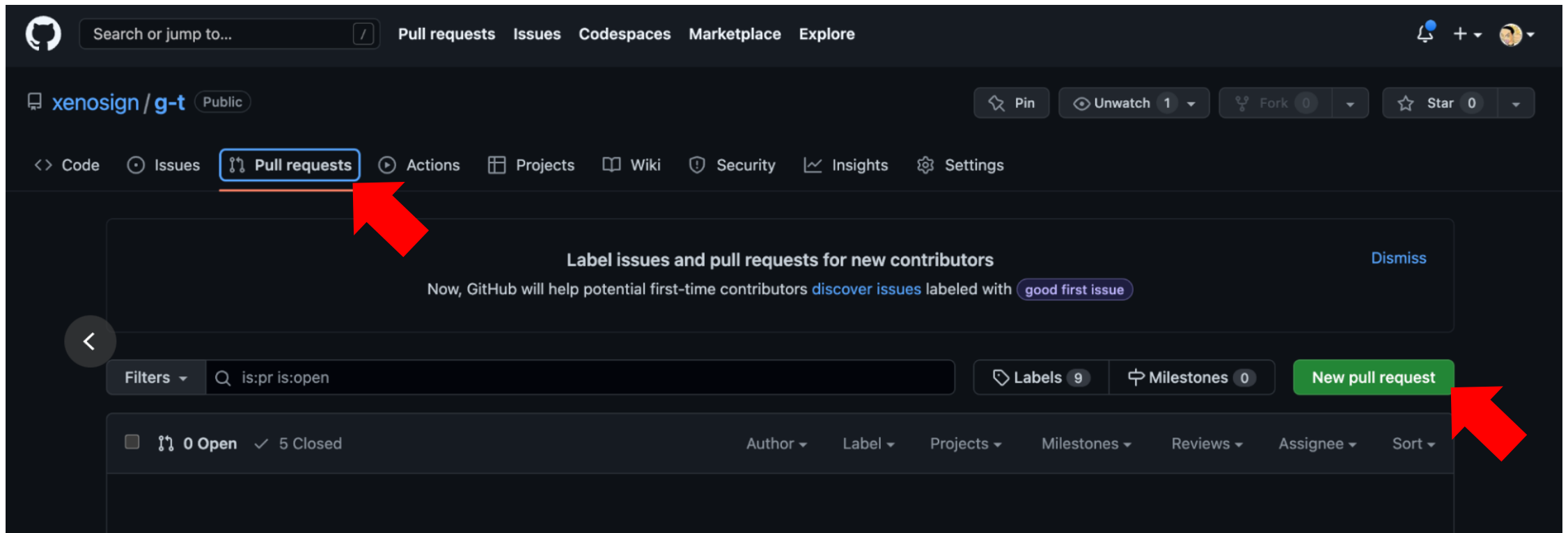


- 이제 main 의 최신 사항이 자신의 브랜치에 반영이 되었습니다!
- 그런데 지금 자신의 브랜치에 main 의 코드가 반영 된 것은 Github 에 반영이 되어 있을까요? → 당연히 안되어 있습니다!
- 일단 작업할 사항을 더 작업 → 작업 마무리 → commit → push → PR → Merge 를 하면 됩니다!

전원 Part, PR이 생기질 않아요!



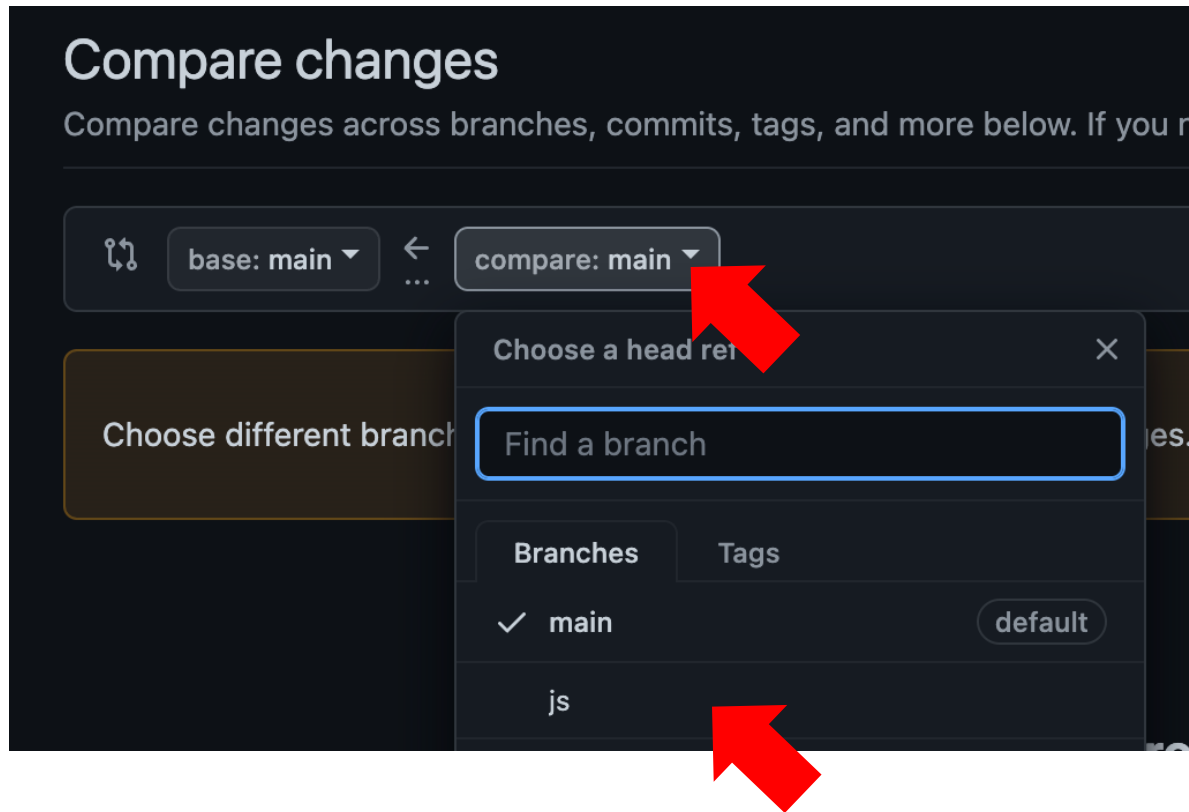
- Push 를 한 다음 Github에 갔을 때 자동으로 PR이 안생기는 경우가 있습니다! 이럴 땐?



전원 Part, PR이 생기질 않아요!



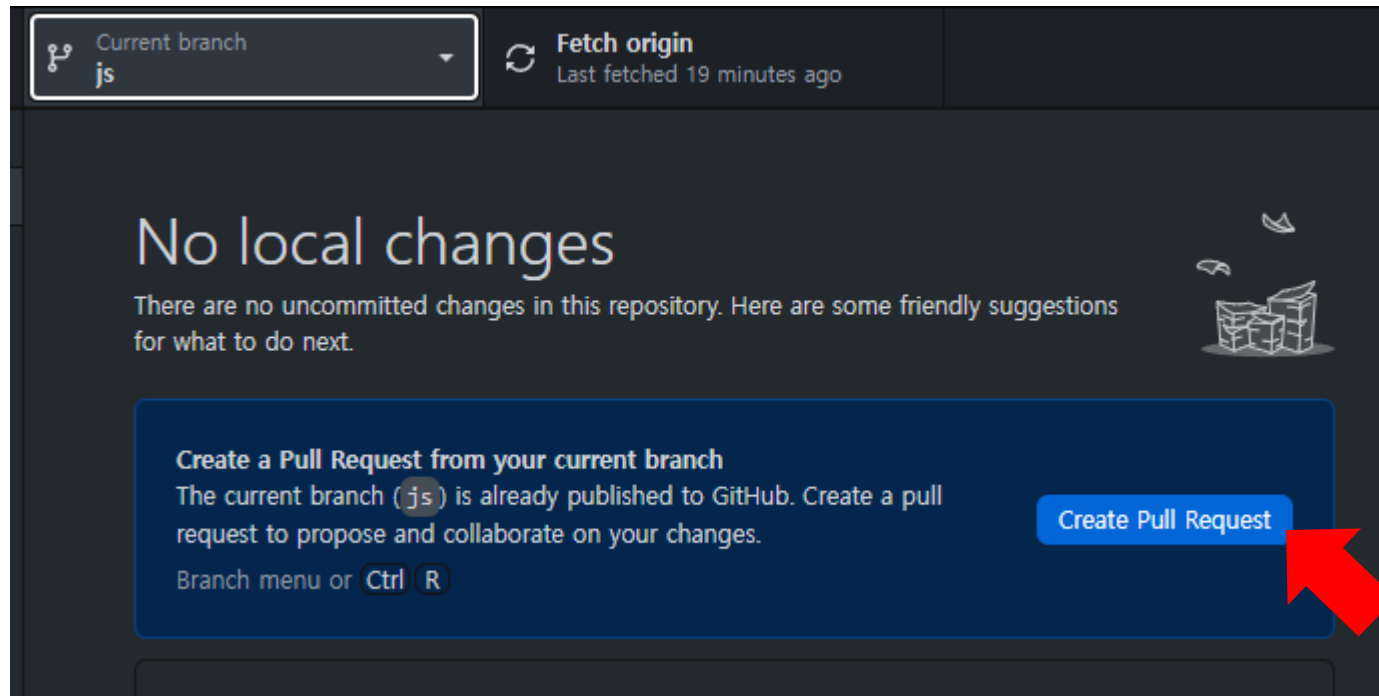
- 자신의 브랜치를 선택해 주시면 됩니다~!



전원 Part, PR이 생기질 않아요!



- 아니면 시간이 조금(?) 지나면 자동으로 생성되는 경우도 많습니다!
- 혹은 Github Desktop 을 이용하시면 됩니다!







영상 기반 랜딩 페이지

KDT Team Tetz 5th

😊 Codign with fun

👥 Coding with us

2023 KDT Tetz Team 5th

영상 기반 랜딩 페이지 <https://movie-main-page.netlify.app/>



영상부터 구해봅시다

<https://www.pexels.com/ko-kr/>

<https://pixabay.com/videos/>



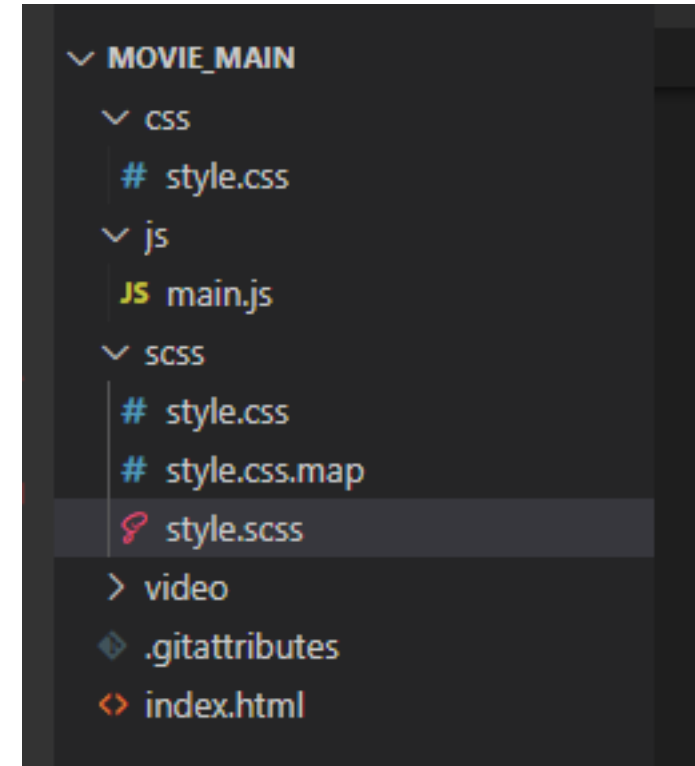
뼈대를 세우자

HTML



HTML 만들기 전에 세팅하기

- Index.html 문서 만들기
- Fontawesome 키트 추가하기
- JS 폴더 → 메뉴 처리를 위한 main.js 파일
추가 및 링크
- SCSS 폴더 → scss/style.css 링크
- Video 폴더 → 사용할 비디오 클립 삽입





기본 CSS 설정

기본 CSS 세팅



- 구글 폰트 가져오기

- Architects Daughter
- Fredericka the Great

- _reset.scss 구현

- 마진, 패딩 X / 보더 박스 설정
- 리스트 스타일 X
- A 태그 속성 X



```
/* RESET */
* {
  margin: 0px;
  padding: 0px;
  box-sizing: border-box;
}

ul,
ol {
  list-style: none;
}

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

_reset.scss



```
/* FONT */
@import
url("https://fonts.googleapis.com/css2?family=Architects+Daughter&family=F
redericka+the+Great&display=swap");

/* RESET */
@import "reset";

// COMMON
body {
  background-color: black;
}
```

style.scss



HEADER

HEADER, HTML



- 로고 역할을 하는 H1
- GNB(Global Navigation Bar)
 - 메뉴는 마음대로
 - about / team / news / community
- SNS 파트
 - 여기도 마음대로
 - Search / Mail / Facebook
- 반응형에 사용할 햄버거 메뉴



```
<!-- HEADER -->
<header>
  <h1><a href="#">TetzKDT</a></h1>

  <ul id="gnb">
    <li><a href="#">ABOUT</a></li>
    <li><a href="#">TEAM</a></li>
    <li><a href="#">NEWS</a></li>
    <li><a href="#">COMMUNITY</a></li>
  </ul>

  <ul class="sns">
    <li><a href="#"><i class="fas fa-search"></i></a></li>
    <li><a href="#"><i class="far fa-envelope"></i></a></li>
    <li><a href="#"><i class="fab fa-facebook-square"></i></a></li>
  </ul>

  <a href="#" class="btn_menu"><i class="fas fa-bars"></i></a>
</header>
```

HEADER, CSS



- Header

- Width: 100% / border-bottom: 1px solid #999;

- 가로로 배치 해야겠죠?

- 각각 아이টে을 양끝에 붙여서 균등 배치 → justify-content: space-between;
 - 수직 중앙 정렬 → align-items: center;
 - 너무 양끝에 붙어 있으니 여백 주기 → padding : 5vh 5vw;



```
/* HEADER */  
header {  
  width: 100%;  
  border-bottom: 1px solid #999;  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
  padding: 5vh 5vw;  
}
```

HEADER, CSS



- 버거 메뉴는 반응형에서 보일 요소이므로 → `display: none;`
- 로고 폰트 설정
 - `font: bold 24px/1 "Architects Daughter";`
 - `color: #555;`

```
header {  
  h1 a {  
    font: bold 24px/50px "Architects Daughter";  
    color: #555;  
  }  
  .btn_menu {  
    display: none;  
  }  
}
```



GNB

GNB, CSS



- 역시나 가로 배치 해야겠쥬? → `display: flex;`
- Li 태그
 - 각각 여백 주기 → `margin: 0px 20px;`
- A 태그
 - Font: 16px/1 "Architects Daughter";
 - Color: #555;
 - Letter-spacing: 1px;



```
header {  
  // 헤더 코드  
  
  #gnb {  
    display: flex;  
  
    li {  
      margin: 0px 20px;  
  
      a {  
        font: 16px/1 "Architects Daughter";  
        color: #333;  
        letter-spacing: 1px;  
      }  
    }  
  }  
}
```



SNS

실습, SNS 파트 작업하기!



- Gnb 파트를 참고하여 SNS 파트 구현!



```
.sns {  
  display: flex;  
  
  li {  
    margin: 0px 10px;  
  
    a {  
      font-size: 16px;  
      color: #333;  
    }  
  }  
}
```



Video Frame

Video Frame, HTML



- 멀티 미디어 콘텐츠의 경우 figure 태그를 사용
- Video 태그로 영상 삽입
 - 속성
 - Autoplay : 자동 재생 여부
 - Muted : 음소거 여부
 - Loop : 반복 여부
 - Controls : 컨트롤 패널 삽입 여부 (기본 속성 : X)
- 영상의 경우 muted 속성이 없으면 자동 재생이 안됩니다!



```
<!-- BACK VIDEO -->  
  <figure>  
    <video src="./video/video.mp4" autoplay muted loop></video>  
  </figure>
```


Video Frame, CSS



- **Figure**

- Width: 100%; / height: 100vh;
- position: absolute; / top: 0px / left: 0px;
- Overflow: hidden;
- Z-index: 1;

- **Video**

- Width: 100%;
- Height: 100%;
- Object-fit: cover; (background-size 와 동일)



```
/* VIDEO */
figure {
  width: 100%;
  height: 100vh;
  position: absolute;
  top: 0px;
  left: 0px;
  overflow: hidden;
  z-index: 1;

  video {
    width: 100%;
    height: 100%;
    object-fit: cover;
  }
}
```

Video Frame, Text HTML



- 특정 섹션의 내용이므로 article 사용, class → text
- P 태그로 텍스트 삽입

```
<figure>
  <video src="./video/video.mp4" autoplay muted loop></video>
  <article class="text">
    <p>KDT<br>Team Tetz 1st</p>
  </article>
</figure>
```

Video Frame, Text CSS



- 텍스트 영역 설정
 - Position: absolute;
 - Left: 6vw; / top: 50%; / translate(0, -50%);
- P 태그
 - Font: normal 8vw/1 "Fredericka the Great";
 - Color: rgb(218, 218, 218);



```
figure {  
  
  .text {  
    position: absolute;  
    left: 6vw;  
    top: 45%;  
    transform: translate(0, -50%);  
  
    p {  
      font: normal 8vw/1 "Fredericka the Great";  
      color: rgb(218, 218, 218);  
    }  
  }  
}
```

Video Frame, Text HTML



- 같은 요소가 2번 반복!? → list 사용

```
<ul>
  <li>
    <a href="#">
      <i class="fa-solid fa-face-laugh-wink"></i>
      <strong>Codign with fun</strong>
    </a>
  </li>
  <li>
    <a href="#">
      <i class="fa-solid fa-people-group"></i>
      <strong>Coding with us</strong>
    </a>
  </li>
</ul>
```

Video Frame, Text CSS



- UI
 - 가로 배치? Display → flex;
 - Margin-top: 30px;
- Li
 - Margin: 0px 30px;
- i 태그
 - Font-size: 20px / color: rgb(218, 218, 218);
 - Margin-right: 10px;
- Strong
 - Font: 14px/1 "Architects Daughter"; / color: rgb(218, 218, 218);
 - Letter-spacing: 1px;

```
figure {
  .text {
    ul {
      display: flex;
      margin-top: 30px;
    }

    li {
      margin: 0px 30px;

      i {
        font-size: 20px;
        color: rgb(218, 218, 218);
        margin-right: 10px;
      }

      strong {
        font: 14px/1 "Architects Daughter";
        color: rgb(218, 218, 218);
        letter-spacing: 1px;
      }
    }
  }
}
```





FOOTER

FOOTER, HTML



```
<!-- FOOTER -->  
<footer>  
    2022 KDT Tetz Team 1st  
</footer>
```

FOOTER, CSS



```
/* FOOTER */
footer {
  width: 100%;
  position: absolute;
  bottom: 0px;
  left: 0px;
  font: 12px/1 "arial";
  color: rgb(218, 218, 218);
  letter-spacing: 2px;
  text-align: center;
  padding: 5vh 5vw;
  z-index: 2;
}
```



반응형

작업하기

메뉴가 잘리는 영역 파악 or 데이터로!



- 개발자 도구를 사용하여 메뉴가 잘리기 시작하는 넓이 값을 바탕으로 반응형 작업 or 일반적 크기(태블릿, 핸드폰)에 맞춰서 작업
- 우리는 잘리기 시작하는 넓이로 설정!

```
/* MOBILE */  
@media screen and (max-width: 968px) {}
```

반응형 CSS 작업



- 안보이게 처리 할 것
 - Gnb
 - SNS
- 보이게 처리할 것
 - Btn_menu
- 반응형에서 텍스트 크기 키우기!



```
/* MOBILE */
@media screen and (max-width: 968px) {
  header {
    #gnb {
      display: none;
    }

    .sns {
      display: none;
    }

    #snb {
      display: block;
    }

    .btn_menu {
      display: block;
    }

    figure {
      .text {
        top: 50%;

        p {
          font-size: 15vw;
        }
      }
    }
  }
}
```



Snb

추가하기

Snb, HTML



- Snb 영역을 따로 만들기 → div
- btn_menu 는 snb 안으로
- Gnb 의 메뉴를 그대로 옮길 것이므로 UL 사용
- GNB 의 메뉴 그대로 가져오기
- 마지막 Li 는 SNS 항목 추가



```
<div id="snb">
  <a href="#" class="btn_menu"><i class="fas fa-bars"></i></a>
  <ul class="sub_menu hide">
    <li><a href="#">ABOUT</a></li>
    <li><a href="#">TEAM</a></li>
    <li><a href="#">NEWS</a></li>
    <li><a href="#">COMMUNITY</a></li>
    <li>
      <a href="#"><i class="far fa-envelope"></i></a>
      <a href="#"><i class="fab fa-facebook-square"></i></a>
    </li>
  </ul>
</div>
```

Snb, CSS



- Sub_menu

- 문서 흐름에서 탈출 + 위치 내 맘대로 → `position: absolute;`
- 배경을 영상 배경과 비슷한 색으로 설정
- 오른쪽 여백은 햄버거 메뉴와 동일하게 → `right: 5vw;`
- 적당한 padding 주기
- 헤더의 border 와 동일한 border 주기
- Border-radius 취향대로 설정
- 글자 오른쪽 정렬!
- 버튼 클릭에 대한 반응을 `opacity` 로 할 것이므로 `opacity: 1;`
- `Transition: 0.4s;`



```
header {  
  
  #snb {  
  
    .sub_menu {  
      position: absolute;  
      padding: 10px 25px;  
      background-color: rgb(173, 179, 164);  
      border: 1px solid #999;  
      border-radius: 10px;  
      right: 5vw;  
      text-align: right;  
      opacity: 1;  
      transition: 0.4s;  
  
      &.hide {  
        opacity: 0;  
      }  
    }  
  }  
}
```

Snb, CSS



- Sub_menu li
 - 적절한 상하 여백
 - 폰트 설정
- Sub_menu li a i
 - 여백 주기



```
header {  
  
  #snb {  
  
    .sub_menu {  
  
      li {  
        padding: 10px 0px;  
        font: 16px/1 "Architects Daughter";  
  
        a i {  
          margin-left: 8px;  
        }  
      }  
    }  
  }  
}
```



JS

기능 추가

Btn_menu 기능 추가



- Sub_menu의 hide class 존재 여부에 따라 클래스 추가 → 삭제 구현

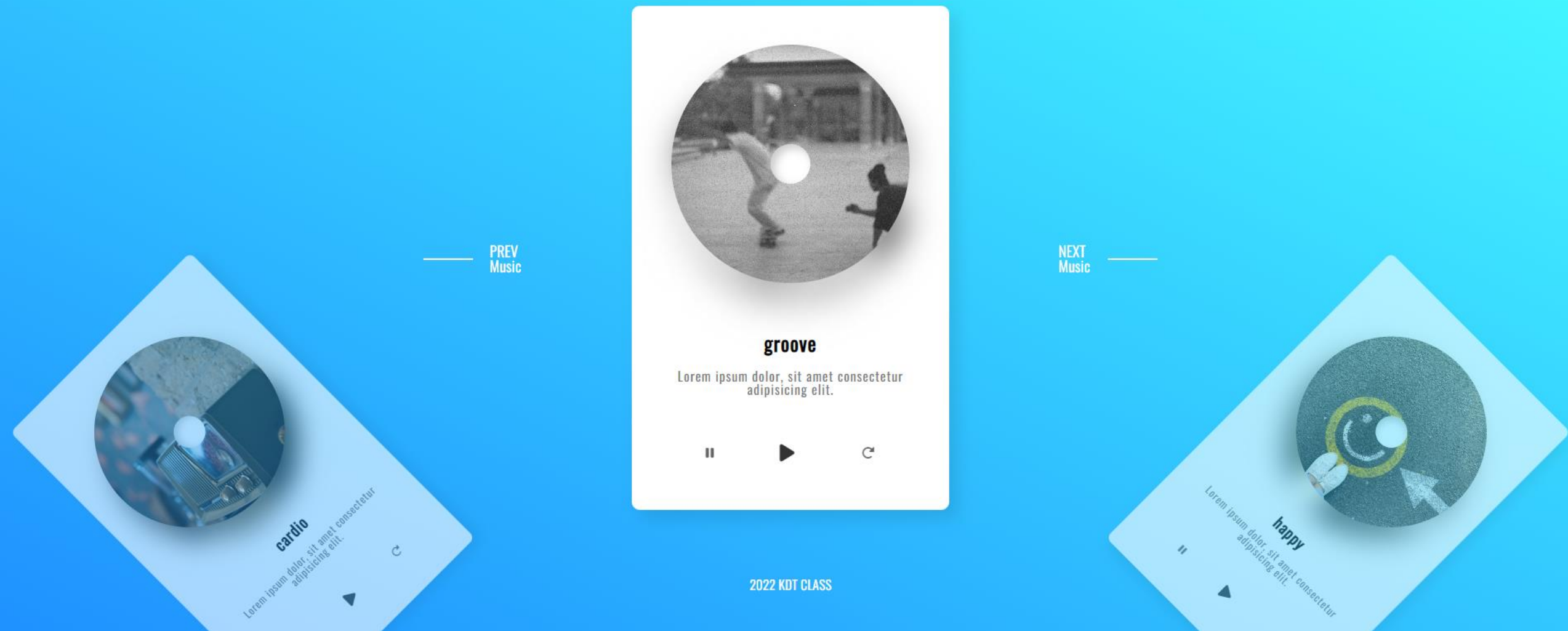
```
const snbBtn = document.querySelector("#snb .btn_menu");
const subMenu = document.querySelector("#snb .sub_menu");

snbBtn.addEventListener("click", function () {
  if (subMenu.classList.contains("hide")) {
    subMenu.classList.remove("hide");
  } else {
    subMenu.classList.add("hide");
  }
});
```






뮤직 플레이어 제작!



<https://tetz-music-player.netlify.app/>



기본 세팅하기~!



기본 세팅하기!

- 음악 및 이미지 파일 다운로드!
- 폰트 어썸 연결하기!
- JS 파일 연결하기!(Defer)
- Scss 파일 컴파일 → CSS 파일 링크!
- 폰트 가져오기
 - Oswald / VT323 (각자 원하는 것으로 가져오기!)
- _reset 설정



```
<script
  src="https://kit.fontawesome.com/d15cbbf71d.js"
  crossorigin="anonymous"
></script>
<link rel="stylesheet" href="./css/style.css" />
<script defer src="./js/main.js"></script>
```

Index.html

```
* {
  margin: 0px;
  padding: 0px;
  box-sizing: border-box;
}
```

```
ul,
ol {
  list-style: none;
}
```

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

_reset.scss



```
@charset "UTF-8";  
@import "reset";  
  
@import url("https://fonts.googleapis.com/css2?family=VT323&display=swap");  
// font-family: 'VT323', monospace;  
@import url("https://fonts.googleapis.com/css2?family=Oswald&display=swap");  
// font-family: "Oswald", sans-serif;
```

style.scss



기본 UI 설정하기!

기본 UI, HTML 작업



- Figure 태그 하나 사용!
- H1 으로 로고 작업!
- A 태그로 bars 메뉴 설정(실제로 작동은 X)
- P 태그로 푸터 설정!



```
<body>
  <figure>
    <h1>
      <strong>TetzKDT</strong><br>
      <span>Coding with fun</span>
    </h1>

    <a href="#" class="menu">
      <i class="fas fa-bars"></i>
    </a>

    <p>2022 KDT CLASS</p>
  </figure>
```

기본 UI, CSS 작업!



- Body

- 전체 폰트 설정

- Figure

- 뷰포트 만큼 크기 주기
 - Overflow: hidden
 - Position: relative
 - Background: linear-gradient(25deg, dodgerblue, rgb(71, 255, 255));

```
body {  
  font: 16px/1 "Oswald";  
}  
  
figure {  
  width: 100%;  
  height: 100vh;  
  overflow: hidden;  
  position: relative;  
  background: linear-gradient(25deg, dodgerblue, rgb(71, 255, 255));  
}
```



기본 UI, CSS 작업!



- H1
 - Position: absolute;
 - Top: 7vh / left: 4vw
 - 줄 간격을 없애기 위하여 → font-size: 0;
- Strong
 - Font: bold 40px/1.4 "Oswald"
 - Color: #fff / letter-spacing: 1px;
- Span
 - Font: 16px/1 "Oswald" / color: #fff / opacity: 0.8 / letter-spacing: 1px



```
figure {  
  h1 {  
    position: absolute;  
    top: 7vh;  
    left: 4vw;  
    font-size: 0;  
  
    strong {  
      font: bold 40px/1.4 "Oswald";  
      color: #fff;  
      letter-spacing: 1px;  
    }  
  
    span {  
      font: 16px/1 "Oswald";  
      color: #fff;  
      opacity: 0.8;  
      letter-spacing: 1px;  
    }  
  }  
}
```

기본 UI, CSS 작업!



- `.menu`
 - Position: absolute / top: 8vh / right: 4vw
 - font-size: 30px / color: #fff
- `> p {`
 - Position: absolute
 - bottom: 7vh / left: 50% / transform: translate(-50%, 0);
 - color: #fff



```
figure {  
  
  .menu {  
    position: absolute;  
    top: 8vh;  
    right: 4vw;  
    font-size: 30px;  
    color: #fff;  
  }  
  
  > p {  
    position: absolute;  
    bottom: 7vh;  
    left: 50%;  
    transform: translate(-50%, 0);  
    font: 16px/1 "Oswald";  
    color: #fff;  
  }  
}
```




음악 패널 작성하기!

패널, HTML 작업



- Section 태그 사용!
- 각각의 음악 패널은 article 로 작업
 - 각각 패널마다 transform 효과를 주어야 하므로 div.inner 를 배치하여 transform 이 중복되어 효과가 안걸리는 문제를 회피!

```
section>article*8>.inner
```

```
<section>  
  <article>  
    <div class="inner"></div>  
  </article>
```

패널, CSS 작업



- 패널은 CSS 를 컴포넌트로 만들어 작업!
- _panel.scss 파일 작성
- Style.scss 파일에 panel 임포트

```
@import "panel";
```

패널, CSS 작업



- `_panel.scss` 파일에서 작업 시작!
- Section
 - Width: 20vw / height: 65vh
 - Position: absolute / left: 50% / top : 50% / margin-left: -10vw / margin-top: 32.5vh
- Article
 - Width: 100% / height: 100%
 - Position : absolute / top : 0px / left: 0px

```
section {  
  width: 20vw;  
  height: 65vh;  
  position: absolute;  
  left: 50%;  
  top: 150%;  
  margin-left: -10vw;  
  margin-top: -32.5vh;  
  
  article {  
    width: 100%;  
    height: 100%;  
    position: absolute;  
    top: 0px;  
    left: 0px;
```



패널, CSS 작업



- **.inner**
 - Width: 100% / height: 100% / background-color: #fff
 - Padiing: 5vh 2.5vw 8vh / border-radius: 10px;
 - Box-shadow: 10px 10px 20px rgba(0, 0, 0, 0.1);
 - Opacity: 0.6

```
section {  
  article {  
    .inner {  
      width: 100%;  
      height: 100%;  
      background-color: #fff;  
      padding: 5vh 2.5vw 8vh;  
      border-radius: 10px;  
      box-shadow: 10px 10px 20px rgba(0, 0, 0, 0.1);  
      opacity: 0.6;  
    }  
  }  
}
```



패널, 회전 주기



- 각각의 패널에 회전 값을 JS 를 통해서 각각의 회전 값을 한번에 처리해 봅시다!
- Section 요소 가져오기
- Section 내부의 article 요소 전부 가져오기
- Article 개수 선언 및 각도 구하기


```
const frame = document.querySelector("section");
const list = frame.querySelectorAll("article");
const len = list.length;
const deg = 360 / len;

for (let i = 0; i < len; i++) {
  list[i].style.transform = `rotate(${deg * i}deg`;
}
```



패널, 회전 주기



- 패널이 지금은 자기 자신의 중앙을 기준으로 회전 되어 있으므로 특정 기준 점을 기준으로 퍼져야 하므로 translateY 값 부여

```
for (let i = 0; i < len; i++) {  
  list[i].style.transform = `rotate(${deg * i}deg) translateY(-100vh)`;
```

- 패널이 너무 위로 가있으므로!
- Section 의 top → 150%

```
section {  
  width: 20vw;  
  height: 65vh;  
  position: absolute;  
  left: 50%;  
  top: 150%;  
  margin-left: -10vw;  
  margin-top: -32.5vh;  
  transition: 1s;
```



좌우 이동
버튼 만들기!

버튼, HTML 작업



- Section 태그 아래에 nav 태그로 작업!
- 이전, 이후 버튼이 있어야 하므로! 2개 작업

```
<nav class="btnPrev">  
  <span>PREV Music</span>  
</nav>  
  
<nav class="btnNext">  
  <span>NEXT Music</span>  
</nav>
```

버튼, CSS 작업



- 버튼도 컴포넌트로 작업!
- _btns.scss 파일 생성
- Style.scss 파일에 임포트!

```
@import "panel";  
@import "btns";
```

버튼, CSS 작업



- **.btnPrev**
 - Width: 60px / height: 60px
 - Position: absolute / top: 50% / left: 50%
 - Transform(-20vw, -50%)
 - Display: flex / justify-content: flex-start / align-items: center
 - Font-size: 0 / padding-left: 20px / cursor: pointer
- **Span**
 - Font-size: 18px / color: #fff / transition: 0.5s

```
.btnPrev {  
  width: 60px;  
  height: 60px;  
  position: absolute;  
  top: 50%;  
  left: 50%;  
  transform: translate(-20vw, -50%);  
  display: flex;  
  align-items: center;  
  justify-content: flex-start;  
  cursor: pointer;  
  font-size: 0;  
  padding-left: 20px;  
  
  span {  
    font-size: 18px;  
    color: #fff;  
    transition: 0.5s;  
  }  
}
```





버튼, 움직이는 화살표 만들기

- 요런거는 보통 가상 요소 선택자로 만듭니다!
- `::Before` / `::after` 둘 다 선언해서 만들고 Ani 효과 주기
- `::Before`
 - `Content: ""`
 - `Display: block / width: 100% / height: 2px / background-color: #fff`
 - `Position: absolute / left: 0px`
 - `Transform-origin: left center / transform: rotate(-180deg)`
 - `Transition: 0.5s`


```
.btnPrev {  
  &::before,  
  &::after {  
    content: "";  
    display: block;  
    width: 100%;  
    height: 2px;  
    background-color: #fff;  
    position: absolute;  
    left: 0px;  
    transform-origin: left center;  
    transform: rotate(-180deg);  
    transition: 0.5s;  
  }  
}
```

```
&::after {  
  transform: rotate(180deg);  
}
```



버튼, 움직이는 화살표 만들기



- Hover 효과 주기
 - 글자는 속 사라지게!
 - ::Before / ::After 는 각도를 변경해서 화살표처럼 보이도록
 - 30도를 기준으로 위 아래로 위치 하도록 설정!



```
.btnPrev {  
  &:hover {  
    span {  
      transform: translate(100%, 0);  
      opacity: 0;  
    }  
  
    &::before {  
      transform: rotate(-30deg);  
    }  
  
    &::after {  
      transform: rotate(30deg);  
    }  
  }  
}
```

실습, Next 버튼도 작업해 봅시다!



- Prev 버튼과 거의 유사한 Next 버튼 작업하기!



좌우 이동 버튼

클릭 액션 처리

버튼, JS 처리!



- 버튼을 각각 불러옵니다!
- 해당 패널의 위치 값을 저장할 변수 만들기!
- 각각 버튼에 이벤트 붙이기!
- 0을 기준으로 next 를 누르면 $+1 * 45\text{deg}$ 씩 돌아가고
- Prev 를 누르면 $-1 * 45\text{deg}$ 씩 돌아가면 됩니다!
- 다만, 해당 속성 값은 누적이 아니므로 next 가 2번 눌리면 45deg 에서 90deg 로 돌아야 합니다!
- 즉, 해당 패널의 위치 값을 0 을 기준으로 정해서 곱하면 됩니다!



```
const prev = document.querySelector(".btnPrev");
const next = document.querySelector(".btnNext");
let num = 0;

prev.addEventListener("click", function (e) {
  frame.style.transform = `rotate(${deg * ++num}deg)`;
});

next.addEventListener("click", function (e) {
  frame.style.transform = `rotate(${deg * --num}deg)`;
});
```

- 돌아가는 애니메이션을 줘야 하는데, 돌아가는 UI는 Section 이므로 Section 에 transition: 1s; 값 추가

```
section {
  transition: 1s;
```



가운데 패널에 활성화 주기

활성화 패널, on 클래스 부여



- 첫번째 패널인 Article 에 on 클래스를 부여

```
<section>
  <article class="on">
    <div class="inner">
```

- On 클래스를 가진 패널은
 - Opacity: 1 / transform: scale(1) 부여

```
&.on {
  .inner {
    opacity: 1;
    transform: scale(1);
  }
}
```

활성화 패널, JS 처리하기



- 전역 변수로 활성화 패널의 위치를 저장 → `let active = 0;`
- Prev 버튼 처리
 - 초창기 위치가 0
 - 0에서 prev 가 눌리면? → 제일 마지막(article 의 개수 - 1)으로 이동
 - 그 외에는 자신의 값에서 -1 만 해주면 됩니다!
 - 그리고 현재 모든 article 에서 on 이라는 클래스를 제거 후 → active 위치에
만 on 클래스 추가!



```
let active = 0;

prev.addEventListener("click", function (e) {
  frame.style.transform = `rotate(${deg * ++num}deg)`;

  if (active === 0) {
    active = len - 1;
  } else {
    active--;
  }

  for (let el of list) {
    el.classList.remove("on");
  }
  list[active].classList.add("on");
});
```

활성화 패널, JS 처리하기



- Next 버튼 처리

- 이 친구는 반대로, 마지막 위치에서 next 가 눌리면 0으로 변경!
- 나머지는 클릭 되면 +1 해주면 되겠죠?
- 역시 모든 article 에서 on 클래스 제거 후 → active 위치에 on 클래스 부여



```
next.addEventListener("click", function (e) {  
    frame.style.transform = `rotate(${deg * --num}deg)`;  
  
    if (active === len - 1) {  
        active = 0;  
    } else {  
        active++;  
    }  
  
    for (let el of list) {  
        el.classList.remove("on");  
    }  
    list[active].classList.add("on");  
});
```



패널 꾸미기

앨범 사진, HTML 작업



- 먼저 앨범 사진을 넣어 보아요!
- 사진을 넣을 div 요소 pic 삽입!
- CD 처럼 보이기 위한 가운데 구멍 dot 도 삽입!

```
<section>
  <article class="on">
    <div class="inner">
      <div class="pic">
        <div class="dot"></div>
      </div>
    </div>
  </article>
```

앨범 사진, CSS 작업



- .pic
 - Width: 15vw / height: 15vw;
- 실제 사진은 ::before 와 ::after 를 사용해서 처리 합니다!
 - Content: ""
 - Display: block / width: 100% / height: 100%
 - position: absolute / top: 0px / left: 0px / border-radius: 50%
 - background-repeat: no-repeat / background-position: center / background-size: cover
 - 이미지 주소로 이미지 삽입!


```
section {
  article {
    .inner {
      .pic {
        height: 15vw;
        width: 15vw;
        position: relative;

        &::before,
        &::after {
          content: "";
          display: block;
          width: 100%;
          height: 100%;
          position: absolute;
          top: 0px;
          left: 0px;
          border-radius: 50%;
          background-repeat: no-repeat;
          background-position: center;
          background-size: cover;
          background-image: "url()";
        }
      }
    }
  }
}
```



앨범 사진, CSS 작업



- 지금은 사진 2장이 겹쳐진 상태이죠?
- ::before 만 따로 이미지를 처리해서 자연스러운 그림자 효과를 줍시다
 - Transform: translste(0, 10%) / filter: blur(20px) brightness(130%);
- .dot 가운데 구멍도 처리해 보죠
 - Width: 2.5vw / height: 2.5vw / postion: absolute / top: 50% / left: 50% / transform: translate(-50%, -50%) / background-color: #fff / border-radius: 50% / box-shadow: inset 5px 5px 10px rgba(0, 0, 0, 0.2); / z-index: 3



```
section {
  article {
    .inner {
      .pic {

        ::before {
          transform: translate(0, 10%);
          filter: blur(20px) brightness(130%);
        }

        .dot {
          width: 2.5vw;
          height: 2.5vw;
          position: absolute;
          top: 50%;
          left: 50%;
          transform: translate(-50%, -50%);
          background-color: #fff;
          border-radius: 50%;
          box-shadow: inset 5px 5px 10px rgba(0, 0, 0, 0.2);
          z-index: 3;
        }
      }
    }
  }
}
```

앨범 사진, JS로 일괄 삽입을 해봅시다!



- 현재 앨범 사진과 음악 mp3 파일명은 동일한 상태이죠?
- JS에서 배열을 사용해서 일괄적으로 이미지를 삽입해 BoA요!
 - 각각의 파일명이 담긴 배열 만들기
 - 이미 사용한 반복문이 있으니, 해당 반복문에 처리를 해봅시다!



```
const names = ["cardio", "groove", "happy", "light", "lily", "limes", "pop", "swing"];

for (let i = 0; i < len; i++) {
  list[i].style.transform = `rotate(${deg * i}deg) translateY(-100vh)`;

  const pic = list[i].querySelector(".pic");
  pic.style.backgroundImage = `url(../img/${names[i]}.jpg)`;
}
```

앨범 사진, before / after 에 상속 시키기



- 지금은 .pic 에 사진이 들어간 상태입니다!
- .pic 사진을 저 멀리 먼저 치웁시다
 - Background-repeat: no-repeat / background-position: -9999px - 9999px;
- 그리고 ::before ::after 에 .pic 의 사진 위치를 상속 시킵니다!



```
.pic {  
  height: 15vw;  
  width: 15vw;  
  position: relative;  
  background-repeat: no-repeat;  
  background-position: -9999px -9999px;  
  
  &::before,  
  &::after {  
    content: "";  
    display: block;  
    width: 100%;  
    height: 100%;  
    position: absolute;  
    top: 0px;  
    left: 0px;  
    border-radius: 50%;  
    background-repeat: no-repeat;  
    background-position: center;  
    background-size: cover;  
    background-image: inherit;  
  }  
}
```

앨범 제목, HTML 영역



- .text 영역 만들기

- H2

- P

```
<section>
  <article class="on">
    <div class="inner">
      <div class="pic">
        <div class="dot"></div>
      </div>
      <div class="text">
        <h2></h2>
        <p>Lorem ipsum dolor, sit amet consectetur.</p>
      </div>
    </div>
  </article>
```


앨범 제목, JS로 제목도 일괄 적용



- 이미 선언한 이름 배열을 또 쓰면 되겠죠?

```
const names = ["cardio", "groove", "happy", "light", "lily", "limes", "pop", "swing"];

for (let i = 0; i < len; i++) {
  list[i].style.transform = `rotate(${deg * i}deg) translateY(-100vh)`;

  const pic = list[i].querySelector(".pic");
  pic.style.backgroundImage = `url(../img/${names[i]}.jpg)`;

  const title = list[i].querySelector(".text>h2");
  title.innerHTML = `${names[i]}`;
}
```

앨범 제목, text 영역 CSS 작업



- **.text**
 - Width: 15vw / text-align: center
 - Position: absolute / margin-top: 60px / letter-spacing: 1px;
- **H2**
 - Margin-bottom: 20px;
- **P**
 - Color: #777;



```
section {  
  article {  
    .inner {  
  
      .text {  
        width: 15vw;  
        text-align: center;  
        position: absolute;  
        margin-top: 60px;  
        letter-spacing: 1px;  
  
        h2 {  
          margin-bottom: 20px;  
        }  
  
        p {  
          color: #777;  
        }  
      }  
    }  
  }  
}
```



음악 플레이어, HTML 작업

- UI 태그 + icon 으로 음악 플레이어를 만듭시다!
- UI > li * 3
 - 각각 li 에 icon 으로 버튼 주기

```
<div class="text">
  <h2></h2>
  <p>Lorem ipsum dolor, sit amet consectetur adipisicing elit.</p>
  <ul class="control">
    <li class="pause"><i class="fas fa-pause"></i></li>
    <li class="play"><i class="fas fa-play"></i></li>
    <li class="reload"><i class="fas fa-redo-alt"></i></li>
  </ul>
</div>
```

음악 플레이어, CSS 작업



- UI
 - 가로 배치죠? → `display: flex`
 - 버튼이 적절히 퍼지도록 → `justify-content: space-around`
- UI li
 - `Cursor: pointer;`
 - `Opacity: 0.6 / transition: 0.5s`
- `&.play`
 - 플레이 버튼은 더 크고 선명하게 보이도록 → `transform: scale(1.5) / opacity: 0.8;`

```
section {  
  article {  
    .inner {  
      .text {  
  
        .control {  
          display: flex;  
          justify-content: space-around;  
          margin-top: 60px;  
  
          li {  
            cursor: pointer;  
            opacity: 0.6;  
            transition: 0.5s;  
  
            &.play {  
              transform: scale(1.5);  
              opacity: 0.8;  
            }  
          }  
        }  
      }  
    }  
  }  
}
```



음악 플레이어, CSS 작업



- Hover 효과 주기
 - 각각의 버튼 → 1.5배 / opacity: 0.8
 - Play 버튼 → 1.8배 / opacity: 1

```
li {  
  
    &:hover {  
        transform: scale(1.5);  
        opacity: 0.8;  
  
        &.play {  
            transform: scale(1.8);  
            opacity: 1;  
        }  
    }  
}
```



앨범 이미지 회전

앨범 회전, Keyframes 작업



- 앨범은 ::before 와 ::after 2개의 이미지가 있으므로 2개의 keyframes 작업 필요
- 하나는 자신의 위치에서 360deg 돌고
- 하나는 자신의 위치 10% 아래에서 360deg 돌면 되므로!



```
@keyframes rotation {  
  0% {  
    transform: rotate(0deg);  
  }  
  
  100% {  
    transform: rotate(360deg);  
  }  
}  
  
@keyframes rotation2 {  
  0% {  
    transform: translateY(10%) rotate(0deg);  
  }  
  
  100% {  
    transform: translateY(10%) rotate(360deg);  
  }  
}
```

앨범 회전, CSS 작업



- 사진 요소에 on 이라는 클래스가 부여되면 회전 하도록 설정!

```
.pic {  
  &.on {  
    &::before {  
      animation: rotation2 4s linear infinite;  
    }  
  
    &::after {  
      animation: rotation 4s linear infinite;  
    }  
  }  
}
```

앨범 회전, JS 작업



- Play 버튼을 누르면 on 클래스가 부여 되도록 처리!
- Pause 버튼을 누르면 on 클래스가 제거 되도록 처리!



```
for (let el of list) {  
  const play = el.querySelector(".play");  
  const pause = el.querySelector(".pause");  
  const load = el.querySelector(".reload");  
  
  play.addEventListener("click", function (e) {  
    e.currentTarget.closest("article").querySelector(".pic").classList.add("on");  
  });  
  
  pause.addEventListener("click", function (e) {  
    e.currentTarget.closest("article").querySelector(".pic").classList.remove("on");  
  });  
}
```



음악 파일

삽입 및 재생

음악 파일, JS로 일괄 삽입을 해봅시다!



- 아까 쓰던 반복문이 있네요!?
- createElement 로 audio 엘리먼트 만들기
- setAttribute로 위치(src) 속성 입력하기
- 재생이 무한히 반복해야 하므로 loop 속성도 주기
- 각각의 list 요소에 audio 붙여넣기!



```
for (let i = 0; i < len; i++) {  
  list[i].style.transform = `rotate(${deg * i}deg) translateY(-100vh)`;  
  
  const pic = list[i].querySelector(".pic");  
  pic.style.backgroundImage = `url(../img/${names[i]}.jpg)`;  
  
  const title = list[i].querySelector(".text>h2");  
  title.innerHTML = `${names[i]}`;  
  
  const audio = document.createElement("audio");  
  audio.setAttribute("src", `../music/${names[i]}.mp3`);  
  audio.setAttribute("loop", "loop");  
  
  list[i].append(audio);  
}
```


음악 파일, JS로 재생 처리!



- 아까 play 랑 pause 버튼 작업 했었죠!? 거기로 고고싱
- 재생 버튼을 클릭하면 audio 태그를 선택 후 play() 메소드로 재생
- 멈춤 버튼을 클릭하면 audio 태그를 선택 후 pause() 메소드로 멈춤
- 이번에는 reload 버튼도 작업 합시다!
 - 재생과 완전히 동일하지만 처음부터 틀어야 하므로
 - Load() 로 음악을 다시 로드 시킨 후 → play()

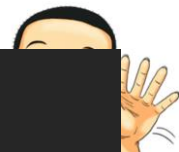
```
for (let el of list) {
  const play = el.querySelector(".play");
  const pause = el.querySelector(".pause");
  const load = el.querySelector(".reload");

  play.addEventListener("click", function (e) {
    e.currentTarget.closest("article").querySelector(".pic").classList.add("on");
    e.currentTarget.closest("article").querySelector("audio").play();
  });

  pause.addEventListener("click", function (e) {
    e.currentTarget.closest("article").querySelector(".pic").classList.remove("on");
    e.currentTarget.closest("article").querySelector("audio").pause();
  });

  load.addEventListener("click", function (e) {
    e.currentTarget.closest("article").querySelector(".pic").classList.add("on");

    e.currentTarget.closest("article").querySelector("audio").load();
    e.currentTarget.closest("article").querySelector("audio").play();
  });
}
```



음악 파일, 여러 개가 한꺼번에 재생?



- 다른 노래가 재생 되면 다른 노래가 멈추도록 작업해 봅시다!!
- 이전 노래의 위치를 저장할 before 라는 전역 변수 생성!
- Before 의 초기 값을 0으로 세팅하고 before 가 0일 경우 현재 패널의 위치를 저장
- 다른 패널에서 재생 또는 Reload 버튼을 눌렀을 경우 이전 위치의 노래를 정지
 - 현재 클릭이 된 패널의 위치와 before 를 비교하여 서로 다를 경우 before 의 노래를 정지 → 현 위치를 before 로 등록!



```
play.addEventListener("click", function (e) {  
    if (before === 0) {  
        before = e.currentTarget;  
    } else if (before !== e.currentTarget) {  
        before.closest("article").querySelector("audio").pause();  
        before.closest("article").querySelector(".pic").classList.remove("on");  
        before = e.currentTarget;  
    }  
  
    e.currentTarget.closest("article").querySelector(".pic").classList.add("on");  
  
    e.currentTarget.closest("article").querySelector("audio").play();  
});
```



