

실습 1. 카페 음료 문서 작성

- 카페 이름은 <h1>, 메뉴 이름은 <h2>
- 카페 전체 음료 리스트는 태그, 음료 최소 3개
 - 의 자식 요소 요소를 넣고 가격, 사이즈를 표시
- 태그 아래에 수평 줄을 추가하고 그 아래에 각각 음료에 대한 설명을 <p>태그에 작성하되, 설명에서 메뉴 이름은 태그로 감싸기
- 카페 이름은 파란색 으로 표시 (style="color:blue;" 속성 추가)
- <p> 태그 내의 음료 설명 중 메뉴 이름은 초록색 으로 표시 (style="color: green;")

실습 2. img, a 태그 실습

- 앞에서 진행한 실습1에 이미지를 추가
- 이미지에 마우스를 올렸을 때 카페 이름이 나옴
- 이미지를 클릭했을 때 <https://google.com> 이 새 창으로 열림

실습 3. Table 달력 실습

2022년 9월

일	월	화	수	목	금	토
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

style="color: 컬러이름;"

속성 사용

실습 4. Table 시간표 실습

시 간 표

	월	화	수	목	금
1교시					
2교시					
3교시					
4교시					
MARVEL					
5교시					
6교시					





MARVEL 로고를 눌렀을 때, 새 탭이 아닌 현재 탭에서

유튜브 마블 영상 링크로 이동

실습 5. 카페 음료 주문 테이블 작성

- 실습 1에서 작성한 카페 메뉴 리스트를 Table 로 만들기
- 음료 이름을 누르면 해당 음료 저버가 있는 페이지로 이동!

스타벅스

구분	음료	가격	이미지
에스프레소	커피	2500	
	아이스 화이트 초콜릿 모카	2500	
	카페 모카	2500	
블렌디드	망고 패션 푸르트 블렌디드	3000	
	딸기 딜라이트 요거트 블렌디드	4000	
	제주 그린 한라봉 모히또 블렌디드	4000	

실습6. CSS 직접 사용해보기

```
<html>
  <head>
    <meta charset="utf-8">
    <style>
      div {

      }
    </style>
  </head>
  <body>
    <div> 안녕하세요. 아무말.... </div>
  </body>
</html>
```

- color, background-color
- font-size, font-weight, font-family, font-style
- width, height
- border
- margin, padding

여러 CSS 속성들을 직접 사용해보자!!

실습7. CSS 선택자 활용하기

2022년 9월

일	월	화	수	목	금	토
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

이 실습에서 비효율적으로 적용시켰던 style을
오늘 배운 개념을 활용하여 바꿔보기.

+ 코드 일부를 함께 캡처해 올려주세요!

실습 8. CSS 실습 (1)

- 다양한 position 속성을 이용해 애벌레 만들기

1

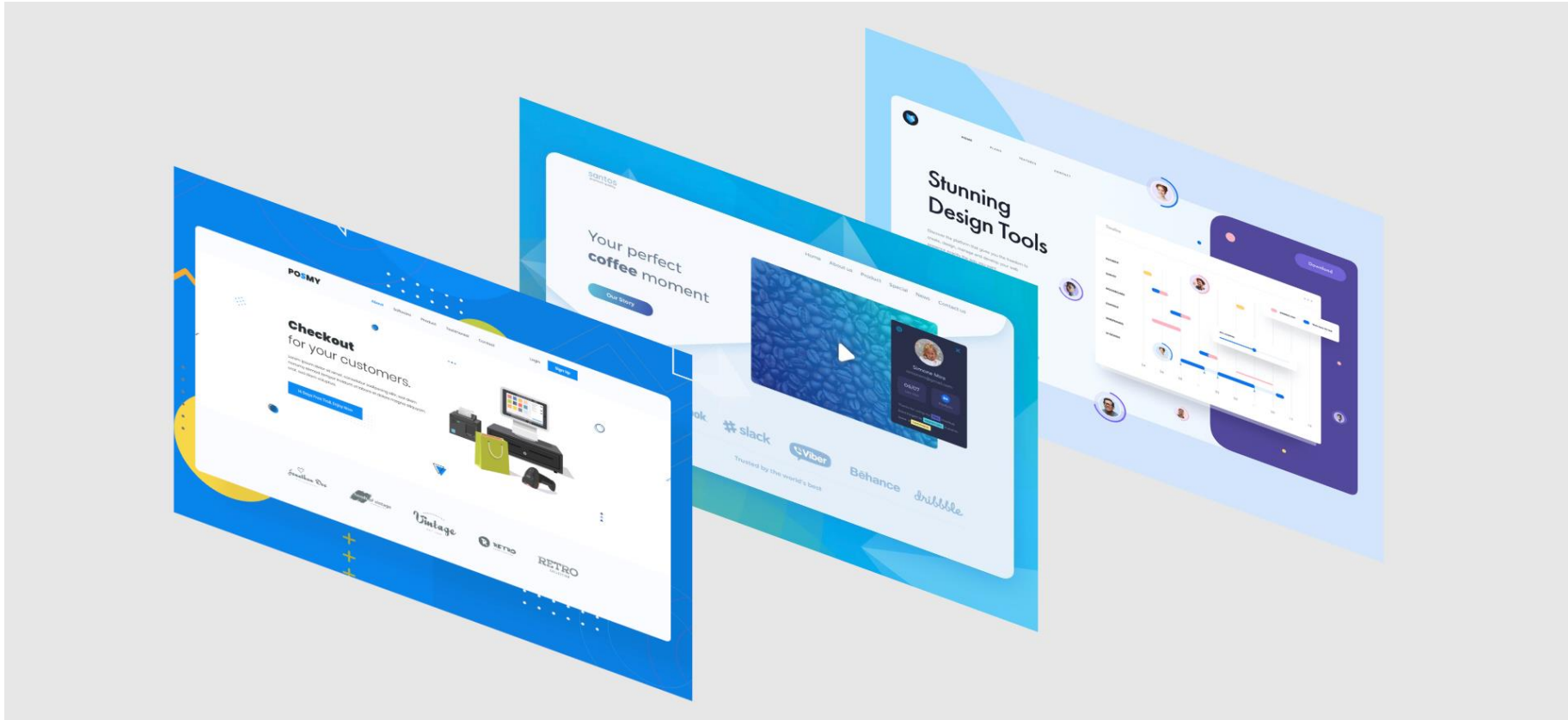


2



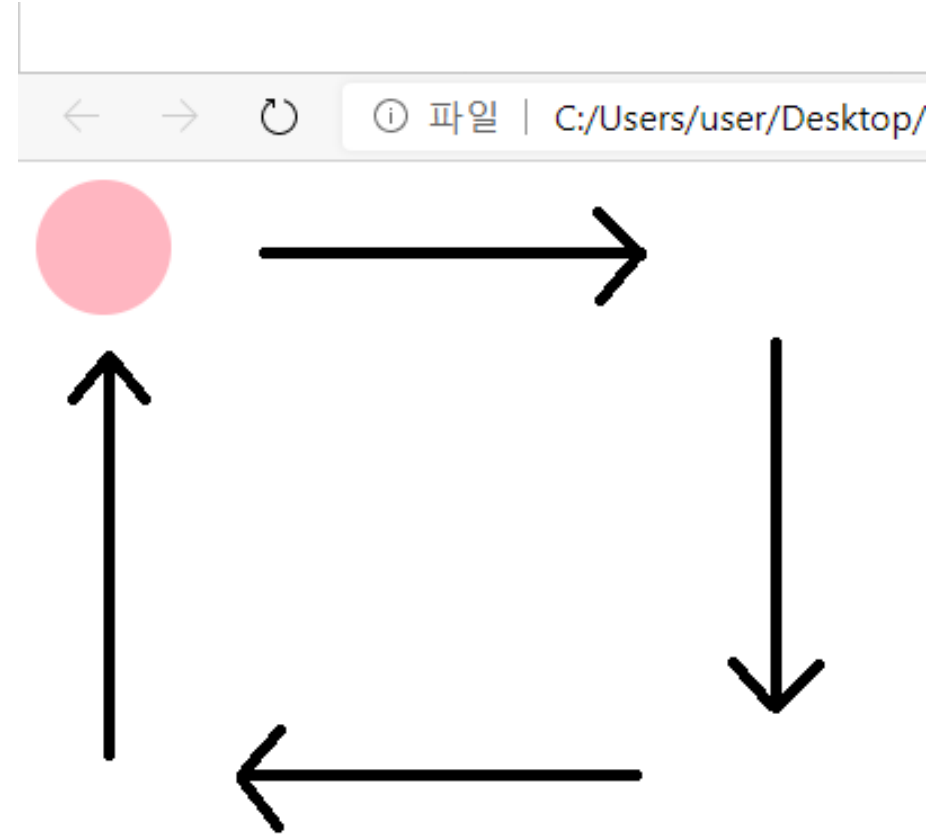
실습 9. CSS 실습 (2)

- Transform 과 z-index 를 이용해 이미지 3개를 연달아 나열하기



실습 10. CSS 실습 (3)

1. 키프레임을 사용하세요.
2. 왼쪽 상단에서 시작하여 시계 방향으로 사각형을 그리는 애니메이션을 완성하세요.
3. 애니메이션 대상은 원입니다.
4. 애니메이션은 3초 동안 지속됩니다.
5. 애니메이션은 6번 반복됩니다.

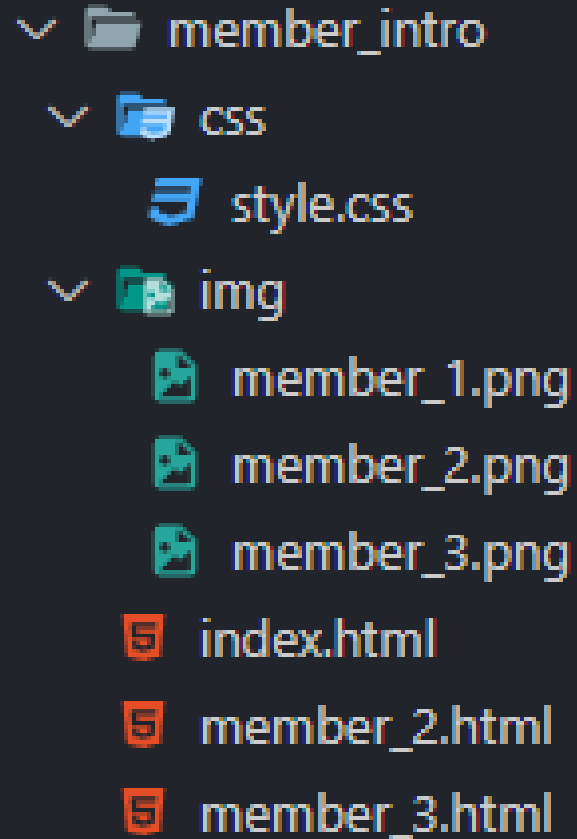


실습 11. CSS 실습 (4)

<https://d2iwdqgro8i2ew.cloudfront.net/lesson/script/53006-1.html>

를 눌러 실행되는 파일처럼 실행되게 만들어주세요.

추가 실습1. 미니 프로젝트



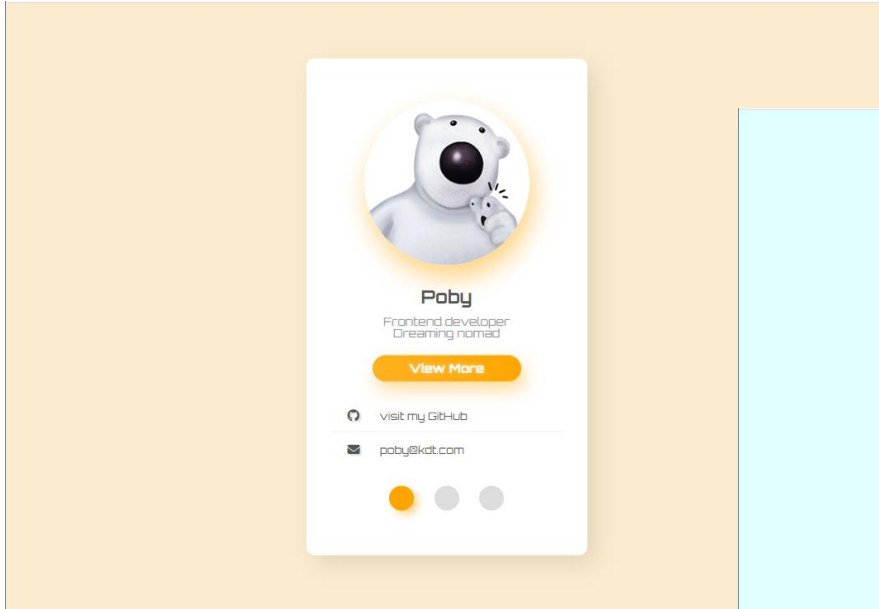
A file explorer window showing the project structure. The root folder is 'member_intro'. Inside it, there are three subfolders: 'css', 'img', and 'index.html'. The 'css' folder contains a file 'style.css'. The 'img' folder contains three files: 'member_1.png', 'member_2.png', and 'member_3.png'. The 'index.html' folder contains three files: 'index.html', 'member_2.html', and 'member_3.html'.

- member_intro
 - css
 - style.css
 - img
 - member_1.png
 - member_2.png
 - member_3.png
 - index.html
 - index.html
 - member_2.html
 - member_3.html

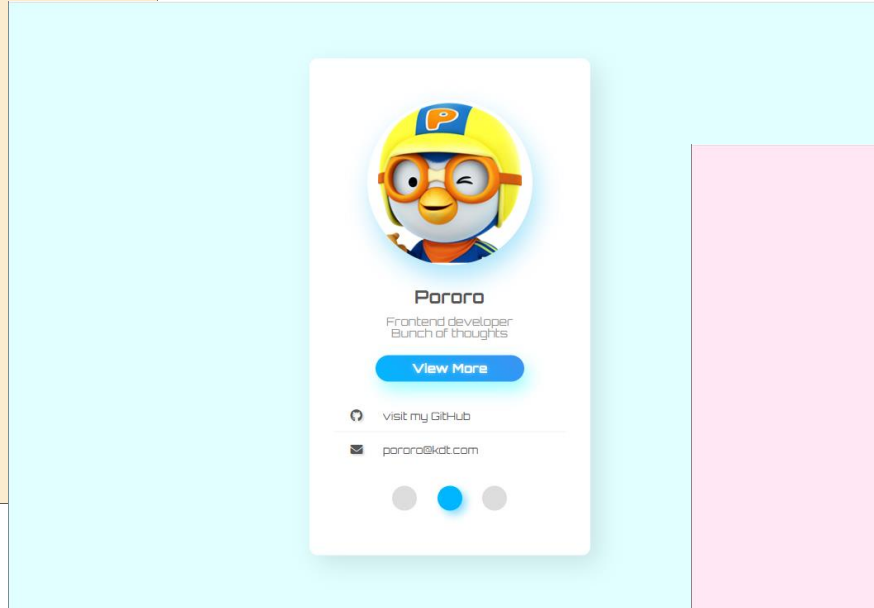
- 프로젝트 구조
 - 필요한 파일을 모두 생성함
 - 하나의 css 파일로 3개의 html 파일을 디자인하기

추가 실습1. 미니 프로젝트

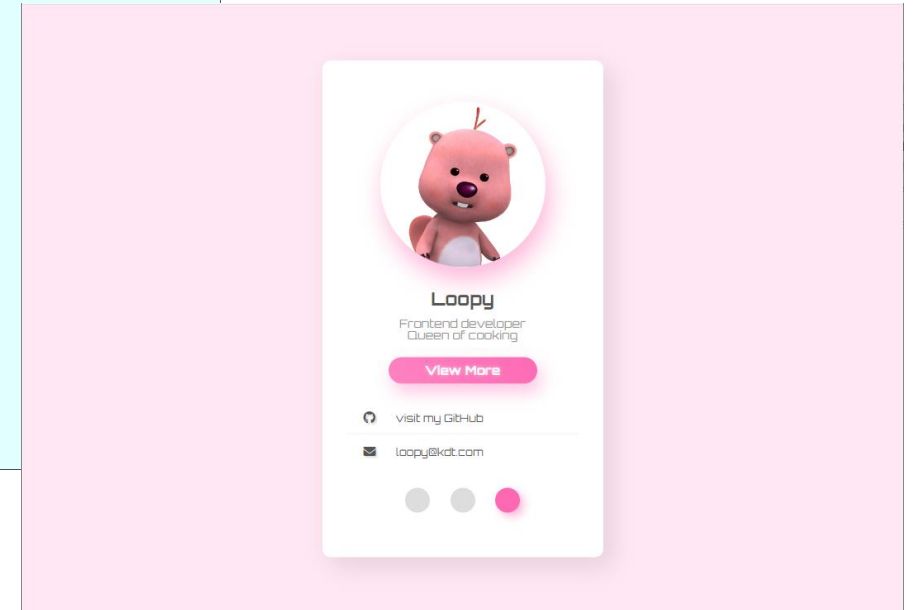
- 목표: 다음 결과물이 나오도록 **html**과 **css** 파일을 수정해주세요!



index.html



member_2.html



member_3.html

*프로젝트 실행 영상 Slack 참고

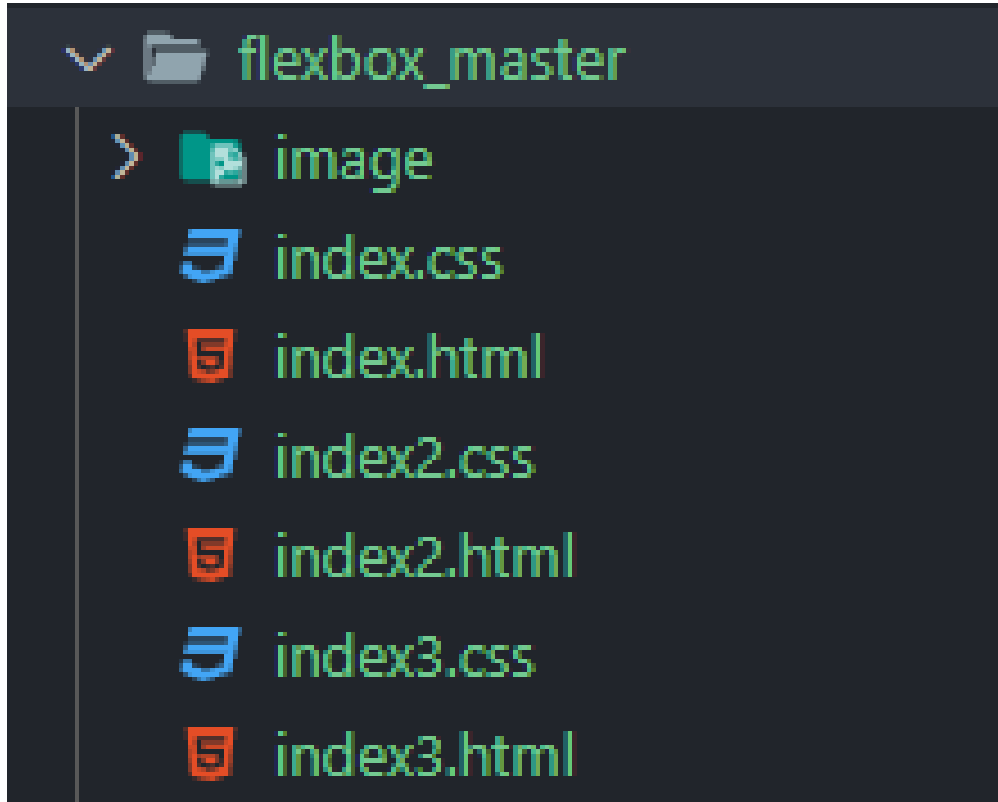
추가 실습1. 미니 프로젝트

- index.html -> member_2.html -> member_3.html 순으로 작업하기
- html 파일 구조는 수정하실 필요는 없어요. html 파일에 적합한 TODO 조건을 충족해주세요.
- style.css 파일에 오늘 배운 css 속성들을 추가하여 프로젝트를 완성해주세요.
 - 오늘 배운 속성이 아닌 경우, 미리 css 파일에 설명과 함께 선언해 두었습니다.
 - 막히거나 궁금한 속성이 있다면 질문해주세요!

추가 실습1. 미니 프로젝트

- 오늘 배운 css 속성 정리
 - color, background-color
 - font-size, font-weight, font-family, font-style
 - width, height
 - border
 - margin, padding
 - box-shadow, text-shadow

추가 실습2. Flexbox 마스터하기



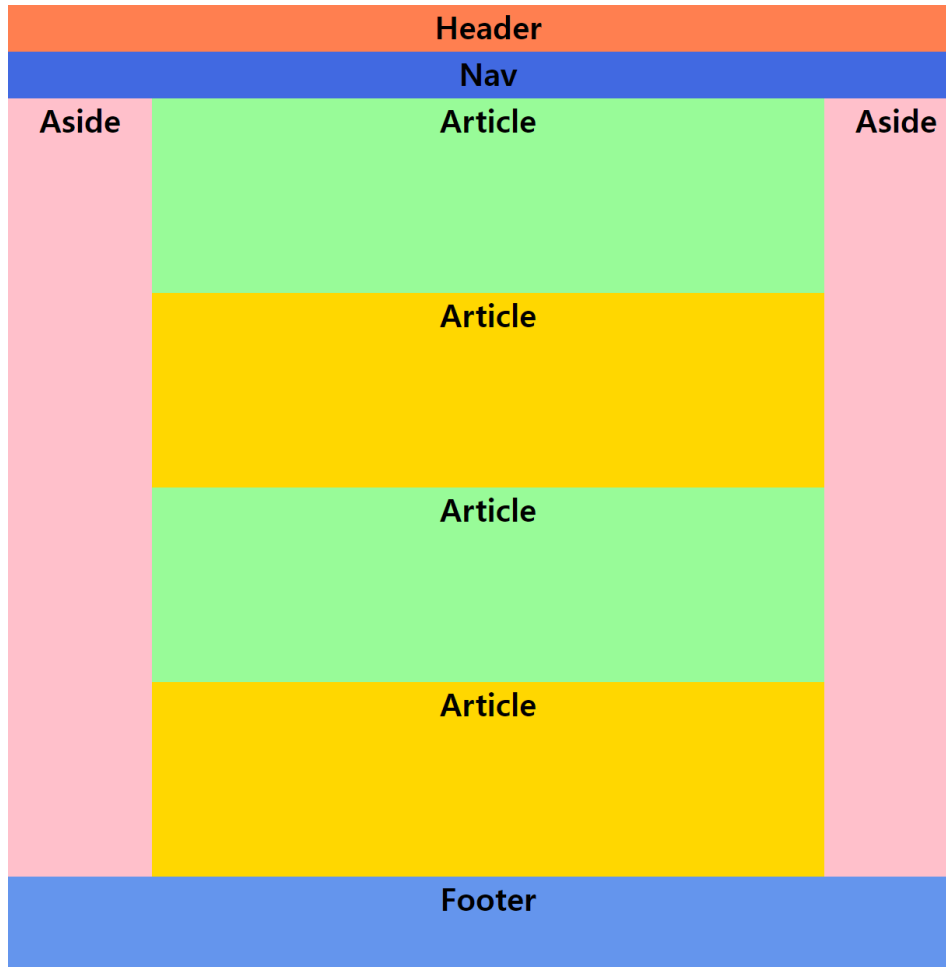
- 파일 구조

- html, css 파일을 생성함
- 필요한 이미지를 추가함

HTML 코드는 최대한 수정하지 않을 것!

CSS만 수정 해주세요~

추가 실습2. Flexbox 마스터 No.1



- **index1.html**
- **[복습] flex를 이용해 시멘틱 웹 구성하기**

추가 실습2. Flexbox 마스터 No.2

Flex를 이용해 여러 종류의 Header를 만들어보자!

LOGO	MENU1	MENU2	MENU3
MENU1	MENU2	MENU3	
LOGO	MENU1	MENU2	MENU3
			LOGIN SIGNUUP
LOGO	MENU1	MENU2	MENU3
			LOGIN

- index2.html

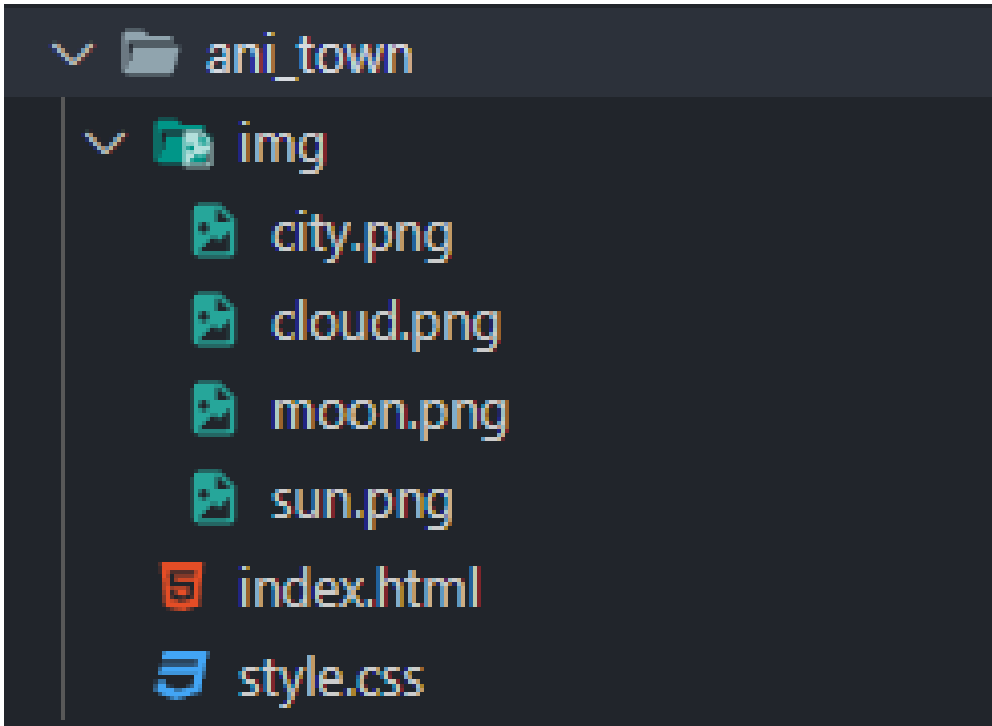
- flex 를 이용해 다양한 header를 만들어 주세요.

추가 실습2. Flexbox 마스터 No.3



- index3.html
- html 파일 구조 수정 금지!!
- flex를 이용해 다음 레이아웃을 구성해주세요.
- Slack에서 실행 영상을 꼭 확인해주세요.

추가 실습3. 미니 프로젝트



- 프로젝트 구조
 - html, css 파일을 생성함
 - 필요한 이미지를 추가함

추가 실습3. 미니 프로젝트



* 애니메이션이 적용되어 있어 프로젝트 실행 영상을 Slack에서 참고해주세요!

추가 실습3. 미니 프로젝트

- 실행영상을 참고(Slack #과제)하여 css 파일을 수정해주세요.
- html 파일 구조는 수정하실 필요는 없어요. html 파일에 적합한 TODO 조건을 충족해주세요.
- style.css 파일에 css 속성들을 추가하여 프로젝트를 완성해주세요.
 - 과제에 필요한 속성 설명과 값을 추가했으니 주석을 참고해주세요.
 - 막히거나 궁금한 속성이 있다면 질문해주세요!

실습 12. Javascript 실습

1. no_1, no_2, no_3 이라는 3개의 변수를 선언하고, 각각 숫자 할당하기
2. no_1, no_2, no_3의 합계를 alert로 출력하는 add 함수 만들기
3. no_1과 no_2의 차이를 alert로 출력하는 sub 함수 만들기
4. no_1에서 no_2를 나눈 값을 alert로 출력하는 divide 함수 만들기
5. no_1과 no_2의 곱을 alert로 출력하는 mul 함수 만들기

실습 13. 구구단 만들기

```

===== 2 단 =====
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
2 * 6 = 12
2 * 7 = 14
2 * 8 = 16
2 * 9 = 18
2 * 10 = 20
===== 3 단 =====
3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
3 * 4 = 12
3 * 5 = 15
3 * 6 = 18
3 * 7 = 21
3 * 8 = 24
3 * 9 = 27
3 * 10 = 30
===== 4 단 =====
4 * 1 = 4
4 * 2 = 8

```

콘솔 창에 구구단 2~9단 띄우기

힌트 1. <script> 에서 콘솔에 글 띄우는 방법

`console.log(띄울 내용);`

힌트 2. 변수와 글자를 연결하는 방법

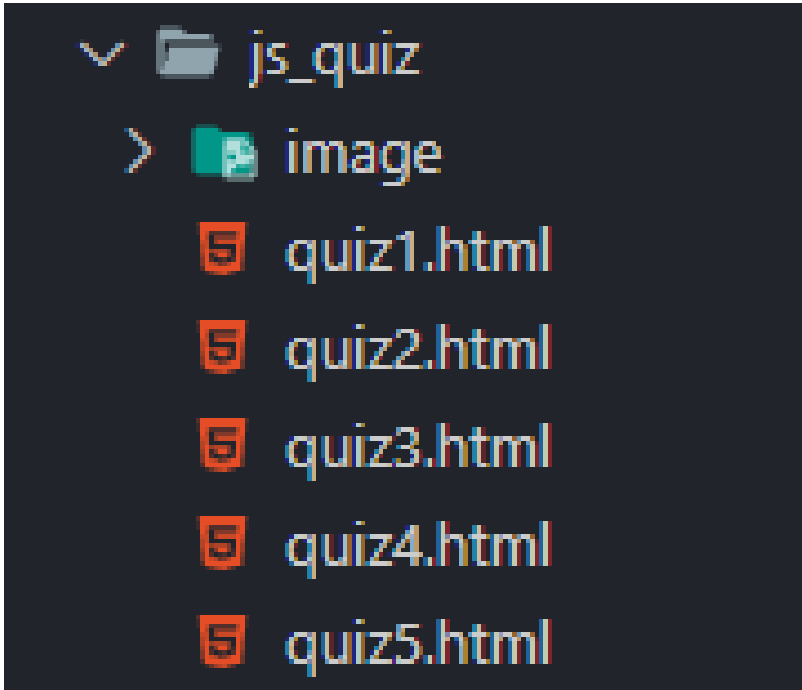
`var i = 0;`

`console.log(i + “입니다.”);`

실습 14. 합계 구하기

정수 n 이 있을 때 2또는 3의 배수의 총합을 구하시오.
(단, n 은 $0 < n < 100$)

추가 실습4. js_quiz 문제 풀기



- js_quiz 압축 파일을 해제해서 **quiz1.html** ~ **quiz5.html**까지 파일을 열어서 각각 풀어주세요.
- 각 html 파일의 **TODO**를 만족하도록 코드를 수정해주세요!

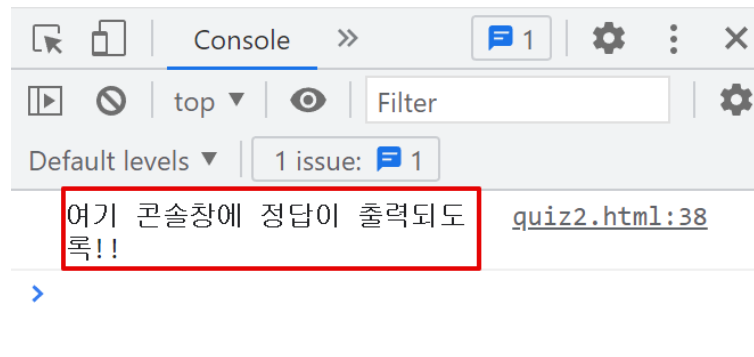
추가 실습4. js_quiz 문제 풀기

- quiz_1.html

- 문제를 다 푸시면 Slack DM 전송해주세요. 정답 파일 공유 드리겠습니다 :)

- quiz_2.html ~ quiz_5.html 캡처 화면 예시

- 콘솔 창에 정답이 함께 보이도록 캡처해주세요!

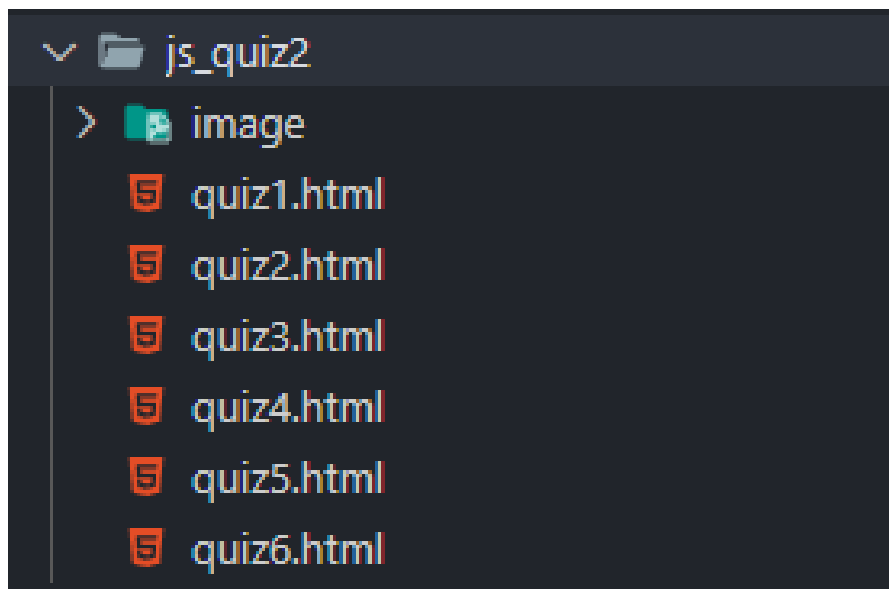


퀴즈2. 스마일 기호 프린트하기

:D 문자열을 콘솔창에 출력하는 printSmile() 함수를 정의하고, 함수 호출하기

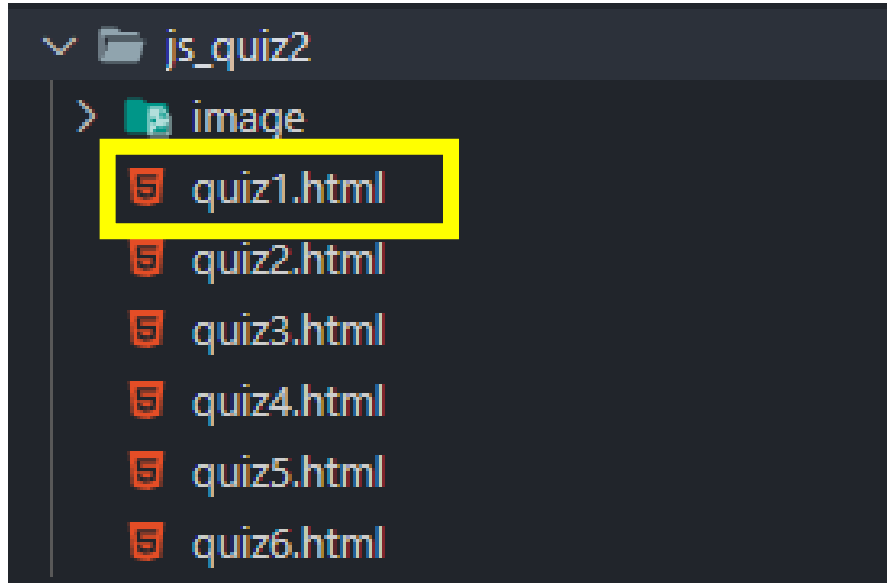
- 함수 호출 예시: `printSmile();`
- 출력 결과 예시: `:D` quiz2.html 콘솔창

추가 실습5. js_quiz2 문제 풀기



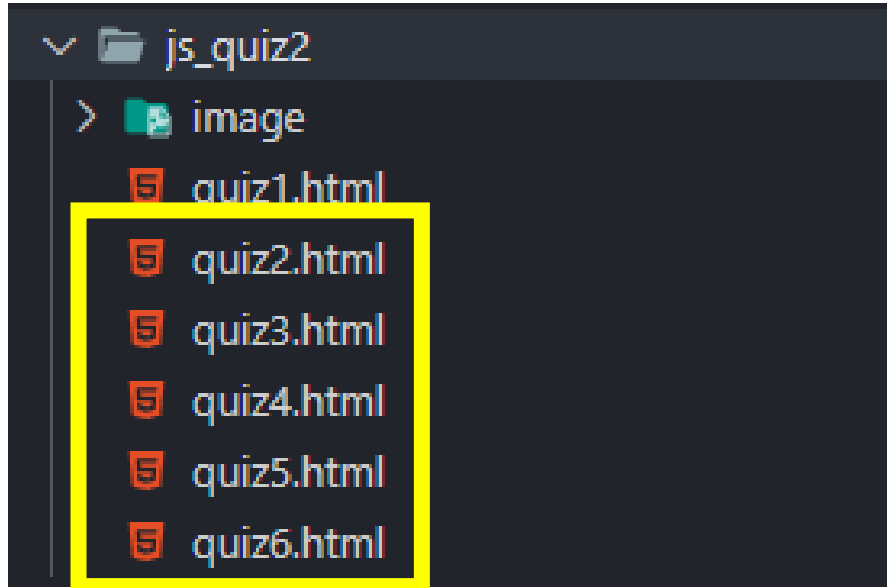
- js_quiz 압축 파일을 해제해서 **quiz1.html** ~ **quiz6.html**까지 파일을 열어서 각각 풀어주세요.
- 각 html 파일의 **TODO**를 만족하도록 코드를 수정해주세요!

추가 실습5. js_quiz2 문제 풀기



- quiz1.html 파일은 객관식 문제입니다.
 - **14:00 에 정답 파일 공개할 예정입니다.**
- 미리 풀어주세요~

추가 실습5. js_quiz2 문제 풀기



- quiz2~6.html 파일
: HTML의 script 태그안에 미리 구조가 정의되어 있는 함수를 조건에 맞게 수정해주세요!

실습 15. 웹페이지 제작 실습

조건 1. html 파일 최소 2개 만들기

조건 2. Bootstrap Component 또는 Utility 5개 이상 사용하기

조건 3. 모든 페이지는 반응형으로

html 파일 캡처해서 업로드 해주세요!

간단 실습 Time

Bootstrap 실습 코드에서
Bootstrap Components에 있는 기능 추가하여,
add -> commit -> push까지 진행해보기

실습 16. Merge(1)

```
<html>  
  <head>  
    <title>test</title>  
  </head>  
  <body>  
    <h1>반가워요</h1>  
    <h5>안녕하세요</h5>|  
  </body>  
</html>
```

1. main 브랜치에서 test.html 파일을 위와 같이 만든 후 push
2. “test” 라는 이름의 branch 만들기
3. main 브랜치에서 test.html의 h5 부분 수정하고 commit
4. test 브랜치로 옮겨가기
5. test 브랜치에서 test.html 내의 title과 h5 내용 수정 후 push
6. main 브랜치에 test 브랜치 merge 하기
7. 충돌 해결 후 push

실습 16. Merge(2)

1. main 브랜치에서 test.html 파일을 위와 같이 만든 후 push
2. “test” 라는 이름의 branch 만들기
3. main 브랜치에서 test.html의 h5 부분 수정하고 commit
4. test 브랜치로 옮겨가기
5. test 브랜치에서 test.html 내의 title과 h5 내용 수정 후 push
6. main 브랜치에 test 브랜치 merge 하기
7. 충돌 해결 후 push

slack 과제 제출 방식

B I

```
1.  
git add .  
git commit -m "study: Create test.html file"  
git push origin master  
2. git checkout -b test  
3.
```

+ @ Aa

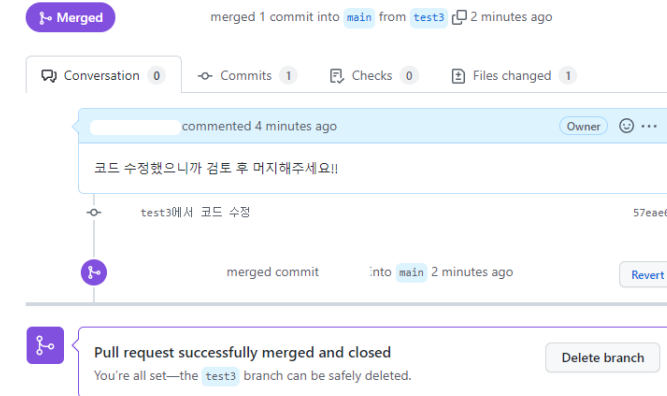
위와 같이 git 명령어와 7번까지 해결한 이후
github 홈페이지에서 test.html 내용 캡처해서 제출

실습 17. Pull Request

1. 앞선 폴더에서 계속해서 진행
2. “test2” 라는 이름의 branch 만들기
3. test2 브랜치로 옮겨가기
4. test2 브랜치에서 test.html 내용 수정 후 push
5. github에서 pull request 생성 후 merge (test2에서 main으로)

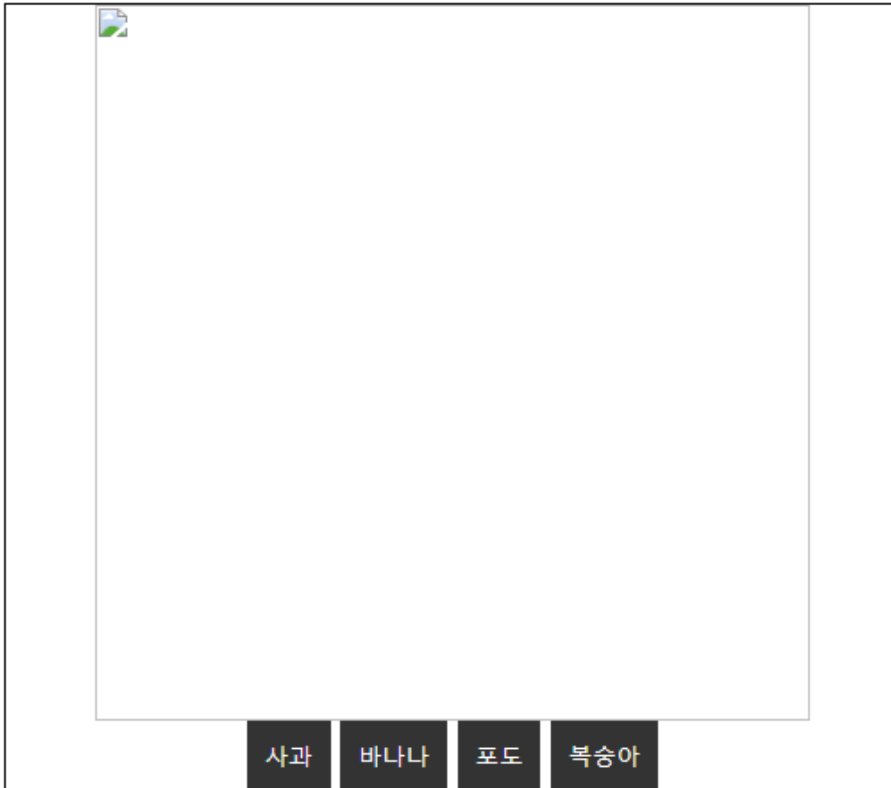
slack 과제 제출 방식

test3에서 코드 수정 #1



github repository > pull request > closed 탭
closed된 pull request 클릭 후 내역 캡처하여 제출

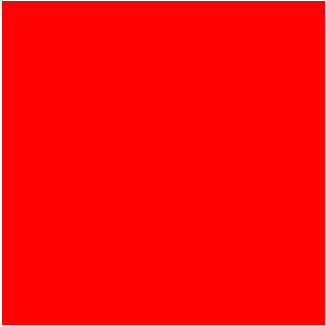
실습 18. 이미지 바꾸기



Jquery를 이용해 과일 버튼을 누르면 해당하는 과일 이미지 나오게 하기

실습 19. 클래스 바꾸기

실행영상



Slack 실행영상 참고

div 클릭할 때 마다 배경색 변경되도록!

실습20. ejs 사용하기

1. app.js에서 app.get() 3개 이상 만들기.
2. 1에서 만든 get 요청을 ejs 파일 <태그>로 이동할 수 있게 만들기.
3. ejs 문법, img 태그 직접 사용해보기.

실습21. 서버 업로드

1. ejs 사용한 프로젝트 파일 서버로 업로드 후 서버에서 실행시키기

힌트

- 1) 서버에서 npm install 을 통해 패키지 설치하기
- 2) 터미널 or Putty 에서 node 실행

팀프로젝트 과제

1. 팀 프로젝트로 사용할 node 프로젝트를 local에서 생성하기

1. app.js, static, views 기본구조 만들기
2. http://localhost:port_num 접속시 main 페이지 보이도록 하기

2. node 프로젝트 github에 올리기

1. github organization 생성 -> repository 생성
2. 작업내역 github에 push!

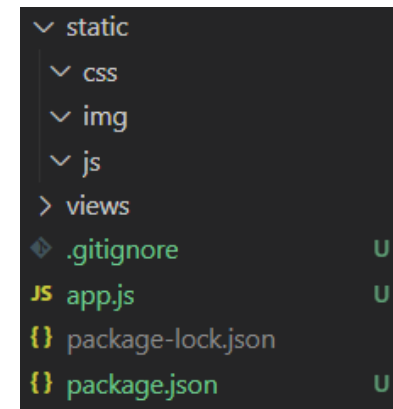
3. NCP 서버에서도 해당 프로젝트 실행되도록 하기

방법1. git clone 명령어 이용

방법2. FileZilla 이용해 프로젝트 폴더 업로드

필수. NCP ACG설정에서 port 번호 추가하기!

-> 서버IP주소:포트번호 접속시 메인 페이지가 보여야 합니다!



프로젝트 구조 예시

막히는 팀은 도움 요청하기!

완성한 팀은 확인 요청하기!

실습23. Callback -> Promise

콜백 함수로 이루어진 코드를 Promise로 변경하기

```
function call(name, cb) {  
  setTimeout(function () {  
    console.log(name);  
    cb(name);  
  }, 1000);  
}
```

```
function back(cb) {  
  setTimeout(function () {  
    console.log('back');  
    cb('back');  
  }, 1000);  
}
```

```
function hell(cb) {  
  setTimeout(function () {  
    cb('callback hell');  
  }, 1000);  
}
```

```
call('kim', function (name) {  
  console.log(name + '반가워');  
  back(function (txt) {  
    console.log(txt + '을 실행했구나');  
    hell(function (message) {  
      console.log('여기는 ' + message);  
    });  
  });  
});
```

🚀 실행 결과

kim

kim 반가워

back

back 을 실행했구나

여기는 callback hell

실습24. Promise - async/await

실습23 에서 promise로 바꾼 코드를 exec 함수를 만들어 실행하게 하기
(이때, exec는 async 함수이다.)

Hint!!

🔪 실행 결과

kim

kim 반가워

back

back 을 실행했구나

여기는 callback hell

```
▼ async function exec(){  
  let user = await login( "kim" );  
  console.log( user + "님 환영합니다.");  
  let videos = await getVideo(user);  
  console.log( videos );  
  let title = await getDetail(videos[0]);  
  console.log( title );  
}  
  
exec();|
```

실습25. Promise로 배경색 변경하기

오늘 수업에서 작성한 콜백지옥으로 작성한 배경색 변경 코드를 Promise를 이용해 변경하기

(10_asynchronous/index.js 참고)

```
setTimeout(function () {  
  document.body.style.backgroundColor = 'red';  
  setTimeout(function () {  
    document.body.style.backgroundColor = 'orange';  
    setTimeout(function () {  
      document.body.style.backgroundColor = 'yellow';  
      setTimeout(function () {  
        document.body.style.backgroundColor = 'green';  
        setTimeout(function () {  
          document.body.style.backgroundColor = 'blue';  
        }, 1000);  
      }, 1000);  
    }, 1000);  
  }, 1000);  
}, 1000);
```

실습 26. get으로 정보 받기

실습26. get으로 정보받기

이름

성별 ☐ 남자 ☐ 여자

생년월일
 년 월 일

관심사 ☐ 여행 ☐ 패션 ☐ 음식

1. 이름, 성별, 생년월일, 관심사 를 포함하여 4개 이상의 정보를 입력받습니다.
2. 이름의 input type은 text 입니다.
3. 성별의 input type은 radio button 입니다.
4. 생년월일의 input type은 select 입니다.
5. 관심사의 input type은 checkbox 입니다.

실습 27. post로 정보 받기

실습27. post로 정보받기

이름

성별 ☐ 남자 ☐ 여자

생년월일 년 월 일

관심사 ☐ 여행 ☐ 패션 ☐ 음식

선택

좋아하는 색상

좋아하는 숫자

1. form 정보는 url에 정보가 보이지 않는 post로 받기
2. 또 다른 input 태그 2개 추가하기 (이때, type은 text, checkbox, radio 를 제외한 다른 것 이용하기)

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

실습 28. post로 필수 정보 받기

실습27. post로 정보받기

이름

성별 ☐ 남자 ☐ 여자

생년월일

2010 년 1 월 1 일

관심사 ☐ 여행 ☐ 패션 ☐ 음식

선택

좋아하는 색상

좋아하는 숫자

회원가입

실습27파일에서 이름과 성별 선택은 필수로 선택하게 만들기
(선택하지 않으면 정보 전송 x)

실습29. axios get 회원가입

실습29. axios get 회원가입

이름

성별 ☒ 남자 ☐ 여자

생년월일 년 월 일

관심사 ☒ 여행 ☐ 패션 ☐ 음식

앞에서 진행한 실습 “회원가입” 을
axios의 **get 메소드**를 이용해 받게끔
작업하기

실습30. axios post 로그인

app.js에서 id, pw를 변수로 저장해두고, 로그인 할 수 있게 만들기

이때, 로그인은 **axios**의 **post**를 이용하기

Axios 의 결과를 받아와 “로그인“ 버튼 아래 메시지로 보여주기

- 실패 메시지는 빨간 글자
- 성공 메시지는 파란 글자
- Ex) 네이버 로그인 화면



The image shows the Naver login interface. At the top, the word "NAVER" is in green. Below it, there's a navigation bar with three tabs: "ID 로그인" (selected), "[1] 일회용 번호", and "QR코드". A small green banner at the top right says "PC방 등 공용PC라면 QR코드 로그인이 더 안전해요 X". The main form has two input fields: the first contains "spreatics" and the second is a password field with dots. Below the fields are two checkboxes: "로그인 상태 유지" (checked) and "IP보안" (unchecked). A red error message is displayed: "아이디(로그인 전용 아이디) 또는 비밀번호를 잘못 입력했습니다. 입력하신 내용을 다시 확인해주세요." Below this is a large green "로그인" button. At the bottom, there are links for "비밀번호 찾기", "아이디 찾기", and "회원가입".

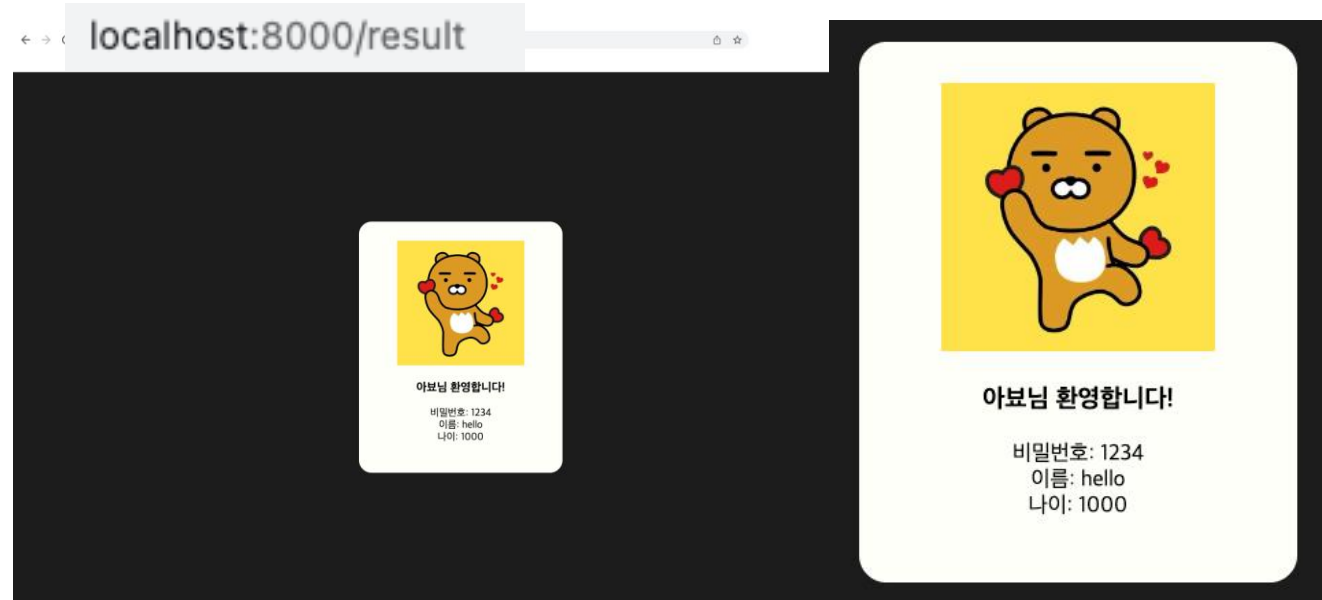
실습31. 파일 업로드

1. 회원가입 일반 폼 전송에 파일 업로드 연결하기 (form submit)
2. 이때, 파일 업로드는 **프로필 사진** 업로드
3. 업로드 후 render 된 페이지에서 업로드 된 이미지 보여주기

← → ↻ ⓘ localhost:8000

파일 업로드 실습31!!!

아이디	
비밀번호	
이름	
나이	
파일 선택	선택된 파일 없음



실습32. SQL - DDL

DDL을 이용하여 아래 테이블을 생성하기

(변경 전) 테이블명: member				
컬럼명	값 형식	기본값	null 여부	기본키
id	varchar(20)		X	O
name	varchar(5)		X	
age	int		O	
gender	varchar(2)		X	
email	varchar(50)		O	
promotion	varchar(2)	"x"	O	

테이블 구조 Slack 제출

```

+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id         | varchar(20)   | NO   | PRI | NULL    |      |
| name       | varchar(5)    | NO   |     | NULL    |      |
| age        | int(2)        | YES  |     | NULL    |      |
| gender     | varchar(2)    | NO   |     | NULL    |      |
| email      | varchar(50)   | YES  |     | NULL    |      |
| promotion  | varchar(2)    | YES  |     | x       |      |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

```

실습33. SQL - DDL

실습32에서 생성한 테이블을 **ALTER** 명령어를 이용해 **구조 변경**하기

(변경 후) 테이블명: member				
컬럼명	값 형식	기본값	null 여부	기본키
id	varchar(10)		X	O
name	varchar(5)		X	
gender	varchar(2)		X	
email	varchar(50)		O	
promotion	varchar(2)	"x"	O	
interest	varchar(100)		O	

*힌트

- **id** 컬럼: 값 형식 변경
- **age** 컬럼: 삭제
- **interest** 컬럼: 추가

Field	Type	Null	Key	Default	Extra
id	varchar(10)	NO	PRI	NULL	
name	varchar(5)	NO		NULL	
gender	varchar(2)	NO		NULL	
email	varchar(50)	YES		NULL	
promotion	varchar(2)	YES		x	
interest	varchar(100)	YES		NULL	

6 rows in set (0.00 sec)

테이블 구조 Slack 제출

실습34. CREATE 문

아래 내용대로 테이블을 생성하기

테이블명 : user

컬럼수 : 6개

컬럼 설명 > 컬럼명, 종류(길이), 비고 순

- 1번째 컬럼 : id VARCHAR(10) not null primary key
- 2번째 컬럼 : pw VARCHAR(20) not null
- 3번째 컬럼 : name VARCHAR(5) not null
- 4번째 컬럼 : gender ENUM('F', 'M', '') default ''
- 5번째 컬럼 : birthday DATE not null
- 6번째 컬럼 : age int(3) not null default 0

테이블 구조 Slack 제출

실습35. INSERT 문

실습34에서 생성한 user 테이블에서 INSERT 문을 이용해 데이터를 넣고
'SELECT * FROM user' 을 이용해 모든 회원 목록을 출력하기

```
회원명부 > id, pw, name, gender, birthday, age 순
```

```
hong1234, 8o4bkg, 홍길동, M, 1990-01-31, 33  
sexysung, 87awjkdf, 성춘향, F, 1992-03-31, 31  
power70, qxur8sda, 변사또, M, 1970-05-02, 53  
hanjo, jk48fn4, 한조, M, 1984-10-18, 39  
widowmaker, 38ewifh3, 위도우, F, 1976-06-27, 47  
dvadva, k3f3ah, 송하나, F, 2001-06-03, 22  
jungkrat, 4ifha7f, 정크랫, M, 1999-11-11, 24
```

테이블 행 내용 캡처 후 Slack 업로드

실습36. SELECT 문

실습34에서 생성한 user 테이블에서 다음 조건을 만족하는 SQL문 작성하기

1. 모든 회원목록을 가져오는데, 이때 birthday 컬럼의 값을 기준으로 오름차순 정렬하여 가져오시오.
2. 회원 목록 중 gender 컬럼의 값이 "M" 인 회원목록을 가져오는데, 이때 name 컬럼의 값을 기준으로 내림차순 정렬하여 가져오시오.
3. 1990 년대에 태어난 회원의 id, name 컬럼을 가져와 목록으로 보여주세요.
4. 6월생 회원의 목록을 birthday 기준으로 오름차순 정렬하여 가져오시오.
5. gender 컬럼의 값이 "M" 이고, 1970 년대에 태어난 회원의 목록을 가져오시오.
6. 모든 회원목록 중 age를 기준으로 내림차순 정렬하여 가져오는데, 그때 처음 3개의 레코드만 가져오시오.
7. 모든 회원 목록 중 나이가 25 이상 50 이하인 회원의 목록을 출력하시오.
8. id 컬럼의 값이 hong1234 인 레코드의 pw 컬럼의 값을 12345678로 변경하시오.
9. id 컬럼의 값이 jungkrat인 레코드를 삭제하시오.

명령어와 결과 캡처 후 Slack 업로드

실습37. MVC(1)

- 동적 폼 전송 수업의 로그인 실습(실습30)을 MVC 구조로 바꾸기

실습30. axios post 로그인

posco x CODINGON

app.js에서 id, pw를 변수로 저장해두고, 로그인 할 수 있게 만들기

이때, 로그인 **axios**의 **post**를 이용하기

Axios 의 결과를 받아와 “로그인“ 버튼 아래 메시지로 보여주기

- 실패 메시지는 빨간 글자
- 성공 메시지는 파란 글자
- Ex) 네이버 로그인 화면



[선택]실습. MVC(2)

- 실습37에서 model/User.js에 아래와 같이 여러 유저 정보를 선언한 후 로그인 성공 여부를 판단하기
 - 데이터 형식 : id//pw//이름
 - 성공 예시) “이름(ex.바나나나나)님 환영합니다.”
 - 실패 예시) “아이디 또는 비밀번호를 잘못 입력했습니다.”

```
// [추가실습]
exports.users = `apple//1234//사과사과
banana//4321//바나나나나
happy//qwer1234//해피해피` ;
```

힌트. split() 함수 사용

ex) string.split("\n") -> 엔터를 기준으로 문자열을 자르고 배열로 반환

실습38. Node.js - MySQL

- 일반 폼 전송
- 회원가입, 로그인, 회원정보 수정, 회원 탈퇴를 DB 와 연동하여 구현
- 이때, CRUD 가 모두 사용
 - Create(**INSERT**) : 회원가입
 - Read(**SELECT**) : 로그인 (회원검색)
 - Update(**UPDATE**) : 회원정보 수정
 - Delete(**DELETE**) : 회원 탈퇴 (삭제)
- (기능 구현 완성 시) CSS로 디자인 적용
 - css 파일은 static 폴더에 생성하기 -> 연결 경로는 /static/css/xxx.css 형식
 - reset.css, nomalize.css, bootstrap, google font 등 활용 가능

실습38. Node.js - MySQL

- user 테이블 구조

```
mysql> DESC user;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id     | int           | NO   | PRI | NULL    | auto_increment |
| userid | varchar(20)   | NO   |     | NULL    |                |
| name   | varchar(10)   | NO   |     | NULL    |                |
| pw     | varchar(20)   | NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

실습38. Node.js - MySQL

• 주소 구조

GET `/user` : 환영 메시지와 함께 회원가입 및 로그인 링크를 보여줌 (`index.ejs`)

- 회원가입 링크 클릭 → GET `/user/signup`
- 로그인 링크 클릭 → GET `/user/signin`

GET `/user/signup` : 회원가입 폼을 보여줌 (`signup.ejs`)

POST `/user/signup` : [회원가입] 버튼 클릭 → 입력 정보 DB 저장 → 성공(true) 응답

GET `/user/signin` : 로그인 폼을 보여줌 (`signin.ejs`)

POST `/user/signin` : [로그인] 버튼 클릭 → DB에서 해당 정보 조회 → 성공(true) or 실패(false) 응답

POST `/user/profile`

- 로그인 성공 시; 로그인 한 사람의 정보를 input에 값을 입력해 보여줌 (`profile.ejs`)
- 로그인 실패 시; `/uesr/signin` 리다이렉트

POST `/user/profile/edit` : 프로필 페이지에서 [수정] 버튼 클릭 → 입력 정보 DB 수정 → '회원정보 성공 수정!' 메시지 응답

POST `/user/profile/delete` : 프로필 페이지에서 [삭제] 버튼 클릭 → 해당 계정 DB 삭제 → `/user/signin` 리다이렉트

• 폴더 구조

```
15_mvc_mysql_practice
├── controller
│   └── Cuser.js
├── model
│   └── User.js
├── node_modules
├── routes
│   └── user.js
├── views
│   ├── 404.ejs
│   ├── index.ejs
│   ├── profile.ejs
│   ├── signin.ejs
│   └── signup.ejs
├── .gitignore
├── app.js
├── init.sql
├── package-lock.json
└── package.json
```

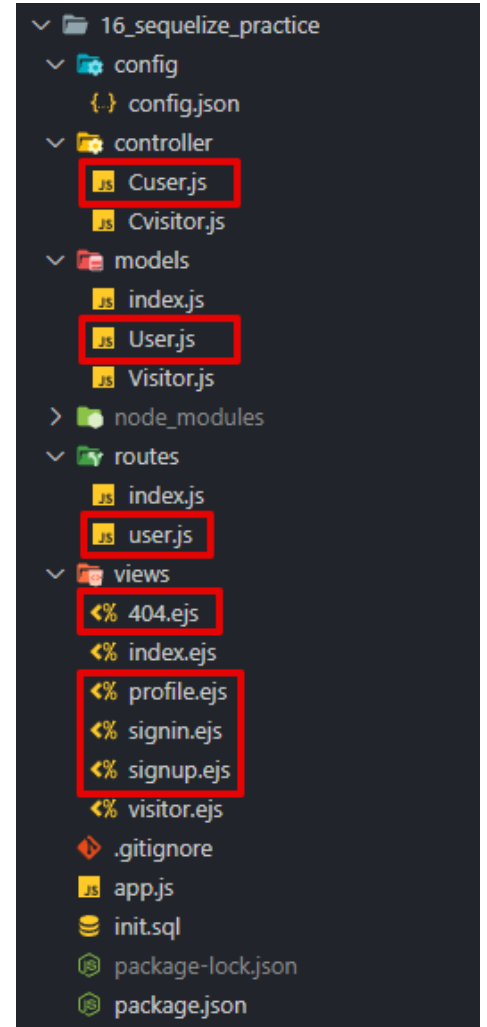
실습38. Node.js - MySQL

- 리다이렉트란?

```
// res.redirect(주소): 리다이렉트할 주소와 함께 응답 보내기!!  
res.redirect('/user/signin');
```

실습39. Sequelize 연습

- 방금 작업한 프로젝트에 회원정보 관리 추가하기
 - 회원가입, 로그인, 회원정보 수정, 회원탈퇴 기능 추가 (실습 38 활용)
 - 이때, **회원**에 대한 **CRUD** 는 모두 **Sequelize** 를 이용하기
 - 폴더구조는 다음과 같이!
 - 방명록과 회원정보 관리 모두 동작하는지 확인



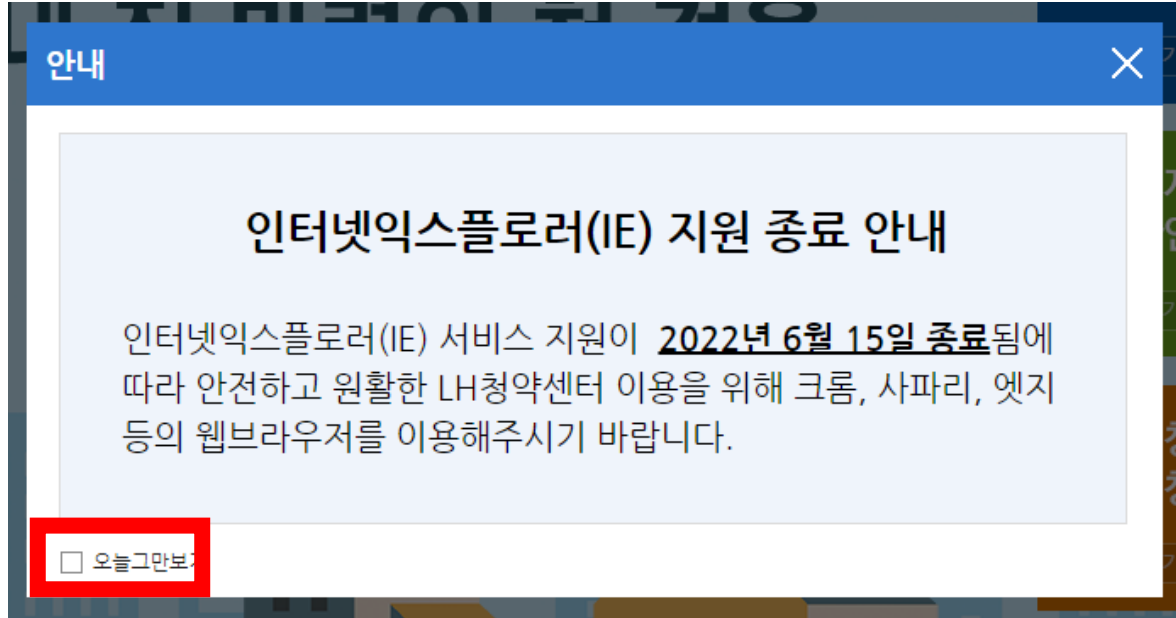
(선택)실습. Sequelize NCP 업로드

- 실습39번 프로젝트 NCP 서버에 업로드해보기
- config/config.json 파일의 production 부분에 코드를 추가하여 NCP 서버에 업로드하기
 - ACG 설정(포트번호) 확인
 - 서버에서 mysql 접속 후 테이블 생성 여부 확인
- models/index.js 파일 config 변수 수정

```
const config = require(__dirname + '/../config/config.json')['production'];
```

```
config > {} config.json > ...
You, 8분 전 | 1 author (You)
1  {
2    "development": {
3      "username": "user",
4      "password": "1234",
5      "database": "kdt",
6      "host": "127.0.0.1",
7      "dialect": "mysql"
8    },
9    "production": {
10     "username": "user",
11     "password": "1234",
12     "database": "kdt",
13     "host": "서버 IP 주소",
14     "dialect": "mysql"
15   },
16   "test": {}
17 }
```

실습40. Cookie 연습



- 페이지 접속했을 때 위와 같은 창 보이게 하고, “오늘 그만 보기” 선택한 후 “닫기” 하면 브라우저를 껐다 켜도 창이 보이지 않게 하기

실습41. Session 연습

- 로그인 한 상태라면 “000님 환영합니다.” 메시지와 로그아웃 버튼 보여주기
- 로그인을 하지 않은 상태라면 “회원가입”, “로그인” 버튼 보여주기

실습42. Socket 연습

- 각 버튼을 누를 때마다 서버로 메시지 전송
- 서버
 - 클라이언트로부터 받은 메시지를 console 에 찍고, 그에 대한 응답을 보내기
- 클라이언트
 - 버튼을 누를 때 서버로 메시지 보내기
 - 서버로부터 받은 응답 메시지를 화면에 보여주기

```
PS E:\Development\github\SeSAC\SeSAC4_web\  
Listening on *:3000  
Server Socket Connected  
PS E:\Development\github\SeSAC\SeSAC4_web\  
PS E:\Development\github\SeSAC\SeSAC4_web\  
Listening on *:3000  
Server Socket Connected  
█
```

Hello Wolrd!

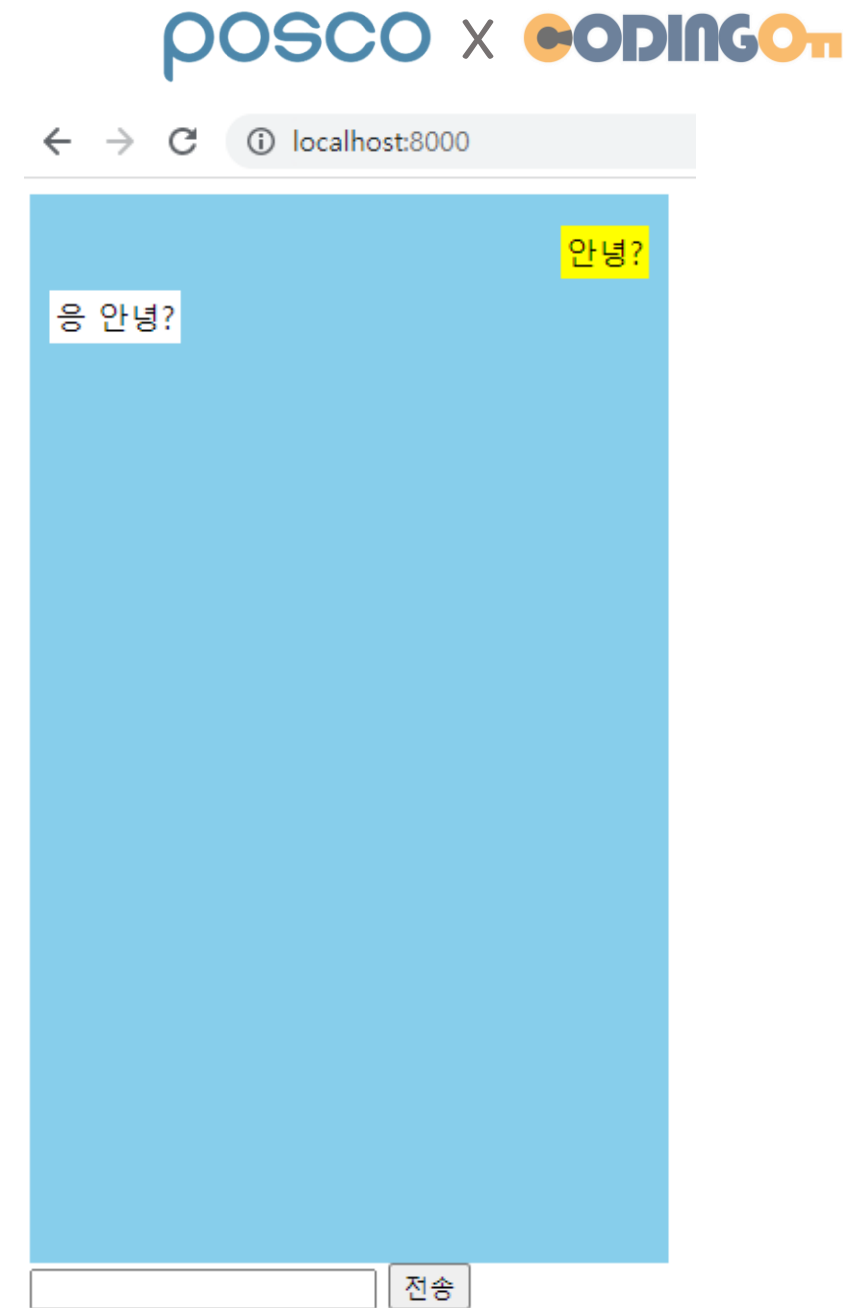
hello study bye

`socket.emit('이벤트명', 정보);` 정보를 보낼 때 사용하는 함수

`socket.on('이벤트명', callback);` 정보를 받을 때 사용하는 함수

실습43. 채팅창 UI 만들기

- 내가 보낸 채팅은 오른쪽에
- 다른 사람이 보낸 채팅은 왼쪽에 표시될 수 있도록



실습44. 채팅창 만들기

- Socket에 담겨 있는 id 정보를 이용하여 000님이 입장했습니다. 라는 메시지 보여주기

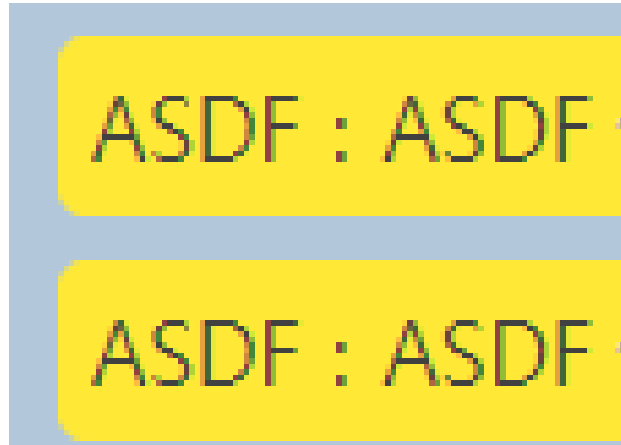
```
io.on( "connection", function ( socket ){
    // 최초 입장했을 때
    console.log( "Server Socket Connected", socket.id );
    io.emit( "notice", `${socket.id}님이 입장하셨습니다.` );
});
```

zEzrHgFXe-XsytExAAAR님이 입장하셨습니다.

전송

실습45. 채팅창 만들기

- 메시지에 누가 작성한 메시지인지 작성자 이름 받고 이름 보여주기
- 아래와 같은 형식으로 보여줘도 OK
 - 작성자 : MESSAGE

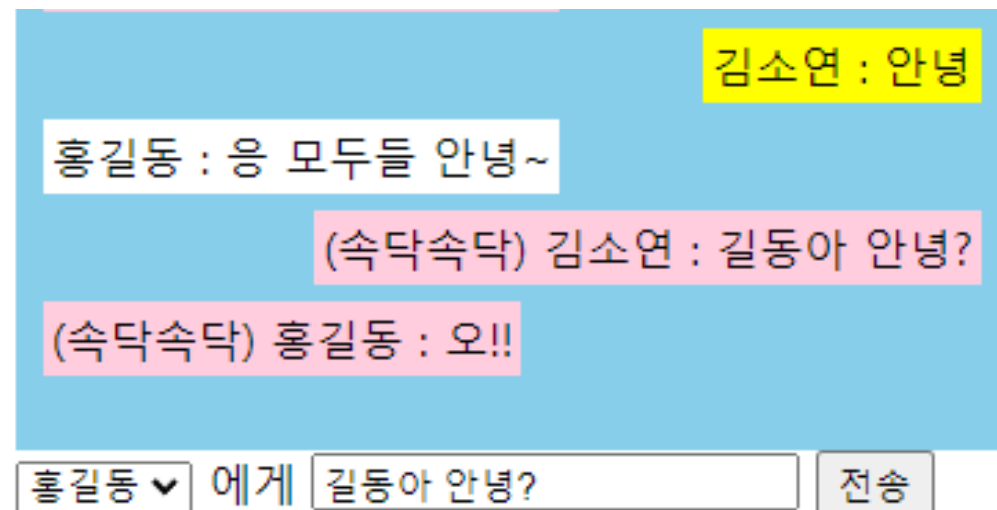


실습46. DM기능 추가하기

- 특정 사람에게만 메시지를 보낼 수 있도록 DM 기능 추가해보기
- DM 메시지는 일반 메시지와 다르게 ui 상으로 변화 주기



A UI mockup showing a light blue header bar. Below it, there is a dropdown menu with '전체' (All) selected. To the right of the dropdown is a text input field with the placeholder '에게' (to) and a '전송' (Send) button.



A UI mockup of a chat conversation window with a light blue background. It shows a sequence of messages: a yellow bubble from '김소연' (Kim Soyeon) saying '안녕' (Hello), a white bubble from '홍길동' (Hong Gildong) replying '응 모두들 안녕~' (Yes, everyone hello~), a pink bubble from '김소연' saying '(속닥속닥) 길동아 안녕?' (Whispering) Gil-dong, hello?), and another pink bubble from '홍길동' replying '(속닥속닥) 오!!' (Whispering) Oh!!). At the bottom, there is a dropdown menu with '홍길동' (Hong Gildong) selected, followed by a text input field with the placeholder '에게' (to) and the text '길동아 안녕?' (Gil-dong, hello?), and a '전송' (Send) button.

실습44. React jsx 실습

- <https://deeplearningon.com/student/index/k-digital-3>
- 접속 후 “React > JSX 실습 진행”

코딩 React



React

0% COMPLETE

리액트 시작하기



작업 환경 설정



JSX란?



JSX 실습



JSX 실습 해답



리액트의 핵심, 컴포넌트



컴포넌트 실습



실습45. React 애벌레 실습

- 이전에 했던 애벌레 실습을 React로 완성하기!!
- 파일을 새로 만들거나 하는 작업은 하지 않아도 됩니당 😊

실습 8. CSS 실습 (1)

- 다양한 position 속성을 이용해 애벌레 만들기

1






2






실습46. React props 실습

- <https://deeplearningon.com/student/index/k-digital-3>
- 접속 후 “React > props 실습 진행”

컴포넌트의 속성, props	
props 실습	
props 실습 해답	

실습47. React state 실습

- <https://deeplearningon.com/student/index/k-digital-3>
- 접속 후 “React > state와 useState 실습” 진행

state와 useState	
state와 useState 실습	
state와 useState 실습 해답	

실습48. 이벤트 핸들링 실습

- <https://deeplearningon.com/student/index/k-digital-3>
- 접속 후 “React > 이벤트 핸들링 실습 진행”
- 함수형으로도 만들어보기

이벤트 핸들링	
이벤트 핸들링 실습	
이벤트 핸들링 실습 해답	

실습49. State & 이벤트 핸들링

검정색 글씨

빨간색 파란색

빨간색 글씨

빨간색 파란색



파란색 글씨

빨간색 파란색



실습50. State & 이벤트 핸들링2

사라져라

안녕하세요

실습51. 컴포넌트 응용 실습

- <https://deeplearningon.com/student/index/k-digital-3>
- 접속 후 “React > 컴포넌트 응용 실습 진행”

컴포넌트 응용	
컴포넌트 응용 실습	
컴포넌트 응용 실습 해답	

- 작성자 :

제목 :

작성

작성자 ▼

검색

번호	제목	작성자
----	----	-----

(선택) 실습53. React 실습

- 실습52에 이어서 “검색” 기능 추가
 - 힌트1. filter() 메서드 사용
 - 힌트2. select 태그 onChange 이벤트

작성자 : 제목 :

번호	제목	작성자
----	----	-----

검색 결과가 없습니다.

실습54. ref 실습

- 실습52에 이어서 기능 추가
 - 이슈: 값을 입력하지 않고 enter키 누르면 테이블에 빈 값이 추가됨
 - 해결: ref를 이용하여 input에 값이 입력되지 않으면 input창에 focus 주기

작성자: 제목:

번호	제목	작성자
1	안녕	민수
2	하이하이	지민
3	멋쟁이	희수



잠깐!

input/textarea에서 조건을 충족하지 못하면 alert 띄우는 것은 UX 측면에서 좋지 않음!!

=> 트렌디한 사이트는 alert를 사용하지 않고 input에 focus 주는 방식을 사용함

실습55. useEffect 실습

- useEffect()를 활용해 가짜 데이터를 Mount 시점에 호출하기!

1. fakePosts 변수에 가짜 데이터를 변수로 저장하기
2. PostList 컴포넌트 Mount 시 setTimeout을 이용해 2초 후, fakePosts 를 state 로 저장하기
3. posts state 길이가 0이라면 loading 메시지 보이기, 길이가 0보다 크다면 post 리스트 보여주기
4. CSS 적용하기

 Post List

No. 1 - sunt aut facere repellat provident occaecati excepturi optio reprehenderit
quia et suscipit suscipit recusandae consequuntur expedita et cum
reprehenderit molestiae ut ut quas totam nostrum rerum...

No. 2 - qui est esse
est rerum tempore vitae sequi sint nihil reprehenderit dolor beatae
ea dolores neque fugiat blanditiis voluptate porro v...

No. 3 - ea molestias quasi exercitationem repellat qui ipsa sit aut
et iusto sed quo iure voluptatem occaecati omnis eligendi aut ad
voluptatem doloribus vel accusantium quis pariatur mole...

No. 4 - eum et est occaecati
ullam et saepe reiciendis voluptatem adipisci sit amet autem
assumenda provident rerum culpa quis hic commodi nesciunt r...

실습55. useEffect 실습

• 동작 실행 결과

Post List
No. 1- sunt aut facere repellat provident occaecati excepturi optio reprehenderit quia et suscipit suscipit recusandae consequuntur expedita et cum reprehenderit molestiae ut ut quas totam nostrum rerum est autem sunt rem eveniet architecto
No. 2- qui est esse est rerum tempore vitae sequi sint nihil reprehenderit dolor beatae ea dolores neque fugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis qui aperiam non debitis possimus qui neque nisi nulla
No. 3- ea molestias quasi exercitationem repellat qui ipsa sit aut et iusto sed quo iure voluptatem occaecati omnis eligendi aut ad voluptatem doloribus vel accusantium quis pariatur molestiae porro eius odio et labore et velit aut
No. 4- eum et est occaecati ullam et saepe reiciendis voluptatem adipisci sit amet autem assumenda provident rerum culpa quis hic commodi nesciunt rem tenetur doloremque ipsam iure quis sunt voluptatem rerum illo velit
No. 5- nesciunt quas odio repudiandae veniam quaerat sunt sed alias aut fugiat sit autem sed est voluptatem omnis possimus esse voluptatibus quisest aut tenetur dolor neque
No. 6- dolorem eum magni eos aperiam quia ut aspernatur corporis harum nihil quis provident sequi mollitia nobis aliquid molestiae perspiciatis et ea nemo ab reprehenderit accusantium quas voluptate dolores velit et doloremque molestiae
No. 7- magnam facilis autem dolore placeat quibusdam ea quo vitae magni quis enim qui quis quo nemo aut saepe quidem repellat excepturi ut quia sunt ut sequi eos ea sed quas
No. 8- dolorem dolore est ipsam dignissimos aperiam dolorem qui eum facilis quibusdam animi sint suscipit qui sint possimus cum quaerat magni maiores excepturi ipsam ut commodi dolor voluptatum modi aut vitae
No. 9- nesciunt iure omnis dolorem tempora et accusantium consectetur animi nesciunt iure dolore enim quia ad veniam autem ut quam aut nobis et est aut quod aut provident voluptas autem voluptas
No. 10- optio molestias id quia eum quo et expedita modi cum officia vel magni doloribus qui repudiandae vero nisi sit quos veniam quod sed accusamus veritatis error

실습55. useEffect 실습

- CSS 적용 결과



Loading...



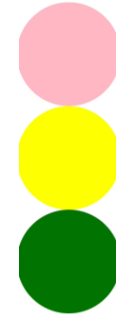
실습56. Sass 실습(1)

- 애벌레 코드의 css를 scss 확장자 파일로 만들어 실행시키기



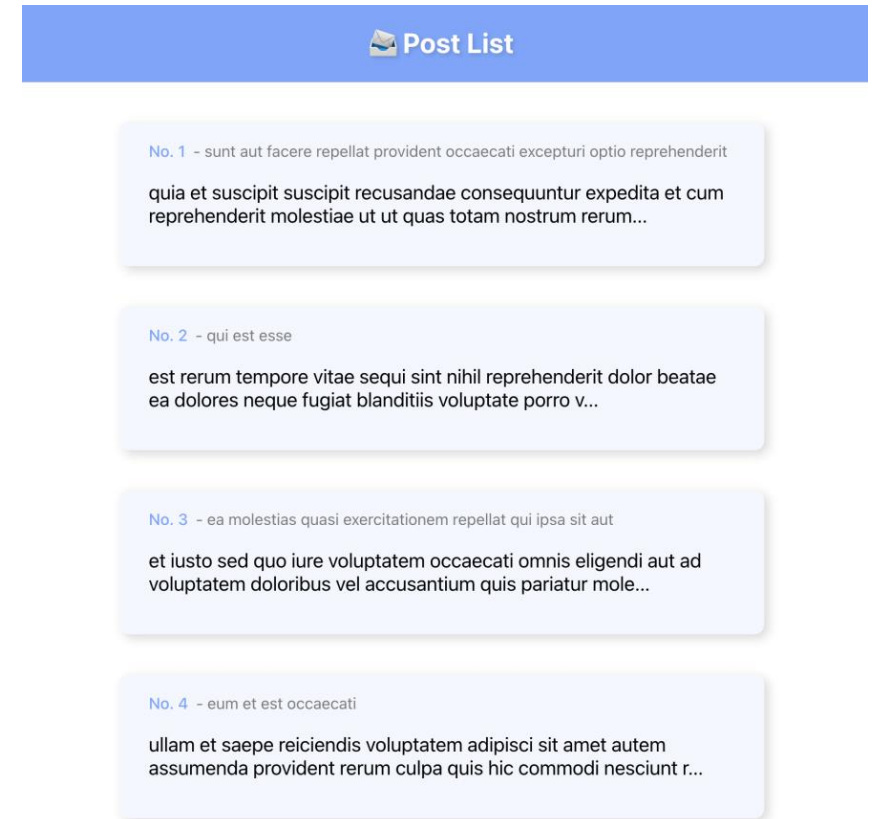
실습57. Sass 실습(2)

- scss 확장자로 만들기
- mixin 이용해 시계방향으로 10만큼 움직이는 핑크 원, 20만큼 움직이는 노란색 원, 30만큼 움직이는 초록 원 만들기
- 이때, 색상은 변수 이용하기



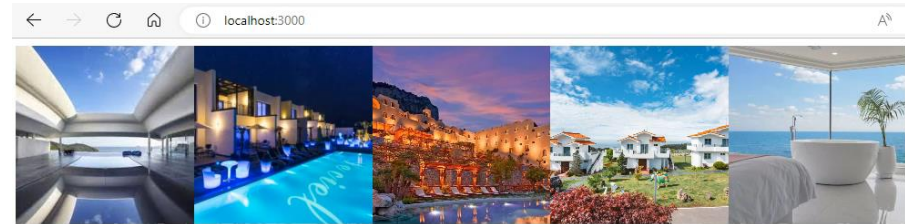
실습58. Sass 실습(2)

- 실습55 이어서 CSS를 SCSS로 변경해 효율적인 코드로 변경하기
 - 색상 변수 사용



[선택] 실습59. Sass 이미지 반응형

- 이미지는 public/images 폴더에 저장하기
- 최소 5개의 이미지 사용하기
- breakpoint
 - 768px 이하에서는 3개씩 보이기
 - 532px 이하에서는 1개씩 보이기



[선택] 실습60. Router 실습

- navbar 코드는 다음과 같을 때,
실행 영상을 참고하여 router 설정하기



```
return (  
  <>  
    <Link to="/">  
      <h2>ReactRouter 실습</h2>  
    </Link>  
    <nav>  
      <ul>  
        <li style={{ margin: '4px' }}>  
          <Link to="/student/kdt">학생KDT</Link>  
        </li>  
        <li style={{ margin: '4px' }}>  
          <Link to="/student/codingon">코딩온</Link>  
        </li>  
        <li style={{ margin: '4px' }}>  
          <Link to="/student/new?name=kdt3rd">searchParams</Link>  
        </li>  
      </ul>  
    </nav>  
  </>  
)  
);
```