

Hello,

KDT 웹 개발자 양성 프로젝트

5기!

with





문자 관련

JS 기능들!

문자열 길이, .length



- 문자열.length();

```
const str = "Hello, world";  
  
console.log(str.length);  
  
// 12
```

문자열 위치 찾기, indexOf



- 문자를 찾을 문자열.**indexOf("찾을 문자열")**:

```
const str = "Hello, world";  
  
console.log(str.indexOf("world"));  
  
// 7
```

문자열 자르기, slice



- 문자열.slice(시작위치, 종료위치);

```
const str = "Hello, world";  
  
console.log(str.slice(0, 5));  
  
//Hello
```

문자열 바꾸기, replace



- 문자열.replace("찾을 문자", "바꿀 문자");

```
const str = "Hello, world";  
  
console.log(str.replace("world", "뽀로로"));  
  
// Hello, 뽀로로
```

문자열 반복하기, repeat



- 문자열.repeat(반복 횟수);

```
let str = "우영";  
  
let longStr = str.repeat(10) + "우";  
  
console.log(longStr);
```

앞 뒤 공백 문자 제거하기, trim



- 문자열.trim():

```
const str = "Hello, world";
console.log(str.trim());
// Hello, world
```



숫자 + 수학 관련

JS 기능들!

수학 함수들



```
console.log("abs: ", Math.abs(-999));           // 절대값  
  
console.log("min: ", Math.min(10, 33));          // 작은 값  
console.log("min: ", Math.min(10, 33, 6, 2, -1)); // 작은 값  
  
console.log("max: ", Math.max(10, 33));          // 큰 값  
console.log("max: ", Math.max(10, 33, 2, 1, 77)); // 큰 값  
  
console.log("ceil: ", Math.ceil(3.14));           // 올림  
  
console.log("floor: ", Math.floor(3.14));          // 버림  
  
console.log("round: ", Math.round(3.6));           // 반올림  
console.log("round: ", Math.round(3.4));           // 반올림  
  
console.log("random: ", Math.random());            // 랜덤
```



배열 관련

기능들!

배열, arr.map



- 기존 배열에 특정 작업을 한 배열을 리턴하는 메소드
- arr.**map**(function(**배열의값**, **인덱스**, **원본배열**) {});

```
const fruits = ["사과", "파인애플", "수박", "포도", "아린지"];

let obj = fruits.map(function (item, index) {
  return {
    id: index,
    name: item,
  };
});

console.log(obj);
```

배열 함수, filter



- 조건에 부합하는 배열 요소만을 반환
- arr.filter(function (매개변수) { return 조건});**

```
let numbers = [1, 2, 3, 4, 5, 6];
let arr;

arr = numbers.filter((num) => num > 3);

console.log(arr);

const words = ['spray', 'limit', 'elite', 'exuberant',
'destruction', 'present'];

const result = words.filter(word => word.length > 6);

console.log(result);
```

배열, arr.reduce



- Reduce 는 가장 많은 기능을 할 수 있습니다!
- arr.reduce(function(누산기, 배열의값, 인덱스, 원본배열), 누산기 초기 값);



```
const arr = [1, 2, 3, 4, 5];
const result = arr.reduce((acc, item, index) => {
  return (acc += item);
}, 0);
console.log(result); // 15

const arr2 = [1, 2, 3, 4, 5];
const result2 = arr2.reduce((acc, item, idx) => {
  return (acc += item);
}, 10);
console.log(result2); // 25
```

배열 함수, includes



- 해당 배열에 지정한 요소가 있는지 확인하는 메소드
- arr.include(요소);

```
let numbers = [1, 2, 3, 4, 5, 6];
console.log(numbers.includes(3));
console.log(numbers.includes(7));

const words = ['spray', 'limit', 'elite', 'exuberant', 'destruction', 'present'];
console.log(words.includes('elite')));
console.log(words.includes('pororo'));
```

배열 함수, find / findIndex



- **find** : 배열에 특정 값이 있는지 찾고 반환
- **findIndex** : 배열에 특정 값이 있는지 찾고 위치를 반환

```
const fruits = ["Apple", "Banana", "Cherry"];

const result1 = fruits.find(item => {
    return /^A/.test(item);
})

const result2 = fruits.findIndex(item => {
    return /^C/.test(item);
})

console.log(result1);
console.log(result2);
```



Object

객체!



Object.assign()

- 객체의 병합에 사용되는 메소드

```
const obj1 = { a: 1, b: 2 };
const obj2 = { b: 3, c: 4 };

const returnObj = Object.assign(obj1, obj2);

console.log(obj1);
console.log(returnObj);
```

```
▶ {a: 1, b: 3, c: 4}
▶ {a: 1, b: 3, c: 4}
```

- 기존에 선언된 객체에 사용하지 않고 가상의 Object 라는 것을 불러와서 사용 → 정적 메소드!



구조 분해 할당

(비구조화 할당)

구조 분해 할당



- 객체 또는 배열의 각각의 값을 분해하여 변수에 넣어 사용하는 방법

```
const user = {  
  id: 1,  
  name: "tetz",  
  email: "xenosign@naver.com",  
};  
  
const { id, name, email, address } = user;  
// 기본값 설정 const { id, name, email, address = "KOREA" } = user;  
// 특정 변수에 넣기 const { id, name: tetz, email, address = "KOREA" } = user;  
  
console.log(id);  
console.log(name);  
console.log(email);  
console.log(address);  
  
const fruits = ["사과", "딸기", "망고", "수박"];  
const [a, b, c, d] = fruits;  
console.log(a, b, c, d);
```



전개 연산자

전개 연산자



- 배열의 값을, 단위로 구분 하여 전개시켜주는 연산자

```
const fruits = ["사과", "바나나", "수박"];
console.log(fruits);
console.log(...fruits);
// console.log("사과", "바나나", "수박");

function conLog(a, b, c) {
  console.log(a, b, c);
}

conLog(fruits[0], fruits[1], fruits[2]);
conLog(...fruits);
```

나머지 연산자(매개 변수일 때)



- 매개 변수에 너무 많은 값이 들어올 때 나머지 연산자를 사용하여 한꺼번에 처리 가능

```
const fruits = ["사과", "바나나", "수박", "망고",
"딸기"];  
  
function conLog(a, b, ...c) {
  console.log(a, b, c);
}  
  
conLog(...fruits);
```

```
const fruits = ["사과", "바나나", "수박", "망고", "딸기"];  
  
function conLog(...rest) {
  rest.forEach((element) => {
    console.log(element);
  });
}  
  
conLog(...fruits);
```

문자열을 사실 진짜 배열로!



```
let string = "apple";
let strToArr = [...string];
console.log(strToArr);

let strToArr2 = str.split("");
console.log(strToArr2);
```

- 전개 연산자(...) 사용으로 “apple” 을 “a”, “p”, “p”, “l”, “e” 로 변환
- 그것을 [] 안에 넣어서 배열로 변환!

옹알이



- <https://school.programmers.co.kr/learn/courses/30/lessons/120956>

<input checked="" type="checkbox"/>	옹알이 (1) 코딩테스트 입문	Lv. 0	8,006명	28%
-------------------------------------	---------------------	-------	--------	-----



```
function solution(babbling) {
  let answer = 0;
  const words = [ 'aya', 'ye', 'woo', 'ma' ];

  const padArr = babbling.map((el) => {
    for (let i = 0; i < words.length; i++) {
      const padStr = '0'.repeat(words[i].length);
      el = el.replace(words[i], padStr);
    }
    return el;
  });

  console.log(padArr);

  const resultArr = padArr.map((el) => el.replace(/0/g, ''));

  console.log(resultArr);
  console.log(resultArr.filter((el) => el === ''));

  answer = resultArr.filter((el) => el === '').length;

  return answer;
}
```



<https://www.acmicpc.net/problem/1000>



```
const fs = require('fs');

const input = fs.readFileSync('/dev/stdin').toString().split(' ');

const A = parseInt(input[0], 10);
const B = parseInt(input[1], 10);

console.log(A + B);
```





```
≡ stdin ×  
dev > ≡ stdin  
1 1 2
```

탐색기 ... JS test.js × ≡ stdin

BJ dev ≡ stdin JS test.js > ...

```
1 const fs = require("fs");
2
3 const input = fs.readFileSync("./dev/stdin").toString().split(" ");
4
5 const A = parseInt(input[0], 10);
6 const B = parseInt(input[1], 10);
7
8 console.log(A + B);
9
```



```
const fs = require("fs");

const input = fs.readFileSync("./dev/stdin");
console.log(input);
```

```
[Running] node "c:\Users\lhs\Desktop\bj\test.js"
<Buffer 31 20 32>
```



```
const fs = require("fs");

const input =
fs.readFileSync("./dev/stdin").toString();
console.log(input);
```

```
[Running] node "c:\Users\lhs\Desktop\bj\test.js"
1 2
```



```
const fs = require("fs");

const input =
fs.readFileSync("./dev/stdin").toString().split(" ");
console.log(input);
```

```
[Running] node "c:\Users\lhs\Desktop\bj\test.js"
[ '1', '2' ]
```



```
const fs = require("fs");

const input = fs.readFileSync("./dev/stdin").toString().split(" ");
console.log(input);

const A = parseInt(input[0], 10);
const B = parseInt(input[1], 10);

console.log(A + B);
```





<https://www.acmicpc.net/problem/1037>

예제 입력 1 복사

```
2  
4 2
```

예제 입력 2 복사

```
1  
2
```

예제 입력 3 복사

```
6  
3 4 2 12 6 8
```





```
stdin2  x  JS test2.js
dev > stdin2
1 14
2 14 26456 2 28 13228 3307 7 23149 8 6614 46298 56 4 92596
```

```
const fs = require("fs");

const input =
fs.readFileSync("./dev/stdin2").toString().split("\n");
console.log(input);
```

```
[Running] node "c:\Users\lhs\Desktop\bj\test2.js"
[ '14\r', '14 26456 2 28 13228 3307 7 23149 8 6614 46298 56 4 92596' ]
185192
```



New Line

CRLF / LF / CR

CR : Carriage Return (\r)

LF : Line Feed (\n)

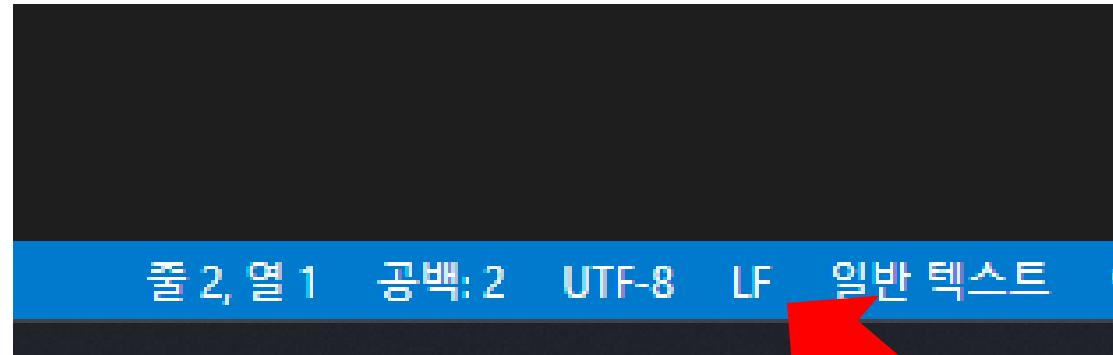
해당 용어(CR, LF)와 이 용어들의 조합(CRLF)은 새로운 줄 (New line) 으로 바꾸는 방식을 의미한다.

CR 과 LF 는 타자기 시절 부터 줄바꿈을 위해 사용하던 방식인데 각각의 의미는 다음과 같다.

CR : 현재 커서를 줄 올림 없이 가장 앞으로 옮기는 등작

LF : 커서는 그 자리에 그대로 둔 상황에서 종이만 한 줄 올려 줄을 바꾸는 등작

대표적으로 Microsoft 사의 Windows 는 CRLF (\r\n) 을 기본으로 사용하는 반면
Unix/Linux 에서는 LF (\n) 만으로 줄바꿈을 하고 있다.



```
[Running] node "c:\Users\lhs\Desktop\bj\test2.js"
[ '14', '14 26456 2 28 13228 3307 7 23149 8 6614 46298 56 4 92596' ]
185192
```



```
const fs = require("fs");

const input =
fs.readFileSync("./dev/stdin2").toString().split("\n");
// 배열 값을 " " 로 나눈 배열로 변경
const arr1 = input[1].split(" ");
const arr2 = input[1].split(" ");

// 문자를 다시 10진법 숫자로 전환
const numArr1 = arr1.map((el) => parseInt(el, 10));
const numArr2 = arr2.map((el) => parseInt(el, 10));
```





지금 시작합니다



2018 러시아 월드컵 조추첨식

POT 1

RUS
GER
BRA
POR
ARG
BEL
POL
FRA

SBS

GROUP A

A1
A2
A3
A4

GROUP B

B1
B2
B3
B4

GROUP C

C1
C2
C3
C4

GROUP D

D1
D2
D3
D4

GROUP E

E1
E2
E3
E4

GROUP F

F1
F2
F3
F4

GROUP G

G1
G2
G3
G4

GROUP H

H1
H2
H3
H4

2018 WORLD CUP RUSSIA 2018

프론트 프로젝트 조 추첨 결과



mamihlapinatapai^[1] 마밀라피나타파이

칠레 남부 티에라델푸에고 지역의 "야간"(Yaghan)족 원주민들
이 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 조용하면
서도 긴급하게 오가는 미묘한 눈빛.



mamihlapinatapai^[1]

마밀라피나타파이

칠레 남부 티에라델푸에고 지역의 "야간"(Yaghan)족 원주민들이 이 쓴 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 (██████████)

조장하실분?

ㅋㅋ 2016.01.05 22:10 (██████████)

미친ㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋ

빵 터짐 2016.01.05 23:48 (██████████)

████ 첫 댓글이 하드캐리

○○ 2016.01.08 04:13 (██████████)

ㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋ



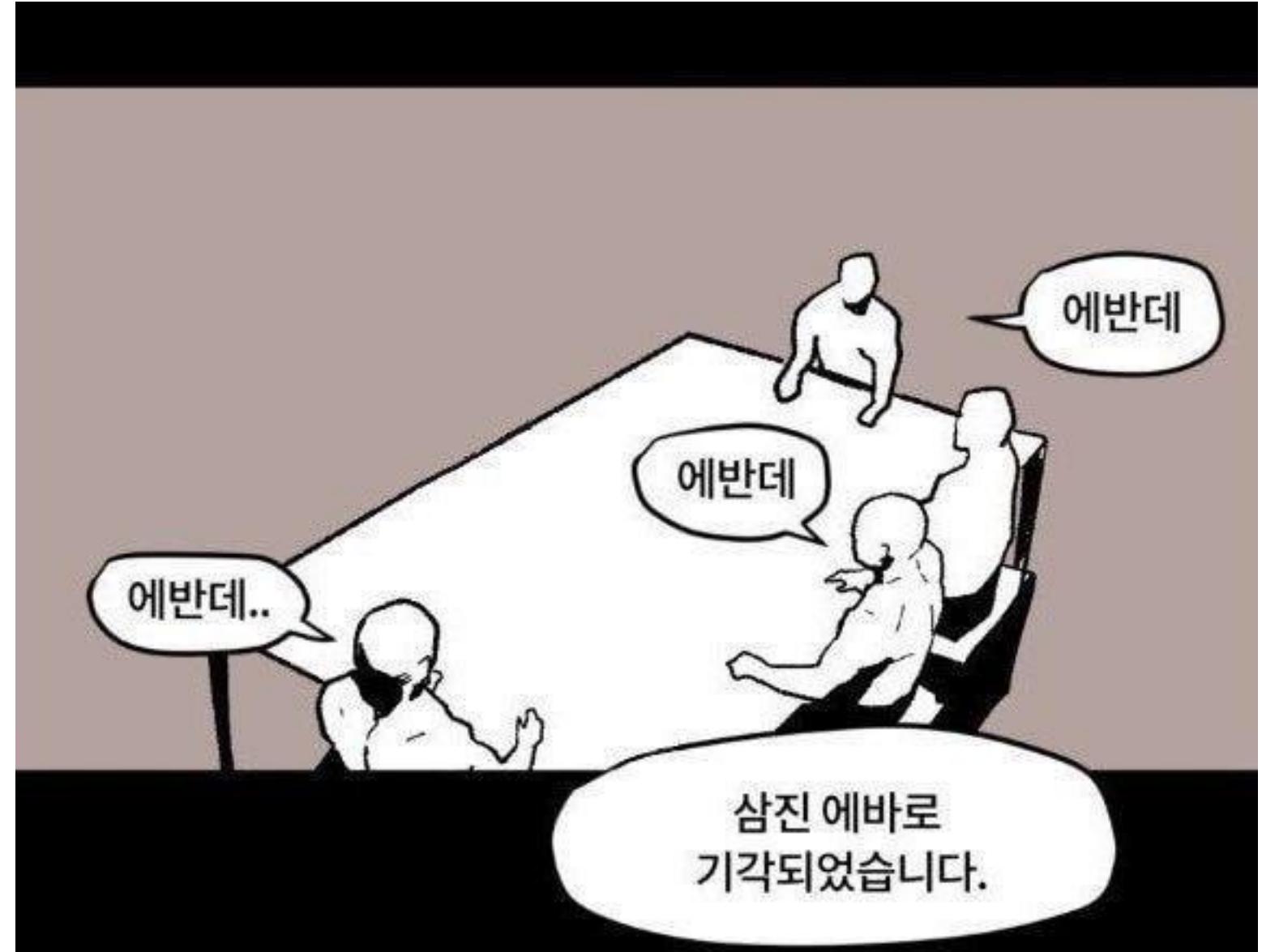
회의를 마치고 온 뒤
그의 모습에는

회의를 시작하지

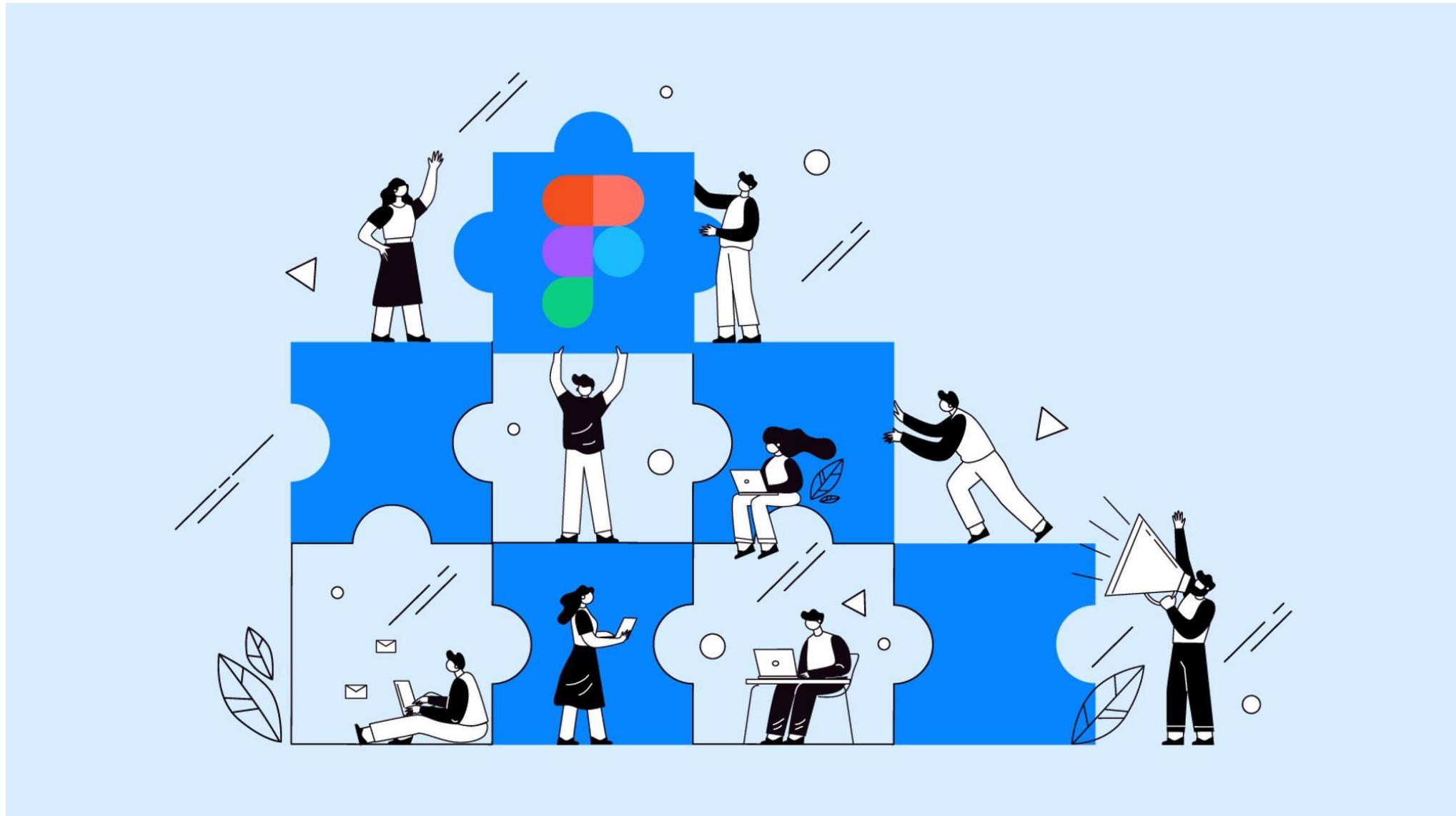


흔한회의실풍경_부장님_무서우니까_얼굴_좀_펴세요.jpg

회의를 마치고 온 뒤
그의 모습에는







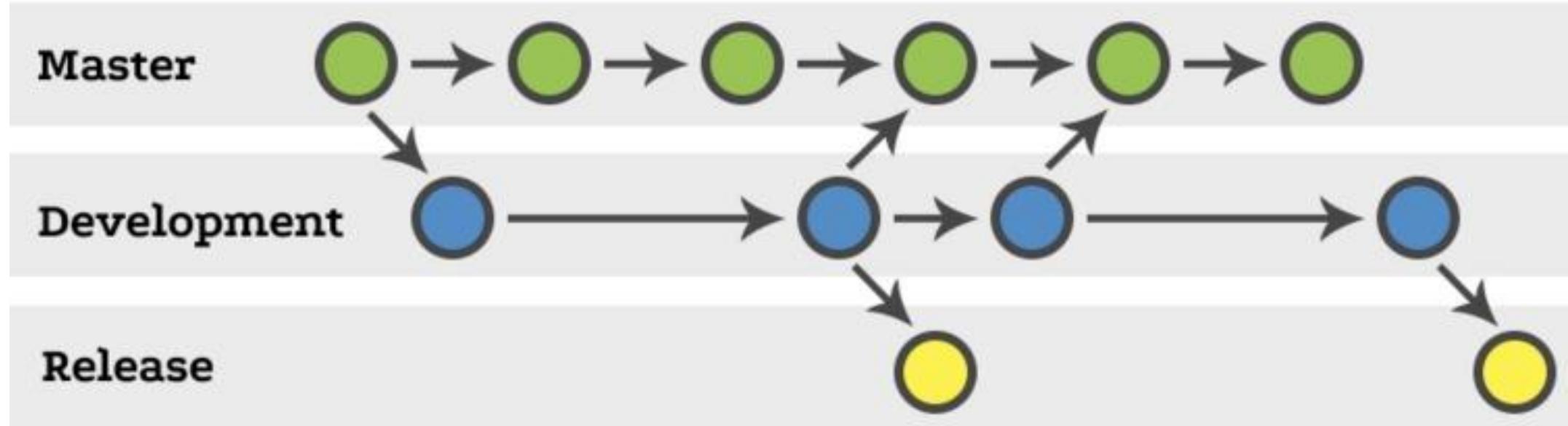


Git branch

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



새로운 git 생성

새로운 깃 저장소 만들기!



- 원하는 폴더를 만들어 주세요!
- 해당 폴더를 Vscode 로 열어 주세요!
- 터미널을 켜고!
- **Git init**
- **Index.html** 파일 만들고 기초 코드 넣기!
- **Git add . → git commit -m "fc"**

새로운 것 **브랜치** 만들기!



- **Git branch css**
- **Git branch**
 - 만들어진 브랜치 확인
- **Git switch css / git checkout css**
- **Git branch**
 - 현재 브랜치 위치 확인

새로운 것 **브랜치**에 파일 추가하기!



- Style.css 파일 추가!
- Index.html 파일에 링크도 걸기!
- 해당 내용 커밋하기!
- **Git switch master / Git checkout master**
- Master 브랜치에는 아까 변경한 사항이 전혀 반영 되어 있지 않죠?
- 이런 식으로, 각자가 담당한 부분을 작업을 하고 작업이 완료 되면 그 때에 합치면 됩니다!



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 만들기!



- 먼저 master의 index.html 페이지 제목을 Master로 변경해 봅시다.
- 깃 커밋하기!
- Git switch css / git checkout css
- Css 브랜치의 index.html 파일의 제목을 CSS로 변경해 봅시다
- 깃 커밋하기!
- Git switch master / git checkout master



Conflict 상황 만들기!

- Git merge css

```
tetz@DESKTOP-P7Q40LL MINGW64 ~/Desktop/git (master)
$ git merge css
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.
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"> You, 2시간 전 • com
</head>
```

Conflict 상황 해결하기!



```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
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"> You, 2시간 전 • com
</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!



- 여기서는 무조건 Vscode 를 사용하는 편이 정신 건강에 좋습니다
- Git 으로 해결하려 들면, vim으로 코드 수정 해야 하거든요 ㅎㅎㅎ

```
<meta name="viewport" content="width=device-width, initial-scale=1.
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">      You, 2시간 전 • com
</head>
```

```
tecmint@TecMint ~
VIM - Vi IMproved
version 8.0.3
by Bram Moolenaar et al.
Modified by pkg-vim-maintainers@lists.alioth.debian.org
Vim is open source and freely distributable

Help poor children in Uganda!
type :help iccf<Enter>      for information

type :q<Enter>                to exit
type :help<Enter> or <F1>    for on-line help
type :help version8<Enter>   for version info

0,0-1          All
```



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"> You, 2시간 전 • com
</head>
```

- **Accept Current Change** 를 누르면 master의 코드를 보존 합니다

```
<meta name="viewport" content="width=device-width, initial-scale=1.
<title>Master</title>
<link rel="stylesheet" href="./style.css">
</head>
```



Conflict 상황 해결하기!

```
<meta name="viewport" content="width=device-width, initial-scale=1.

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"> You, 2시간 전 • com
</head>
```

- **Accept Incoming Change** 를 누르면 css 브랜치의 코드를 보존

```
<meta name="viewport" content="width=device-width, initial-scale=1.

<title>CSS</title>
<link rel="stylesheet" href="./style.css">
```



Conflict 상황 해결하기!

```
<meta name="viewport" content="width=device-width, initial-scale=1.">
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"> You, 2시간 전 • com
</head>
```

- **Accept Both Change** 를 누르면 master와 css의 코드를 전부 보존합니다!

```
<title>Master</title>
<title>CSS</title>
<link rel="stylesheet" href="./style.css">
```

Conflict 상황 해결하기!



- Compare Changes를 누르면 코드 비교 에디터가 켜집니다!

The screenshot shows a code editor with two panes. The left pane displays the current version of the file, and the right pane displays the incoming changes. A red bar highlights the conflict area around line 8, which contains the title element. The incoming changes suggest changing the title from 'Master' to 'CSS'. The code editor interface includes tabs for 'index.html !' and 'index.html: Current Changes ↔ Incoming Changes', and navigation arrows at the top right.

```
index.html !
index.html: Current Changes ↔ Incoming Changes

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>Master</title>
9   <link rel="stylesheet" href="./style.css">
10 </head>
11
12 <body>
13
14 </body>
15
16 </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>CSS</title>
9   <link rel="stylesheet" href="./style.css">
10 </head>
11
12 <body>
13
14 </body>
15
16 </html>
```

- Conflict 된 코드 량이 많으면 해당 에디터가 도움이 됩니다!

Conflict 상황 해결하기!



- 일단 원하는 방향으로 conflict를 해결해 줍니다!
- 우리는 제목을 Master CSS로 변경해 봅시다!

```
meta name="viewport" content="width=device-width, initial-scale=1">
<title>Master CSS</title>
<link rel="stylesheet" href=".style.css">
</head>
```

Conflict 상황 해결하기!



```
tetz@DESKTOP-P7Q40LL MINGW64 ~/Desktop/git (master|MERGING)
$
```

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

```
tetz@DESKTOP-P7Q40LL 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

종합 실습

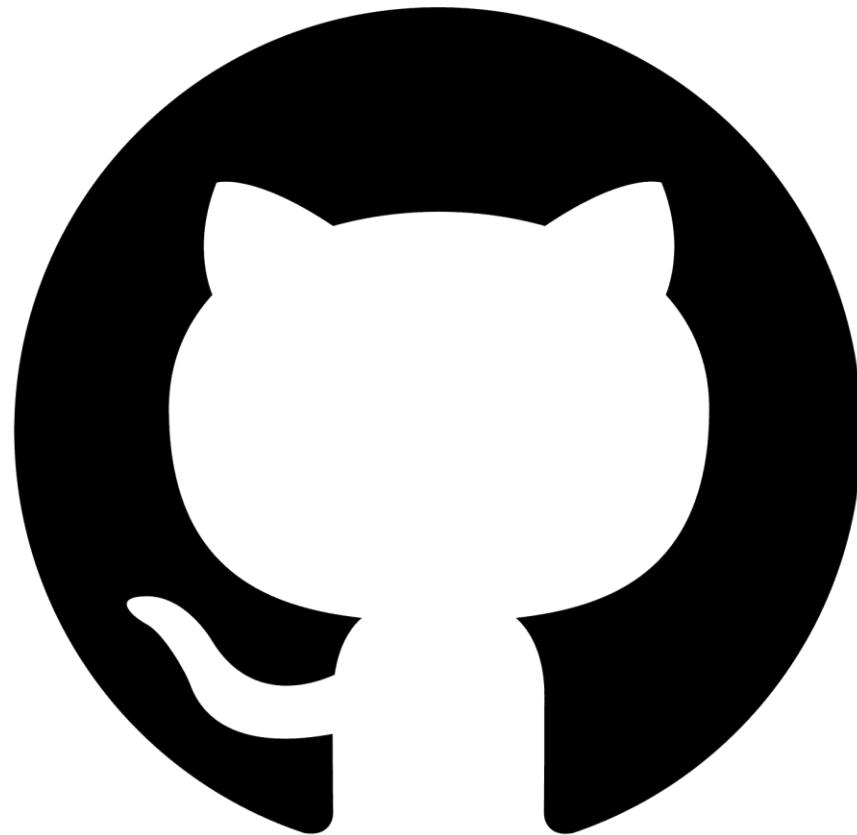


- JS 라는 브랜치를 만듭니다!
- JS 라는 브랜치로 이동 후, main.js 라는 파일을 만들고 alert 창으로 “!!”라는 문구를 띄우는 코드를 삽입!
- Index.html 파일에 해당 js 파일을 연결해 줍니다!
- JS 브랜치에서 변경 사항 커밋!
- JS 브랜치의 내용을 Master 에 Merge!

종합 실습



- Master의 main.js에 가서 alert 창의 문구를 “??”로 변경!
- 변경 내용 커밋
- JS 브랜치로 이동하여, alert 창의 문구를 “!?”로 변경!
- 변경 내용 커밋
- Master에서 JS 브랜치를 Merge!
- Conflict 상황 해결하고 Merge 완료 하기!



GitHub

online



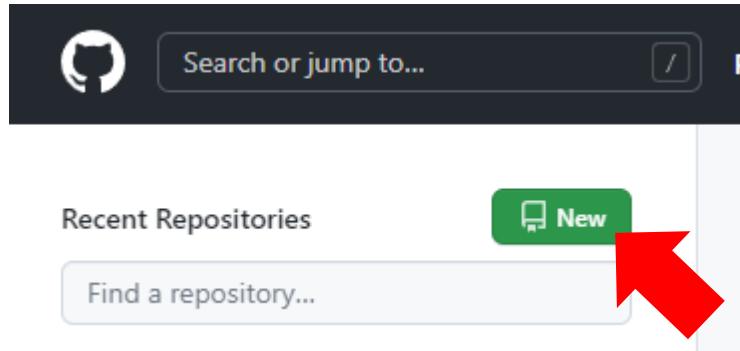


조장 Part

온라인 협업을 시작해 봅시다!



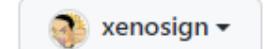
- 조장 분은 깃허브에 협업 연습을 할 Repo를 생성해 봅시다!



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 *

portfolio2



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!



The screenshot shows the GitHub Desktop application interface. On the left, a sidebar displays repository icons and names: 'xenosign/panorama', 'xenosign/poll-project', 'xenosign/portfolio', 'xenosign/portfolio2' (which is highlighted with a red arrow), 'xenosign/refactor', 'xenosign/smartboard', and 'xenosign/ssac_lhi react'. The 'File' menu is open at the top, showing options like 'New repository...', 'Add local repository...', 'Clone repository...' (selected and highlighted with a blue border), 'Options...', and 'Exit'. In the center, a 'Clone a repository' dialog box is open, showing tabs for 'GitHub.com' (selected), 'GitHub Enterprise', and 'URL'. It includes a search bar labeled 'Filter your repositories' and a list of repositories. At the bottom of the dialog, there's a 'Local path' field containing 'D:\git2' and a 'Choose...' button, followed by 'Clone' and 'Cancel' buttons. Three red arrows point from the text labels below to the corresponding elements in the screenshot: one to the 'Clone repository...' menu item, one to the 'Choose...' button, and one to the 'Clone' button.

File Edit View Repository E

New repository... Ctrl+N

Add local repository... Ctrl+O

Clone repository... Ctrl+Shift+O

Options... Ctrl+,

Exit Alt+F4

Clone a repository

x GitHub.com GitHub Enterprise URL

Filter your repositories

xenosign/panorama

xenosign/poll-project

xenosign/portfolio

xenosign/portfolio2

xenosign/refactor

xenosign/smartboard

xenosign/ssac_lhi react

Local path

D:\git2 Choose...

Clone Cancel

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



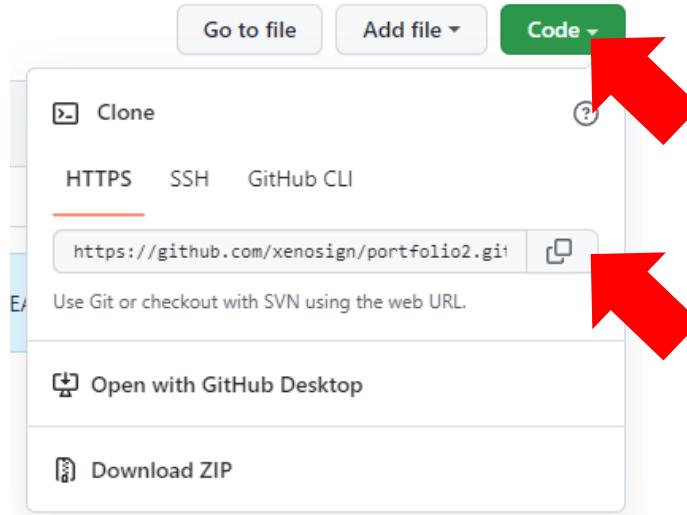


팀원 Part

팀원들은, 해당 repo를 clone 합시다!



- 조장이 만든 git 주소를 받으세요! 해당 주소로 접속!

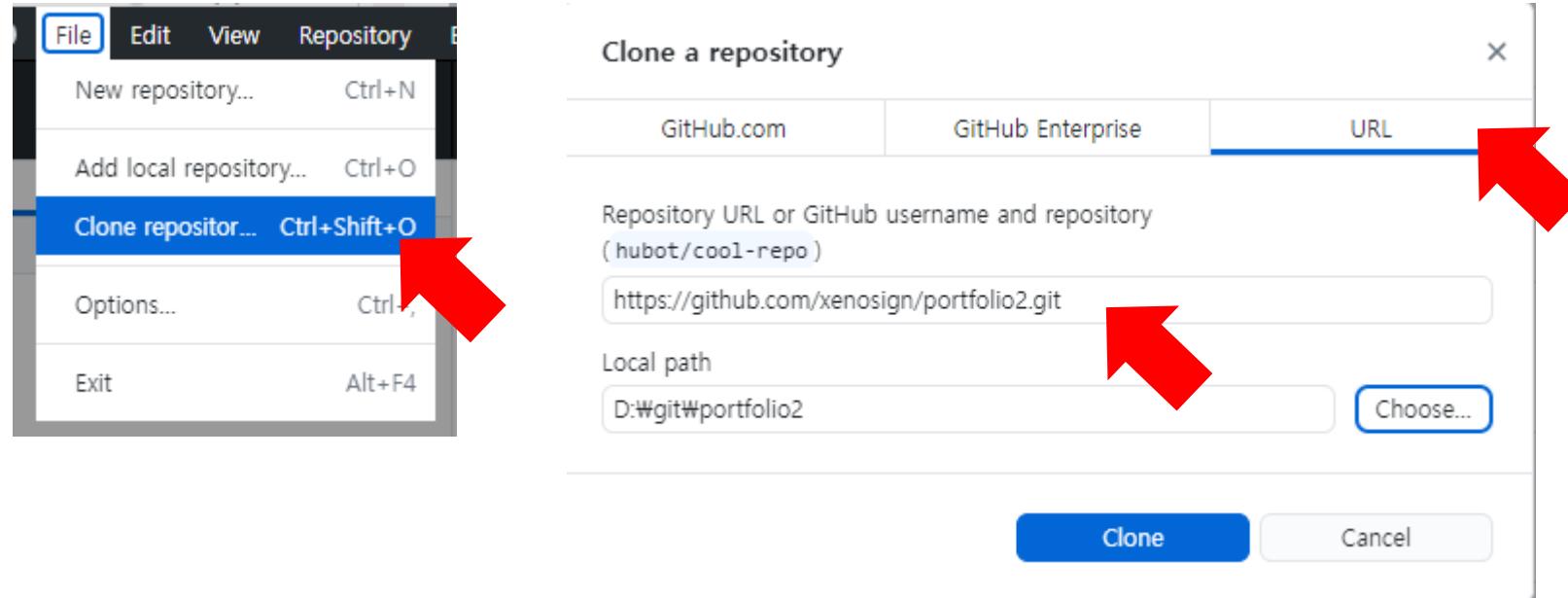


- **Git clone** 복사한 주소

팀원들은, 해당 repo를 clone 합시다!



- 조장이 만든 git 주소를 받으세요! 해당 주소로 접속!



팀원 Part, 클론한 Repo에서!



- 새로운 브랜치 생성!
- Git branch css
- **Git switch css / git checkout css**
- Style_본인이름.css 파일 생성!
- Reset 코드 추가!
- 커밋!
- **Git push --all**



Error

팀원 Part, Error 발생!



- 권한이 없기 때문입니다!
- 아무리 Public Repo라고 해도 아무나 push하면 안되겠죠?
- Clone은 아무나 가능하지만, push는 혀용된 인원만 가능합니다!
- 그럼 팀원을 초대해 봄시다!



조장 Part

조장 Part, 팀원 초대!



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

Manage access

You haven't invited any collaborators yet

Add people

The screenshot shows a GitHub repository settings page for 'xenosign / portfolio2'. A red arrow points to the 'Collaborators' button under the 'Access' section. Another red arrow points to the 'Add people' button under the 'Manage access' section. The 'General' tab is selected in the sidebar.

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

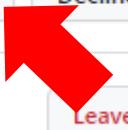
Sign out

Organizations

New organization

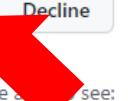
 **portfolio2**  Invitation expires in 7 days. 

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

[Join](#) [Decline](#) 
[Leave](#)

 + 

xenosign invited you to collaborate

[Accept invitation](#) [Decline](#) 

 Owners of portfolio2 will be able to see:

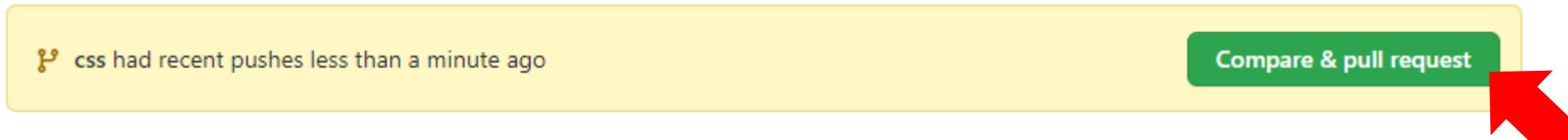
- Your public profile information
- Certain activity within this repository
- Country of request origin
- Your access level for this repository
- Your IP address

Is this user sending spam or malicious content?
[Block xenosign](#)

팀원 Part, 다시 Push!



- Git push --all
- 그리고 깃허브로 가봅시다!



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

style.css 파일 추가 및 Reset 작업!

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

Create pull request ▾

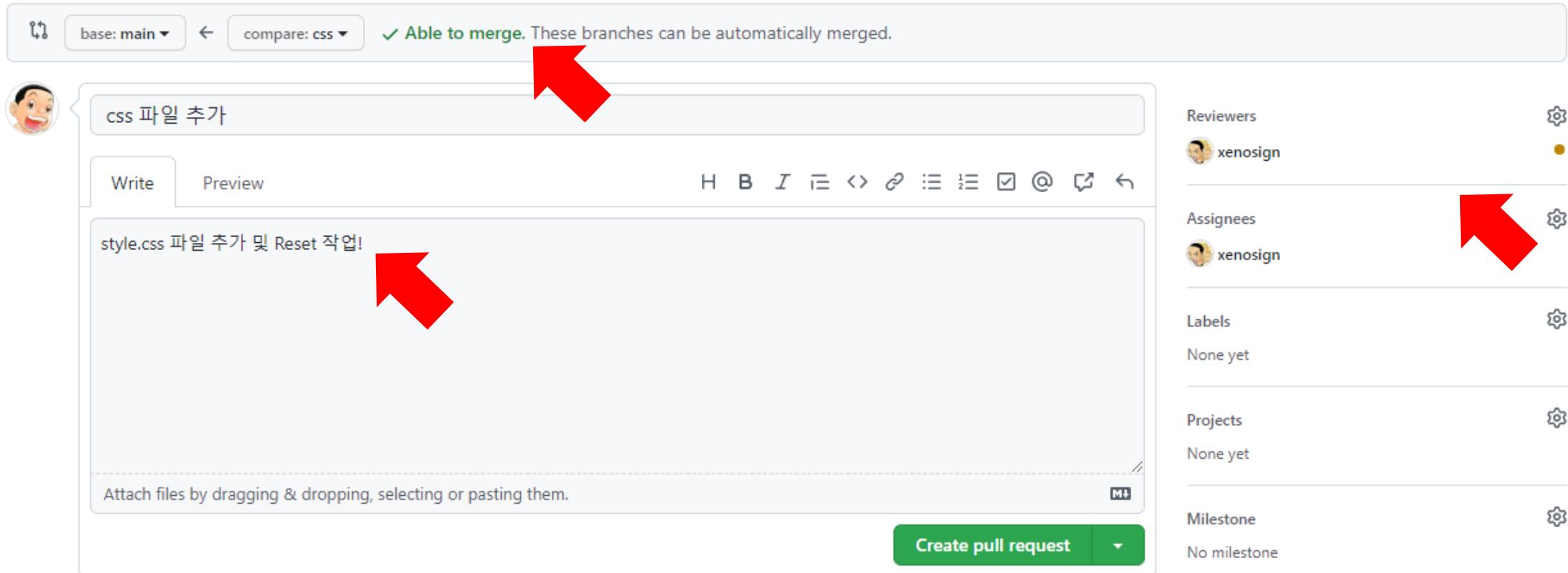
Reviewers: xenosign

Assignees: xenosign

Labels: None yet

Projects: None yet

Milestone: No milestone



팀원 Part, 깃헙에서 Merge!



- **Reviewers**

- 말 그대로 해당 코드를 리뷰해야 할 사람들을 추가 할 수 있습니다!

- **Assignees**

- 해당 코드의 Merge 여부를 결정할 사람을 지정합니다!
- 보통의 팀에서는 팀장급(또는 프로젝트 리더)이 Assignees로 등록 되며, Reviewers는 팀원들을 포함합니다!
- 그리고 merge 룰 을 정할 수 있는데요.
 - Reviewers 의 승인이 있어야만 Merge가 가능하게 등 다양하게 정할 수 있습니다.



CSS 파일 추가 #1

Edit Code ▾

Open

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

Conversation 0

-o Commits 1

Checks 0

Files changed 2

+4 -0



TetzSpreatics commented now

Collaborator ...

style.css 파일 추가 및 Reset 작업!

-o 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

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](#).



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

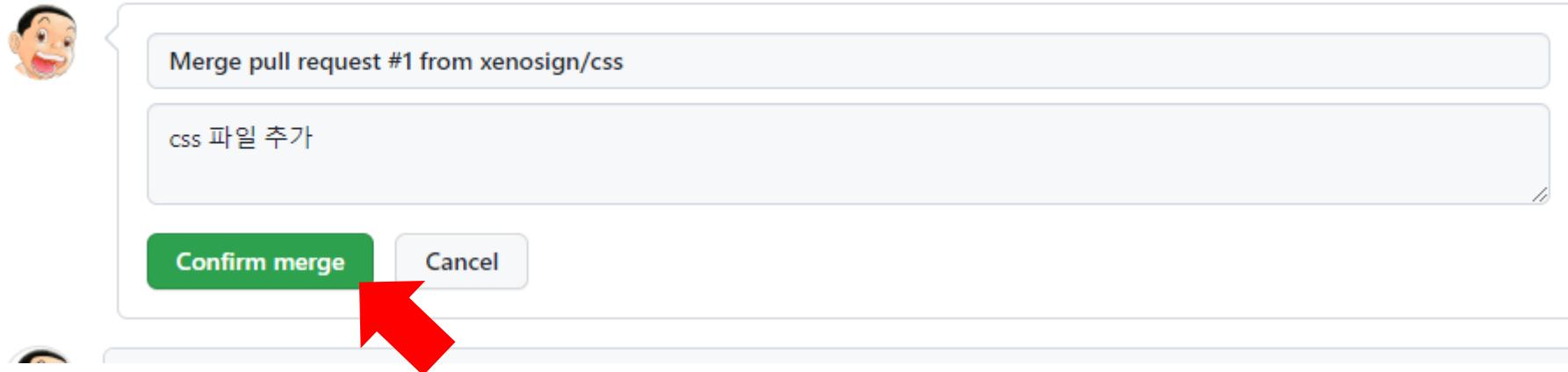
Customize

팀원 Part, 깃헙에서 Merge!



- 현재는 룰을 정하지 않아서 바로 Merge가 가능한 상황입니다!

Add more commits by pushing to the `css` branch on [xenosign/portfolio2](#).



팀원 Part, 깃헙에서 Merge!



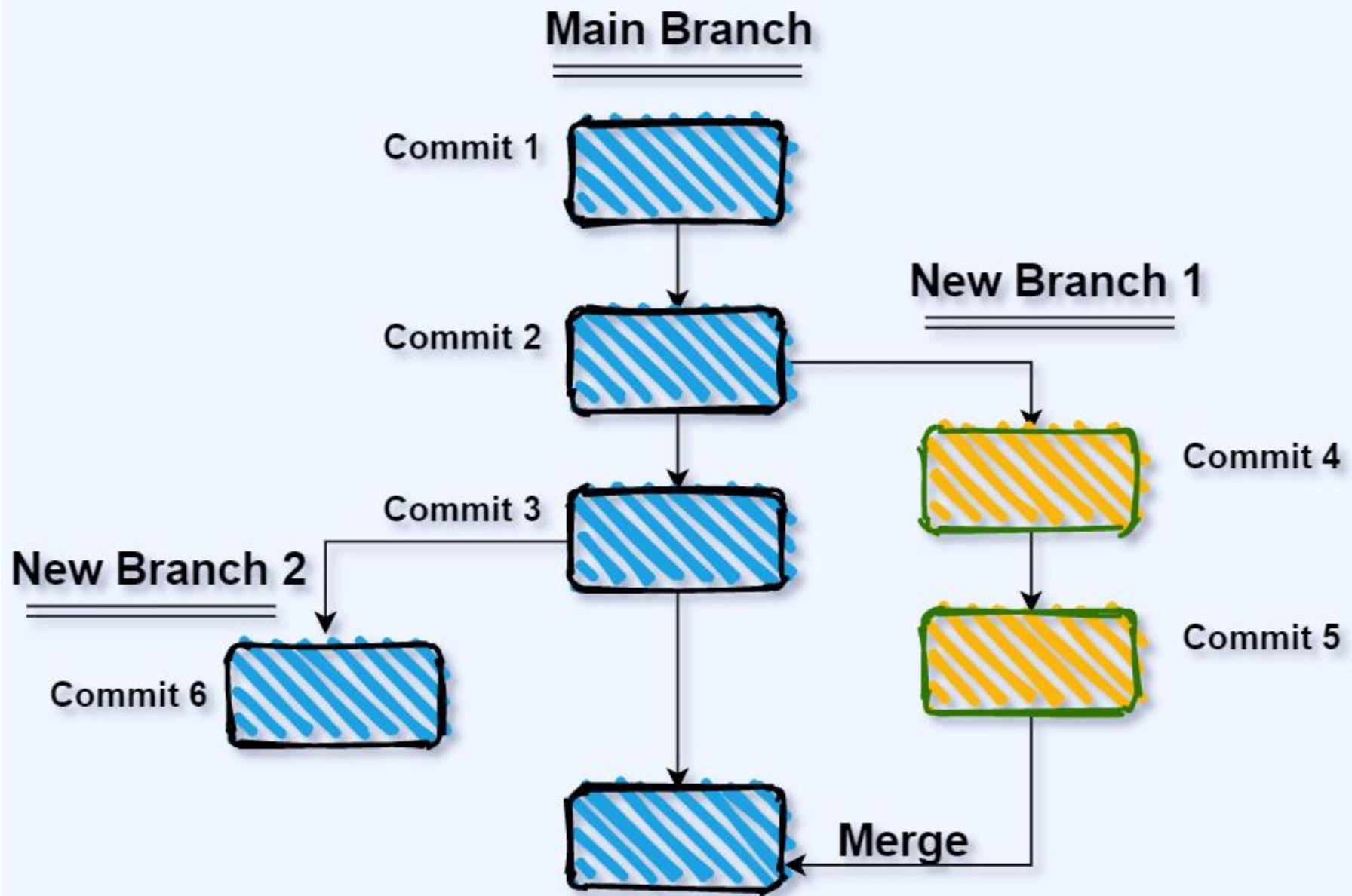
- Merge가 완료되어 main 브랜치에 style.css 파일이 생성

A screenshot of a GitHub repository interface. At the top, there are buttons for 'main' (selected), '2 branches', '0 tags', 'Go to file', 'Add file', and 'Code'. Below this is a list of commits:

Author	Commit Message	Date	Commits
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

At the bottom of the list, there is a red arrow pointing to the 'style.css' entry. Below the list is a light blue footer bar with the text 'Help people interested in this repository understand your project by adding a README.' and a green 'Add a README' button.

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



팀원 Part, pull 받기!



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

팀원 Part, pull 받기!



Screenshot of a GitHub desktop application interface showing a repository named "portfolio2" with the main branch selected. The "Changes" tab is active, showing "0 changed files". A central message says "No local changes". Below it, a suggestion box highlights "Pull 2 commits from the origin remote" with a "Pull origin" button, which is highlighted by a large red arrow. Other options include opening the repository in an external editor, viewing files in Explorer, and opening the GitHub page.

File Edit View Repository Branch Help

Current repository portfolio2 Current branch main Pull origin Last fetched 1 minute ago 2 ↓

Changes History 0 changed files

No local changes

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

Pull 2 commits from the origin remote

The current branch (main) has commits on GitHub that do not exist on your machine.

Always available in the toolbar when there are remote changes or `Ctrl Shift P`

Pull origin

Open the repository in your external editor

Select your editor in Options

Repository menu or `Ctrl Shift A`

Open in Visual Studio Code

View the files of your repository in Explorer

Repository menu or `Ctrl Shift F`

Show in Explorer

Open the repository page on GitHub in your browser

Repository menu or `Ctrl Shift G`

View on GitHub

Summary (required)

Description

R+

Commit to main



조장 Part



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!



The screenshot shows the GitHub repository settings page for a repository named 'jjang'. The left sidebar includes sections for Branches, Tags, Actions, Webhooks, Environments, Pages, Security, Code security and analysis, Deploy keys, Secrets, Integrations, GitHub apps, and Email notifications. The 'Branches' section is currently selected. A modal window titled 'Protect matching branches' is open, showing a 'Branch name pattern' field containing 'main' with a red arrow pointing to it. Below this, under 'Protect matching branches', there are two checked options: 'Require a pull request before merging' and 'Require approvals'. The 'Require approvals' section includes a dropdown menu set to '1' with a red arrow pointing to it, and a list of numbers from 1 to 6 below it. A message at the top of the modal states: 'Your GitHub Free plan can only enforce rules on its public repository. If you have private repositories, you can protect them by upgrading to a GitHub Pro or GitHub Team plan.' The URL in the browser bar is 'https://github.com/jjang/jjang/settings/branches'.

- 최소 한명의 Reviewer가 Approve 리뷰를 남겨야만 머지가 가능하도록 설정!



팀원 Part

팀원 Part, 새로운 브랜치 생성!



- **Git branch js**
- **Git switch js / git checkout js**
- **main_본인이름.js** 파일 생성 → alert 코드 삽입
- 개인적으로 여기서 부터는 **github desktop** 사용을 추천 합니다!
- 커밋 → 푸쉬

팀원 Part, pull request 수행



js had recent pushes 1 minute ago

Compare & pull request

main ▾ 3 branches 0 tags

Go to file Add file ▾ Code ▾

TetzSpreatics Merge pull request #1 from xenosign/css ...	6e6d956 18 minutes ago	3 commits
index.html	css 파일 추가	35 minutes ago
style.css	css 파일 추가	35 minutes ago

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

Add a README

팀원 Part, pull request 수행



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: js ▾ ✓ Able to merge. These branches can be automatically merged.

JS 파일 추가

Write Preview H B I ᄒ < ᄑ ᄒ ᄒ ᄒ @ ↵

main.js 파일 생성 및 링크 완료!

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

Create pull request ▾

Reviewers
Suggestions
xenosign Request
At least 1 approving review is required to merge this pull request.

Assignees
xenosign

Labels
None yet

Projects
None yet

팀원 Part, pull request 수행



JS 파일 추가 #2

Edit < Code ▾

Open TetzSpreatics wants to merge 1 commit into `main` from `js`

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

TetzSpreatics commented now
main.js 파일 생성 및 링크 완료!

JS 파일 추가 b97c969

TetzSpreatics assigned xenosign now

Add more commits by pushing to the `js` branch on [xenosign/portfolio2](#).

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

Merging is blocked
Merging can be performed automatically with 1 approving review.

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

Reviewers: xenosign [Request](#)
Suggestions: xenosign
Assignees: xenosign
Labels: None yet
Projects: None yet
Milestone: No milestone



조장 Part

조장 Part, Review 수행



Screenshot of a GitHub repository page for `xenosign/portfolio2` (Public).

The navigation bar includes: Code, Issues, Pull requests (1), Actions, Projects, Wiki, Security, Insights, and Settings. A red arrow points to the 'Pull requests' tab.

Below the navigation bar are filters: Filters (dropdown), a search bar containing 'is:pr is:open', Labels (9), Milestones (0), and a green 'New pull request' button.

The pull request list shows two items:

- 1 Open: JS 파일 추가 (by TetzSpreatic, 2 minutes ago, Review required)
- 1 Closed

Below the list are sorting and filtering options: Author (dropdown), Label (dropdown), Projects (dropdown), Milestones (dropdown), Reviews (dropdown), Assignee (dropdown), and Sort (dropdown). A red arrow points to the first pull request item.

조장 Part, Review 수행



This pull request is waiting on your review.

Add your review

JS 파일 추가 #2

[Open](#) TetzSpreatics wants to merge 1 commit into `main` from `js`

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

TetzSpreatics commented 4 minutes ago
main.js 파일 생성 및 링크 완료!

JS 파일 추가 b97c969

TetzSpreatics assigned xenosign 4 minutes ago

xenosign self-requested a review 31 seconds ago

Reviewers: xenosign (Approved)

At least 1 approving review is required to merge this pull request.

Still in progress? Convert to draft

Assignees: xenosign

Labels

조장 Part, Review 수행



JS 파일 추가 #2

Edit <> Code ▾

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

Conversation 1 Commits 1 Checks 0 Files changed 2

Changes from all commits File filter Conversations

Filter changed files

index.html

```
@@ -7,6 +7,7 @@
7    7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8    8      <title>Document</title>
9    9      <link rel="stylesheet" href="style.css">
10   10     +     <script src="main.js"></script>
11   11     </head>
12   12
13   13     <body>
```

main.js

```
... ...
1   + alert("!!"); 
```

Finish your review

Write Preview

승인합니다!

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

Comment
Submit general feedback without explicit approval.

Approve
Submit feedback to approve merging these changes.

Request change
Submit feedback that must be addressed before merging.

Submit review

Submit review

조장 Part, Merge 승인!



Add more commits by pushing to the `js` branch on [xenosign/portfolio2](#).

Changes approved
1 approving review by reviewers with write access. [Learn more](#).

✓ 1 approval

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](#).

Add more commits by pushing to the `js` branch on [xenosign/portfolio2](#).

Merge pull request #2 from [xenosign/js](#)

JS 파일 추가

Confirm merge Cancel

조장 Part, Merge 승인!



main ▾ 3 branches 0 tags

Go to file Add file ▾ Code ▾

 xenosign	Merge pull request #2 from xenosign/js ...	ce63f45 19 seconds ago	5 commits
	index.html	JS 파일 추가	11 minutes ago
	main.js	JS 파일 추가	11 minutes ago
	style.css	css 파일 추가	44 minutes ago



팀원 Part

팀원 Part, pull 받기!



Screenshot of a GitHub desktop application interface showing a repository named "portfolio2" with the main branch selected. The "Changes" tab is active, showing "0 changed files". A central message says "No local changes". Below it, a suggestion box highlights "Pull 2 commits from the origin remote" with a "Pull origin" button, which is highlighted by a large red arrow. Other options include opening the repository in an external editor, viewing files in Explorer, and opening the GitHub page.

File Edit View Repository Branch Help

Current repository portfolio2 Current branch main Pull origin Last fetched 1 minute ago 2 ↓

Changes History 0 changed files

No local changes

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

Pull 2 commits from the origin remote
The current branch (main) has commits on GitHub that do not exist on your machine.
Always available in the toolbar when there are remote changes or `Ctrl Shift P`

Pull origin

Open the repository in your external editor
Select your editor in Options
Repository menu or `Ctrl Shift A`

Open in Visual Studio Code

View the files of your repository in Explorer
Repository menu or `Ctrl Shift F`

Show in Explorer

Open the repository page on GitHub in your browser
Repository menu or `Ctrl Shift G`

View on GitHub

Summary (required)

Description

R+

Commit to main



Conflict



조장 Part



main ▾ portfolio2 / index.html

Go to file

...

xenosign Update index.html

Latest commit a4ac843 now ⏺ History

1 contributor

17 lines (13 sloc) | 340 Bytes

Raw Blame ⌂ ⌂ ⌂

```
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>test</title>
9      <link rel="stylesheet" href="style.css">
10     <script src="script.js"></script>
11 </head>
12
13 <body>
14
15 </body>
16
17 </html>
```





Commit changes

Create index.html

Add an optional extended description...

-o Commit directly to the `main` branch.

↗ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit changes

Cancel





팀원 Part

팀원 Part, 새로운 브랜치 생성!



- Git branch conflict
- Git switch conflict / git checkout conflict
- Index.html의 제목을 conflict로 수정!
- 개인적으로 여기서 부터는 github desktop 사용을 추천 합니다!
- 커밋 → 푸쉬

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#)

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

Leave a comment

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

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

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
--------------------------	------------

```
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
```

A red arrow points from the bottom right towards the merge conflict markers in the code editor. The conflict markers are represented by yellow boxes containing the text '<===== conflict' and '===== >>>>> main'. The code editor interface shows two versions of the file: 'conflict' and 'main', which have been merged into a single file.



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

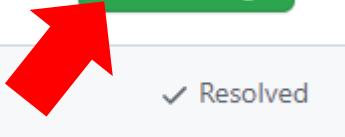


Mark as resolved



Commit merge

✓ Resolved





Add more commits by pushing to the `conflict` branch on [xenosign/portfolio2](#).

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

 1 pending reviewer ▾

 **x Merging is blocked** Merging can be performed automatically with 1 approving review.

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



조장 Part



xenosign / portfolio2 Public

Code Issues Pull requests 1 Actions Projects Wiki Security Insights Settings

Label issues and pull requests for new contributors Dismiss

Now, GitHub will help potential first-time contributors discover issues labeled with good first issue

Filters is:pr is:open Labels 9 Milestones 0 New pull request

1 Open ✓ 2 Closed

Update index.html #3 opened 3 minutes ago by  · Review required



TetzSpreatics requested your review on this pull request.

[Add your review](#)

Update index.html #3

[Edit](#)

[Code](#) ▾



[Open](#) TetzSpreatics wants to merge 2 commits into [main](#) from [conflict](#)

Conversation 0

Commits 2

Checks 0

Files changed 1

+1 -1



0 / 1 files viewed (i) Review changes ▾

Finish your review

Write Preview H B I ≡ < > ⊕ ≡ 1≡ ✓ @ ✉ ↶

Leave a comment

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

Comment
Submit general feedback without explicit approval.

Approve
Submit feedback and approve merging these changes.

Request changes
Submit feedback that must be addressed before merging.

Submit review

1g



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

Changes approved
1 approving review by reviewers with write access. [Learn more](#).

1 approval

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](#).

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

Merge pull request #3 from xenosign/conflict

Update index.html

Confirm merge **Cancel**



팀원 Part

File Edit View Repository Branch Help

Current repository portfolio2

Current branch conflict

Pull origin Last fetched just now 2 ↓

Changes History

0 changed files

No local changes

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

Pull 2 commits from the origin remote
The current branch (`conflict`) has commits on GitHub that do not exist on your machine.
Always available in the toolbar when there are remote changes or `Ctrl Shift P`

Pull origin 

Open the repository in your external editor
Select your editor in Options
Repository menu or `Ctrl Shift A`

Open in Visual Studio Code

View the files of your repository in Explorer
Repository menu or `Ctrl Shift F`

Show in Explorer

Open the repository page on GitHub in your browser
Repository menu or `Ctrl Shift G`

View on GitHub

Summary (required) 

Description

+

Commit to conflict





나마지 팀원들도

컨플릭트

상황 체험하기



성공적인 프로젝트 진행을 위해

반드시 지켜야 할 3가지





1. 일단 pull 받고 시작하기
2. 일단 pull 받고 시작하기
3. 일단 pull 받고 시작하기



TV TOKYO

해치웠나...!



<https://coinpan.com> 코인판 on 2018-01-09



Sass



WHAT?

Sass



Syntactically
Awesome
Style Sheets



css





CSS





css



```
css > main.css > ...
1 .inr{width:1200px; margin:0 auto;}
2
3 .area_visual{position:relative; width:100%; height: calc(100vh - 100px); background-size: cover; background-position:50%;}
4 .area_visual .inr{position:relative; height: calc(100vh - 100px); margin:0 auto;}
5 .area_visual .inr ul{position:absolute; right:0; bottom:20%; width:600px;}
6 .area_visual .inr ul li{position:relative; margin-right:300px; width:300px; height:205px; background: #08495d; color: #fff; font-size: 14px; border-radius: 10px; padding: 10px; text-align: center; transition: all 0.3s ease-in-out; position: relative; z-index: 1; overflow: hidden; border: 1px solid transparent; border-bottom: 1px solid #000; box-sizing: border-box; display: flex; align-items: center; justify-content: center; gap: 10px; }
7 .area_visual .inr ul li:last-child{margin-left:300px; background: #2881ee; right:0; color: #fff; border: 1px solid transparent; border-bottom: 1px solid #000; }
8 .area_visual .inr ul li a{box-sizing:border-box; display:block; width:100%; height:100%; padding:30px; color: #fff; font-size: 14px; border-radius: 10px; position: relative; z-index: 1; transition: all 0.3s ease-in-out; }
9 .area_visual .inr ul li a i{position: absolute; right:30px; bottom:30px; font-size: 14px; color: #fff; border-radius: 50%; width: 20px; height: 20px; display: flex; align-items: center; justify-content: center; gap: 5px; background: #000; }
10 .area_visual .inr ul li a i img{height:33px; }
11 .area_visual .inr ul li a h3{margin-bottom:10px; font-size:22px; }
12 .area_visual .inr ul li a p{font-weight:300; word-break:keep-all; }
13 .area_visual .inr .tit {position: absolute; top:26%; left:0; color: #fff; font-size: 14px; border-radius: 10px; padding: 10px; text-align: center; transition: all 0.3s ease-in-out; position: relative; z-index: 1; }
14 .area_visual .inr .tit small {font-size:24px; padding-bottom: 30px; display: block; font-weight: 300; }
15 .area_visual .inr .tit h2 {font-size: 75px; line-height: 1.1; font-weight: 300; padding-bottom: 75px; text-align: left; }
16 .area_visual .inr .tit h2 span {color: #000; font-weight: bold; }
17 .area_visual .inr .tit p span {display: block; font-weight: 300; font-size: 25px; }
18 .area_visual .inr .tit p strong {font-weight: 400; color: #ddd; font-size: 18px; }
19
20 /*company*/
21 .company {width:100%; max-width:1200px; margin:0 auto; }
22 .company_con {width:100%; overflow:hidden; padding: 80px 20px; box-sizing: border-box; }
23 .text_box {float: left; width:40%; font-size: 18px; font-weight: 300; line-height: 1.5; color: #848484; padding-top: 25px; }
```





Sass

The logo features a large, stylized word "Sass" in a pink, flowing script font. Above the letter "S", there is a graphic element consisting of two interlocking shapes: a red, faceted sphere-like shape and a larger, smooth, light-pink swirl shape that loops around it.





Sass

```
$main-font: "Baskerville"

= title($font)
  font-size: 30px
  font-family: $font

#header
  +title($main-font)
```

Sass

```
$main-font: "Baskerville"  
  
=title($font)  
  font-size: 30px  
  font-family: $font  
  
.header  
  +title($main-font)
```

SCSS

```
$main-font: "Baskerville";  
  
@mixin title($font) {  
  font-size: 30px;  
  font-family: $font;  
}  
  
.header {  
  @include title($main-font);  
}
```



SCSS

```
1 section {  
2   height: 100px;  
3   width: 100px;  
4  
5   .class-one {  
6     height: 50px;  
7     width: 50px;  
8  
9     .button {  
10       color: #074e68;  
11     }  
12   }  
13 }
```

CSS

```
1 section {  
2   height: 100px;  
3   width: 100px;  
4 }  
5  
6 section .class-one {  
7   height: 50px;  
8   width: 50px;  
9 }  
10  
11 section .class-one .button {  
12   color: #074e68;  
13 }
```

SCSS 사용 방법 1



- Node.js 설치 하기
- <https://nodejs.org/ko/>
- 설치 되어 있는 분들은 잠시만 기다려 주세요~!

SCSS 사용 방법 1



- Node.js 가 잘 설치 되었는지 확인

```
lhs@DESKTOP-86MUCGC MINGW64 /d/git/movie_main (main)
$ node -v
v16.14.2
```

- SCSS 파일을 CSS 파일로 변환하여 사용 합니다!

SCSS 사용 방법 1



- Test 폴더 생성
- Test.scss 파일 생성

```
body {  
    background-color: orange;  
  
    ul,  
    ol {  
        list-style: none;  
    }  
  
    a {  
        text-decoration: none;  
    }  
}
```

SCSS 사용 방법 1

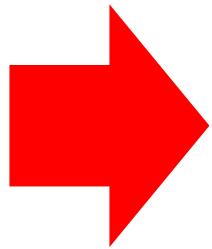


- npx sass --watch **SCSS파일.scss** **CSS파일.css**
- Npx 님! Sass 를 사용해서 scss 파일을 css 파일로 변경해 주세요!
- + 그리고 scss 파일이 보고 있다가(watch) 파일이 변경되면 CSS에 바로 적용해 주세요~!

```
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.
```



```
body {  
    background-color: orange;  
  
    ul,  
    ol {  
        list-style: none;  
    }  
  
    a {  
        text-decoration: none;  
    }  
}
```



```
body {  
    background-color: orange;  
}  
body ul,  
body ol {  
    list-style: none;  
}  
body a {  
    text-decoration: none;  
}  
  
/*# sourceMappingURL=test.css.map */
```

SCSS 사용 방법 1



- Scss 파일을 css 로 변경 & 파일에 변화가 있으면 알아서 scss 변경
- 해당 정보는 ~~.map 파일에 보관이 되며 해당 파일을 기준으로 변경
이 일어나므로 지우시면 안됩니다! 😊

```
{  
  "version": 3,  
  "sourceRoot": "",  
  "sources": [  
    "test.scss"  
  ],  
  "names": [],  
  "mappings": "AAAA;EACE;EACA;;AAEA;AAAA;EAEE;;AAGF;EACE",  
  "file": "test.css"  
}
```

SCSS 사용 방법 2



- VS Code 확장 프로그램 설치하기!

The screenshot shows the VS Code Marketplace search results for 'sass'. The top result is 'Sass' by Syler, which provides Indented Sass syntax highlighting. Below it is 'Live Sass Compiler' by Ritwick Dey, which compiles Sass or Scss to CSS at realtime. Other results include 'Beautify css/less' and 'Sass/Less/Stylus'.

Live Sass Compiler v6.0.2

Glenn Marks | 745,412 | ★★★★★(48)

Compile Sass or Scss to CSS at realtime.

사용 안 함 | 제거 | ⚙️

이 확장은 전역적으로 사용하도록 설정되었습니다.

세부 정보 기능 기여도 변경 로그 런타임 상태

Live Sass Compiler

VSCode Marketplace v6.0.2 downloads 1.5M rating 4.9/5 (48) license MIT

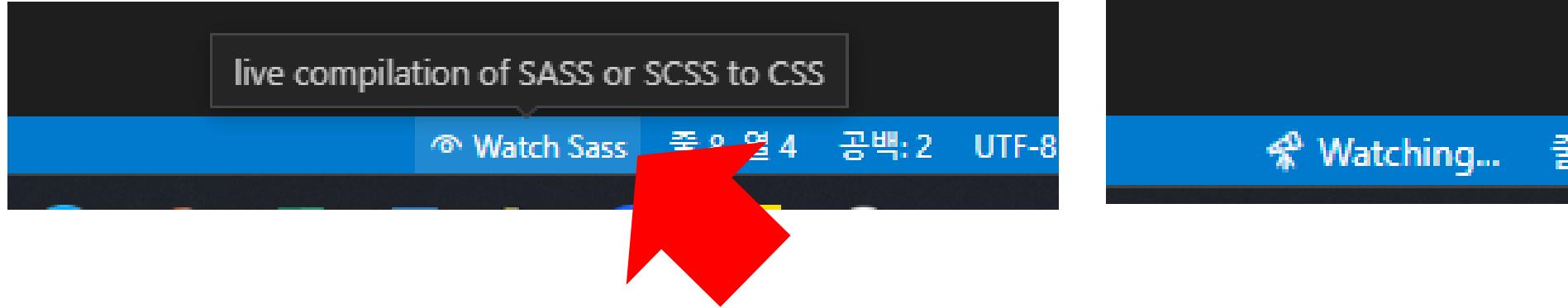
Like it? Please leave a review. Found something wrong? Report an issue.

SCSS 사용 방법 2



- Test2 폴더 생성 → test2.scss 파일 생성
- 코드는 기존 것을 카피!

SCSS 사용 방법 2



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

A screenshot of a file explorer showing a folder named "test2". Inside the folder are three files: "test2.css" (status "U"), "test2.css.map" (status "U"), and "test2.scss" (status "U"). There is also a folder named "video".



npm

Node Package Manager

VS

npx

Node Package eXecute



Syntactically
Awesome
Style Sheets



Import

다른 scss 파일 가져오기, import



- `_reset.scss` 파일 생성
- `_`는 부분 파일이라는 의미

```
//reset
* {
  margin: 0px;
  padding: 0px;
  box-sizing: border-box;
}
ul,
ol {
  list-style: none;
}
a {
  text-decoration: none;
}
```

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";
```



일단 일반

CSS로!

Emmet 으로 뼈대 갖추기!



- 태그명 → 그냥 쓴다
- 클래스명 → . 찍고 쓴다
- 자식 요소 → >
- 형제 → +
- 상위 요소 → ^
- 컨텐츠 → {}
- 속성 → []
- Lorem → lorem숫자(숫자는 단어 수, 단 컨텐츠로 쓰면 안되고 자식으로 써야함)

Emmet 으로 뼈대 갖추기!



```
section.wrap>article.item*4>h2{SCSS 연습}+p>lorem15^a[href="#" ]{more}
```

```
<section class="wrap">
  <article class="item">
    <h2>SCSS 연습</h2>
    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Esse soluta necessitatibus, odio aliquam
       deleniti accusamus.</p>
    <a href="#">more</a>
  </article>
  <article class="item">
    <h2>SCSS 연습</h2>
    <p>Culpa similique aperiam adipisci maxime perspiciatis, provident nemo ipsum totam hic praesentium quos
       sapiente exercitationem?</p>
    <a href="#">more</a>
  </article>
  <article class="item">
    <h2>SCSS 연습</h2>
    <p>Quos rerum delectus neque odio. Eaque id voluptatibus a consequatur dolor laboriosam autem minus dolorem?
       </p>
    <a href="#">more</a>
  </article>
  <article class="item">
    <h2>SCSS 연습</h2>
    <p>Excepturi aut cumque quod veniam ea voluptatibus laborum veritatis! Nisi vel officiis deserunt facere
       veritatis!</p>
    <a href="#">more</a>
  </article>
</section>
```



SCSS 연습

Lorem ipsum dolor sit, amet consectetur adipisicing elit. Veniam, ipsam! Est quas laudantium excepturi debitis.

[more](#)

SCSS 연습

Molestiae eveniet nihil inventore! Sint distinctio atque laborum qui? Nemo, similique nesciunt. Aperiam, laudantium aspernatur.

[more](#)

SCSS 연습

Deserunt maxime, maiores iste magni consequatur saepe earum illo eaque nihil beatae. Reprehenderit, amet tempore.

[more](#)

SCSS 연습

Itaque quaerat ex vel sapiente, rerum, neque eveniet earum at sequi qui vero quidem doloribus.

[more](#)

원래의 방법으로 CSS 만들어 보기



- 하나하나 해봅시다
- body
- Wrap
- Item
- Item h2
- Item p
- Item a



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

ul,
ol {
  list-style: none;
}

a {
  text-decoration: none;
}

body {
  background-color: #eee;
}

.wrap {
  width: 100%;
  border: 2px solid #ddd;
  padding: 100px;
  display: flex;
  justify-content: center;
}
```

```
.wrap .item {
  width: 300px;
  padding: 30px;
  background-color: #fff;
  margin: 50px;
  border-radius: 10px;
  box-shadow: 5px 5px 20px rgba(0, 0, 0, 0.1);
}

.wrap .item h2 {
  font: bold 30px/1 "arial";
  margin-bottom: 20px;
  color: #333;
}

.wrap .item p {
  font: normal 16px/1.5 "arial";
  color: #777;
}

.wrap .item a {
  display: block;
  width: 100px;
  height: 30px;
  border-radius: 15px;
  background-color: #555;
  color: white;
  text-align: center;
  font: bold 16px/30px "arial";
  margin-top: 20px;
}
```



SCSS 연습

Lorem ipsum dolor sit, amet
consectetur adipisicing elit.
Veniam, ipsam! Est quas
laudantium excepturi debitis.

more

SCSS 연습

Molestiae eveniet nihil inventore!
Sint distinctio atque laborum qui?
Nemo, similique nesciunt.
Aperiam, laudantium aspernatur.

more

SCSS 연습

Deserunt maxime, maiores iste
magni consequatur saepe earum
illo eaque nihil beatae.
Reprehenderit, amet tempore.

more

SCSS 연습

Itaque quaerat ex vel sapiente,
rerum, neque eveniet earum at
sequi qui vero quidem doloribus.

more



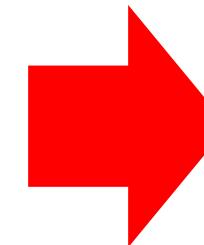
Nesting

Nesting 을 사용해 봅시다!



- {} 를 사용해서 요소간의 상하 관계를 편하게 작성해 봅시다!
- 각각의 자식요소를 {} 안에 넣어 주면 됩니다!
- 백문불여일견!

```
.wrap .item {  
    width: 300px;  
    padding: 30px;  
    background-color: #fff;  
    margin: 50px;  
    border-radius: 10px;  
    box-shadow: 5px 5px 20px rgba(0, 0, 0, 0.1);  
}  
  
.wrap .item h2 {  
    font: bold 30px/1 "arial";  
    margin-bottom: 20px;  
    color: #333;  
}  
  
.wrap .item p {  
    font: normal 16px/1.5 "arial";  
    color: #777;  
}  
  
.wrap .item a {  
    display: block;  
    width: 100px;  
    height: 30px;  
    border-radius: 15px;  
    background-color: #555;  
    color: white;  
    text-align: center;  
    font: bold 16px/30px "arial";  
    margin-top: 20px;  
}
```



```
@import "reset";  
  
.wrap {  
    width: 100%;  
    display: flex;  
    justify-content: center;  
    border: 2px solid #ddd;  
    padding: 100px;  
}  
  
.item {  
    width: 300px;  
    padding: 30px;  
    background-color: #fff;  
    margin: 50px;  
    border-radius: 50px;  
    box-shadow: 5px 5px 20px rgba(0, 0, 0, 0.1);  
}  
  
h2 {  
    font: bold 30px/1 "arial";  
    margin-bottom: 20px; color: #333;  
}  
p {  
    font: normal 16px/1.5 "arial"; color: #777;  
}  
a {  
    display: block;  
    width: 100px;  
    height: 30px;  
    border-radius: 15px;  
    background-color: #555;  
    color: white;  
    text-align: center;  
    font: bold 16px/30px "arial";  
    margin-top: 20px;    }}}}
```

어 그런데 Hover는 어떻게 걸죠?



- 자기 자신 선택자인 & 를 사용하면 됩니다!
- {} 내부에 &를 쓰고 &에 hover 를 걸어주면 됩니다!

```
&:hover {  
    background-color: orange;  
}
```

```
&::after {  
    content: "!!";  
}
```

& 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!(\$)



아..! 끝나니까
세상이 너무 아름답다..!!!

금방까지만 해도
미칠 것 같았는데..!!!







느낌적인 느낌으로 은 무슨 뜻인가요?

HiNative

```
body {
    background-color: #eee;
}

.wrap {
    width: 100%;
    border: 2px solid #ddd;
    padding: 100px;
    display: flex;
    justify-content: center;

    .item {
        width: 300px;
        padding: 30px;
        background-color: #fff;
        margin: 50px;
        border-radius: 10px;
        box-shadow: 5px 5px 20px rgba(0, 0, 0, 0.1);

        h2 {
            font: bold 30px/1 "arial";
            margin-bottom: 20px;
            color: #333;
        }

        p {
            font: normal 16px/1.5 "arial";
            color: #777;
        }

        a {
            display: block;
            width: 100px;
            height: 30px;
            border-radius: 15px;
            background-color: #555;
            color: white;
            text-align: center;
            font: bold 16px/30px "arial";
            margin-top: 20px;
        }

        &:hover {
            background-color: orange;
        }
    }
}
```



색상이 9개!!



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;  
}
```

물론 모든 속성 값도 변수로 사용 가능!



- 중요 속성을 변수로 지정하여 사용 가능!

```
$hund: 100px;  
$btn-shadow: 5px 5px 20px rgba(0, 0, 0, 0.1);
```



```
.wrap {  
width: 100%;  
display: flex;  
justify-content: center;  
border: 2px solid $wrap-border;  
padding: 100px;  
  
.item {  
width: 300px;  
padding: 30px;  
background-color: $wrap_item;  
margin: 50px;  
border-radius: 50px;  
box-shadow: $btn-shadow;  
  
h2 {  
font: bold 30px/1 "arial";  
margin-bottom: 20px;  
color: $wrap_item_h2;  
}  
  
p {  
font: normal 16px/1.5 "arial";  
color: $wrap_item_p;  
}  
  
a {  
display: block;  
width: 100%;  
height: 30px;  
border-radius: 15px;  
background-color: $btn-bg;  
color: $btn-font;  
text-align: center;  
font: bold 16px/30px "arial";  
margin-top: 20px;  
}
```



@mixin



아..! 끝나니까
세상이 너무 아름답다..!!!

금방까지만 해도
미칠 것 같았는데..!!!











MIXIN



JS의 객체 처럼 속성 값들을 저장하여 사용



- 일단 작업한 버튼 스타일을 봅시다!

```
a {  
    display: block;  
    width: 100px;  
    height: 30px;  
    border-radius: 15px;  
    background-color: orange;  
    color: white;  
    text-align: center;  
    font: bold 16px/30px "arial";  
    margin-top: 20px;  
}
```

JS의 객체 처럼 속성 값들을 저장하여 사용



- `_mixin.scss` 파일 생성
- `@mixin` 이름 () {} 으로 선언하여 사용

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

JS의 객체 처럼 속성 값들을 저장하여 사용



- 본래 SCSS 파일에서 mixin을 import!
- 원하는 속성에가서 @include 이름(); 으로 사용

```
@import "mixin";  
  
a {  
    @include button();  
}
```

그런데 그냥 쓰는거나 차이가 없지 않나요?



- 그래서 mixin 은 JS의 함수처럼 **매개변수**를 제공!

```
@mixin button($width, $bg-color, $font-color) {  
    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;  
}
```

```
a {  
    @include button(100%, $btn-bg, $btn-font);  
}
```

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;  
}
```



추가 기능~!

- SCSS 는 친절해서 사실 구문이 틀리면 어디서 틀렸는지 알려줍니다!

```
Compilation Error
Error: No argument named $btn-font.
    |   > d:\git\kdt-5th\scss\scsss\style.scss
44 |     @include button($width: 100%, $bg-color: $btn-bg, $btn-font: $btn-font);
    |     ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ invocation
    |
    |   > d:\git\kdt-5th\scss\scsss\_mixin.scss
1  | @mixin button($width: 100%, $bg-color: grey, $font-color: white) {
    |   _____ declaration
    |
d:\git\kdt-5th\scss\scsss\style.scss 44:7  button()
d:\git\kdt-5th\scss\scsss\style.scss 44:7  root stylesheet
```

협업 시에는!?



- Mixin 요소가 호출 되었을 때 제 3자가 매개 변수의 의미를 찾아보기 어렵기 때문에 해당 매개변수 이름을 명시하여 알려 주기도 합니다.

```
a {  
    @include button($width: 100%, $bg-color: $btn-bg, $font-color: $btn-font);  
}
```

- 그런데 보통 자신도 헷갈리기 때문에 쓰는게 좋습니다!



@for

먼저 Item에 효과를 주겠습니다!



- Wrap에 hover하면 버튼이 회전하면서 올라오는 효과를 줄게요!
- Item
 - Transform: translateY(100px) rotateY(180deg);
 - Opacity: 0;
 - Transition: 0.6s;
- Wrap
 - Perspective: 600px;

먼저 Item에 효과를 주겠습니다!



- &:hover
- Item
 - Transform: translateY(0px) rotateY(0deg);
 - Opacity: 1;



```
.wrap {  
  width: 100%;  
  display: flex;  
  justify-content: center;  
  border: 2px solid $wrap-border;  
  padding: 100px;  
  perspective: 600px;  
  
&:hover {  
  .item {  
    transform: translateY(0px) rotateY(0deg);  
    opacity: 1;  
  }  
}  
  
.item {  
  width: 300px;  
  padding: 30px;  
  background-color: $wrap_item;  
  margin: 50px;  
  border-radius: 50px;  
  box-shadow: $btn-shadow;  
  transform: translateY(100px) rotateY(180deg);  
  transition: 0.6s;  
  opacity: 0;
```

순차적으로 올라오도록 하고 싶어요!



- 실질적으로 돌아가는 Item 요소에 transition-delay 를 줍시다!

```
.item {  
  
  &:nth-child(1) {  
    transition-delay: 0.1s;  
  }  
  &:nth-child(2) {  
    transition-delay: 0.2s;  
  }  
  &:nth-child(3) {  
    transition-delay: 0.3s;  
  }  
  &:nth-child(4) {  
    transition-delay: 0.4s;  
  }  
}
```



@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;  
  }  
}
```

@for 문 특징



- 숫자는 `#{}()` 안에 \$를 사용하는 변수를 써야 합니다!

```
&:nth-of-type(#{$i}) {}
```

- from 1 to 5
 - 1 ~ 4 까지 값이 들어 갑니다!
- from 1 through 5
 - 1 ~ 5 까지 값이 들어 갑니다!



@each

JS 배열의 `forEach` 와 유사합니다!



- 각각 버튼의 배경 색상을 변경해 봅시다!
- 버튼 배경으로 사용할 배열 선언하기!

```
$bgColors: (orange, hotpink, aqua, lightgreen);
```

- `Index()` 메소드로 index 값을 구할 수 있습니다!

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

하는 김에 아까운의 @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);  
    }  
  }  
}
```

실습, 글자 색 변경



- 아이템 내부의 h2, p 태그의 글자 색상도 `@each` 를 사용하여 버튼 색과 동일하게 맞춰 봅시다!

SCSS 연습

Lorem ipsum dolor sit, amet consectetur adipisicing elit. Veniam, ipsam! Est quas laudantium excepturi debitis.

[more](#)

SCSS 연습

Molestiae eveniet nihil inventore! Sint distinctio atque laborum qui? Nemo, similique nesciunt. Aperiam, laudantium aspernatur.

[more](#)

SCSS 연습

Deserunt maxime, maiores iste magni consequatur saepe earum illo eaque nihil beatae. Reprehenderit, amet tempore.

[more](#)

SCSS 연습

Itaque quaerat ex vel sapiente, rerum, neque eveniet earum at sequi qui vero quidem doloribus.

[more](#)



@if

@if 문



- If 문을 사용해서 간단하게 다크 모드를 만들어 봅시다!
- 특정 조건에 따라서 속성을 변경하고 싶을 때 사용!

```
@if $wrap_item == #fff {  
    color: black;  
} @else {  
    color: white;  
}
```

@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;  
    }  
}
```

@if 문



- Item 의 백그라운드 색상을 흰색이 아닌 어두운 계열로 변경해 봅시다!
- \$body-bg 를 #111 로 변경
- \$wrap_item 를 #222 로 변경!



SCSS 연습

Lorem ipsum dolor sit, amet
consectetur adipisicing elit.
Veniam, ipsam! Est quas
laudantium excepturi debitis.

more

SCSS 연습

Molestiae eveniet nihil inventore!
Sint distinctio atque laborum qui?
Nemo, similique nesciunt.
Aperiam, laudantium aspernatur.

more

SCSS 연습

Deserunt maxime, maiores iste
magni consequatur saepe earum
illo eaque nihil beatae.
Reprehenderit, amet tempore.

more

SCSS 연습

Itaque quaerat ex vel sapiente,
rerum, neque eveniet earum at
sequi qui vero quidem doloribus.

more



영상 기반 랜딩 페이지

KDT Team Tetz 5th

☺ Coding 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/>



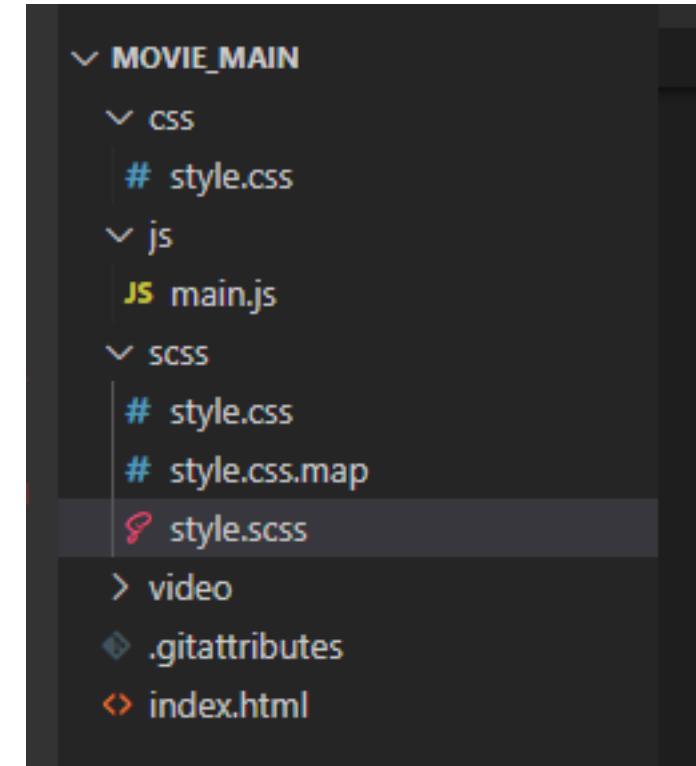
뼈대를 세우자

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 아이템을 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 해야겠죠? → `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 컨텐츠의 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");
  }
});
```

