모바일웹 실습일지4-2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 학번 | | 201835745 | 이름 | 지서경 | |
| 주 의 사 항 | * VS code에서 create-react-app을 이용하되 지정된 프로젝트명으로 작성할 것 - 반드시 react 코드를 사용할 것. * 결과가 맞다면, 브라우져화면을 캡쳐하고, 코드는 텍스트로 입력하여라. * 코드 텍스트의 포맷은 “맑은고딕 9pt”와 “줄간격 110%”으로 통일시킬 것 * **html코드, js코드, jsx 코드, css코드에서 (줄단위로) 주석을 추가하여 코드를 설명하여라.** | | | | |
| 실 습 결 과 | 실습1) 강의록4-2의 p.4 의 그림과 같이, Airbnb 컴포넌트를 만들어라. 단, 아래조건을 만족하여야 한다.  조건1) props, \*.map함수를 사용하여야 한다. 조건2) 한페이지 전체에 출력되도록 한다. | | | | |
| **<< 실행화면 캡쳐 (첫번째 페이지) >>>**  **텍스트, 전자기기, 모니터, 스크린샷이(가) 표시된 사진  자동 생성된 설명** | | | | **<< 왼쪽페이지 설명 >>**  도시의 사진과 도시에 대한 설명이 중앙 가로축을 기준으로 반반씩 나뉘어서 보여진다. 도시에 대한 설명란에는 도시 이름과 거리가 적혀있다. |
| **<< 실행화면 캡쳐 (두번째 페이지) >>>**  **텍스트, 전자기기, 모니터, 디스플레이이(가) 표시된 사진  자동 생성된 설명** | | | | **<< 왼쪽페이지 설명 >>**  페이지 스크롤을 내리면 위의 사진과 형식은 같지만 다른 도시의 내용을 담고있는 섹션이 보여진다. |

|  |  |  |
| --- | --- | --- |
| 실 습 결 과 | **<< 실행화면 캡쳐 (세번째 페이지) >>>** | **<<소스코드의 디렉터리 구조캡쳐>>**  **(이미지파일등 모두포함)**  **텍스트이(가) 표시된 사진  자동 생성된 설명** |
| **<< 실행화면 캡쳐 (네번째 페이지) >>>** |

|  |  |
| --- | --- |
| 실 습 결 과 | **<< public/index.html 주석포함, 코드넣을 공간이 부족하면 양식을 더 만들어서 추가!** >>  <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8" />  <title>AirBnB</title>  </head>s  <body>  <div id="root"></div> <!--사용자 지정 컴포넌트들이 이곳에 렌더링됨-->  </body>  </html> |
| **<< 리액트 코드 ( 파일명: index.js ) 주석포함, 코드넣을 공간이 부족하면 양식을 더 만들어서 추가!** >>  import React from 'react'; //React와 ReactDOM 임포트  import ReactDOM from 'react-dom/client';  import './index.css'; //index.js에서 쓰일 css 스타일 코드 임포트  import Airbnb from './airbnb'; //airbnb.jsx 에서 만든 Airbnb 컴포넌트 임포트  const root = ReactDOM.createRoot(document.getElementById('root')); //index.html의 div 태그를 id(root)로 가져옴  root.render( //root에 렌더링 될 컴포넌트들 작성  <React.StrictMode>  <Airbnb /> {/\* 임포트한 Airbnb 컴포넌트 배치 (사용자 정의 태그) \*/}  </React.StrictMode>  ); |

|  |  |
| --- | --- |
| 실 습 결 과 | **<< css 코드 (파일명: index.css ) 주석포함, 코드넣을 공간이 부족하면 양식을 더 만들어서 추가!** >>  body {  margin: 0; /\* body의 상하좌우측을 모두 브라우저에 붙임 (margin 없애기) \*/  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',  'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',  sans-serif; /\* 폰트 설정 \*/  /\* 텍스트 브라우저에 부드럽게 렌더링 (webkit 기반의 브라우저에만 적용) \*/  -webkit-font-smoothing: antialiased;  -moz-osx-font-smoothing: grayscale;  } |

|  |  |
| --- | --- |
| 실 습 결 과 | **<< 리액트 코드 (파일명: places.js ) 주석포함, 코드넣을 공간이 부족하면 양식을 더 만들어서 추가!** >>  //각 Area 안에 디스플레이 될 도시 정보들  export const Places = [  //1. 서울  {  img: "seoul.jpeg", //도시 이미지 경로  color: "#de3151", //배경색  title: "서울", //도시 이름  distance: 2, //도시와의 거리  },  //2. 인천  {  img: "incheon.png",  color: "#cc2d4a",  title: "인천",  distance: 29,  },  //3. 대구  {  img: "daegu.jpeg",  color: "#d93b30",  title: "대구",  distance: 237,  },  //4. 대전  {  img: "daejeon.jpeg",  color: "#bc1a6e",  title: "대전",  distance: 140,  },  ] |

|  |  |
| --- | --- |
| 실 습 결 과 | **<< 리액트 코드 (파일명: area.jsx ) 주석포함, 코드넣을 공간이 부족하면 양식을 더 만들어서 추가!** >>  import React from "react"; //React 임포트  import "./area.css"; //area.jsx에서 쓰일 css 스타일 코드 임포트  //하나의 도시 섹션을 보여주는 Area 컴포넌트  function Area(props) { //props를 통해 Airbnb 컴포넌트에서 상품 정보를 받아옴  return(  <div className="areaContainer" style={{backgroundColor: props.color}}> {/\* div태그 (Area 컴포넌트 전체를 감쌈) \*/}  <img src={props.img} className="areaImg" alt="areaImg" /> {/\* img태그 (도시 이미지) \*/}  <div className="infoContainer"> {/\* div태그 (도시 정보를 담는 박스) \*/}  <p className="titleTxt">{props.title}</p> {/\* p태그 (도시 이름) \*/}  <p className="disTxt">{props.distance}km 거리</p> {/\* p태그 (도시와의 거리) \*/}  </div>  </div>  );  }  export default Area; //Area 컴포넌트 (element객체) 내보냄 |

|  |  |
| --- | --- |
| 실 습 결 과 | **<< css 코드 (파일명: area.css ), 코드넣을 공간이 부족하면 양식을 더 만들어서 추가!** >>  .areaContainer {  margin: 0; /\* 바깥쪽 여백을 없앰 \*/  display: flex; /\* 아이템들을 flex 형태로 배치 \*/  flex-direction: column; /\* column(열) 방향으로 배치 \*/  width: 100vw; /\* 현재 실행중인 스크린 너비에 맞춰 상대적 너비를 반환 (전체화면) \*/  height: 100vh; /\* 현재 실행중인 스크린 높이에 맞춰 상대적 높이를 반환 (전체화면) \*/  }  .areaImg {  width: 100%; /\* 넓이를 부모 넓이의 100%로 설정 \*/  height: 50%; /\* 높이를 부모 높이의 50%로 설정 (화면의 윗쪽 반) \*/  object-fit: cover; /\* 가로세로 비율을 유지한 채 사이즈가 조절, 비율이 맞지 않으면 이미지를 확대해 컨테이너 완전히 채움 \*/  }  .infoContainer {  padding: 2rem; /\* 안쪽 여백 설정 \*/  height: 50%; /\* 높이를 부모 높이의 50%로 설정 (화면의 아랫쪽 반) \*/  }  .titleTxt {  margin: 0.8rem 0.5rem 0.8rem 0.5rem; /\* 바깥쪽 여백 설정 \*/  color: white; /\* 글씨 색상 설정 \*/  font-size: 5rem; /\* 글씨 크기 설정 \*/  font-weight: bold; /\* 글씨 굵기 설정 \*/  }  .disTxt {  margin: 0.8rem 0.5rem 0.8rem 0.5rem; /\* 바깥쪽 여백 설정 \*/  color: white; /\* 글씨 색상 설정 \*/  font-size: 2.5rem; /\* 글씨 크기 설정 \*/  } |

|  |  |
| --- | --- |
| 실 습 결 과 | **<< 리액트 코드 (파일명: airbnb.jsx ) 주석포함, 코드넣을 공간이 부족하면 양식을 더 만들어서 추가!** >>  import React from "react"; //React 임포트  import Area from "./area"; //Area 컴포넌트 임포트  import { Places } from "./places"; //도시 정보들을 담고있는 Places 임포트  import "./airbnb.css"; //airbnb.jsx에서 쓰일 css 스타일 코드 임포트  //반복되는 Area 컴포넌트들을 하나씩 배치하는 Airbnb 컴포넌트  function Airbnb(props) {  return(  <div className="wrapper"> {/\* div태그 (Airbnb 컴포넌트 전체를 감쌈) \*/}  {/\* map함수를 이용해 반복되는 컴포넌트 렌더링 \*/}  {/\* Places에서 도시 정보를 받아와 map함수를 이용해 index를 key로 item을 불러오고 item의 각 요소들을 배치함 \*/}  {Places.map((item, index) => {  return(  <Area  key={index} //index를 key로 사용해 어떤 원소가 변경되었는지 빠르게 감지하고 이에 맞는 item을 불러옴  img={item.img} //Places의 index번째 item의 img(item.img)를 img에 넣어줌  color={item.color} //Places의 index번째 item의 color(item.color)를 color에 넣어줌  title={item.title} //Places의 index번째 item의 title(item.title)을 title에 넣어줌  distance={item.distance} //Places의 index번째 item의 distance(item.distance)를 distance에 넣어줌  />  )  })}  </div>  );  }  export default Airbnb; //Airbnb 컴포넌트 (element객체) 내보냄 |

|  |  |
| --- | --- |
| 실 습 결 과 | **<< css 코드 (파일명: airbnb.css ) 주석포함, 코드넣을 공간이 부족하면 양식을 더 만들어서 추가!** >>  .wrapper {  margin: 0; /\* 바깥쪽 여백 없앰 \*/  display: grid; /\* 아이템들을 grid 형태로 배치 \*/  grid-template-rows: 1fr; /\* 그리드 아이템을 동일 크기 행으로 배치 \*/  } |

|  |  |
| --- | --- |
| 실 습 결 과 | 실습2) 아래 그림처럼 웹페이지에 출려되도록, 리액트 컴포넌트를 개발하여라. 단, 위1번과 동일한 조건이다. 영화정보 출처는 <https://www.themoviedb.org/tv/110356?language=ko> 또한 컨텐츠를 4개이상 넣을 수 있도록 개발하여라. |
|  |
|  |
|  |

|  |  |  |
| --- | --- | --- |
| 실 습 결 과 | **<< 실행화면 캡쳐 (첫번째 페이지) >>>** | **<<소스코드의 디렉터리 구조캡쳐>> (이미지파일등 모두포함)**  **텍스트, 전자기기, 스크린샷이(가) 표시된 사진  자동 생성된 설명** |
| **<< 실행화면 캡쳐 (두번째 페이지) >>>** |

|  |  |  |
| --- | --- | --- |
| 실 습 결 과 | **<< 실행화면 캡쳐 (세번째 페이지) >>>**  **텍스트, 스크린샷, 화면이(가) 표시된 사진  자동 생성된 설명** | **<<소스코드의 디렉터리 구조캡쳐>> (이미지파일등 모두포함)**  **텍스트, 전자기기, 스크린샷이(가) 표시된 사진  자동 생성된 설명** |
| **<< 실행화면 캡쳐 (네번째 페이지) >>>**  **텍스트, 스크린샷, 전자기기이(가) 표시된 사진  자동 생성된 설명** |
| **<< 실행화면 캡쳐 (다섯번째 페이지) >>>**  **텍스트이(가) 표시된 사진  자동 생성된 설명** |

|  |  |
| --- | --- |
| 실 습 결 과 | **<< public/index.html 주석포함, 코드넣을 공간이 부족하면 양식을 더 만들어서 추가!** >>  <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8" />  <title>Movie Story</title>  </head>  <body>  <div id="root"></div> <!--사용자 지정 컴포넌트들이 이곳에 렌더링됨-->  </body>  </html> |
| **<< 리액트 코드 ( 파일명: index.js ) 주석포함, 코드넣을 공간이 부족하면 양식을 더 만들어서 추가!** >>  import React from 'react'; //React와 ReactDOM 임포트  import ReactDOM from 'react-dom/client';  import MovieStory from './moviestory'; //moviestory.jsx 에서 만든 MovieStory 컴포넌트 임포트  import './index.css'; //index.js에서 쓰일 css 스타일 코드 임포트  const root = ReactDOM.createRoot(document.getElementById('root')); //index.html의 div 태그를 id(root)로 가져옴  root.render( //root에 렌더링 될 컴포넌트들 작성  <React.StrictMode>  <MovieStory /> {/\* 임포트한 MovieStory 컴포넌트 배치 (사용자 정의 태그) \*/}  </React.StrictMode>  ); |

|  |  |
| --- | --- |
| 실 습 결 과 | **<< css 코드 (파일명: index.css ) 주석포함, 코드넣을 공간이 부족하면 양식을 더 만들어서 추가!** >>  body {  margin: 0; /\* body의 상하좌우측을 모두 브라우저에 붙임 (margin 없애기) \*/  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',  'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',  sans-serif; /\* 폰트 설정 \*/  /\* 텍스트 브라우저에 부드럽게 렌더링 (webkit 기반의 브라우저에만 적용) \*/  -webkit-font-smoothing: antialiased;  -moz-osx-font-smoothing: grayscale;  } |

|  |  |
| --- | --- |
| 실 습 결 과 | **<< 리액트 코드 (파일명: infos.js ) 주석포함, 코드넣을 공간이 부족하면 양식을 더 만들어서 추가!** >>  //각 영화 섹션별로 디스플레이 될 영화 정보들  export const Infos = [  //1. 노 리미트  {  img: "nolimit.jpeg", //영화 포스터 이미지 경로  color: "#4682B4", //배경색  title: "노 리미트", //영화 제목  year: "2022", //개봉년도  genre: "드라마", //영화 장르  synopsis: "놀라운 재능을 지닌 젊은 여성. 세계적인 기록을 보유한 노 리미트 프리 다이빙 강사와 깊고 파괴적인 사랑에 빠진다. 매혹적인 영상미의 로맨틱 드라마 영화.", //영화 개요  },  //2. 기묘한 이야기  {  img: "strangerthings.jpeg",  color: "#708090",  title: "기묘한 이야기",  year: "2016",  genre: "미스터리",  synopsis: "인디애나주의 작은 마을에서 행방불명된 소년. 이와 함께 미스터리한 힘을 가진 소녀가 나타나고, 마을에는 기묘한 현상들이 일어나기 시작한다. 아들을 찾으려는 엄마와 마을 사람들은 이제 정부의 일급비밀 실험의 실체와 무시무시한 기묘한 현상들에 맞서야 한다.",  },  //3. 마이 네임  {  img: "myname.jpeg",  color: "#CD853F",  title: "마이 네임",  year: "2021",  genre: "범죄 미스터리",  synopsis: "아빠를 잃었다. 그것도 바로 눈앞에서. 남은 딸은 결심한다. 반드시 내 손으로 복수하겠노라고. 목표를 위해서라면 방법은 상관없다. 마약 조직의 언더커버가 되어 경찰에 잠입하는 것이라 해도.",  },  //4. 센과 치히로의 행방불명  {  img: "spiritedaway.jpeg",  color: "#FFB6C1",  title: "센과 치히로의 행방불명",  year: "2001",  genre: "애니메이션",  synopsis: "평범한 열 살 짜리 소녀 치히로 식구는 이사 가던 중 길을 잘못 들어 낡은 터널을 지나가게 된다. 터널 저편엔 폐허가 된 놀이공원이 있었고 그곳엔 이상한 기운이 흘렀다. 인기척 하나 없는 이 마을의 낯선 분위기에 불길한 기운을 느낀 치히로는 부모님에게 돌아가자고 조르지만 부모님은 호기심에 들떠 마을 곳곳을 돌아다니기 시작한다. 어느 음식점에 도착한 치히로의 부모님은 그 곳에 차려진 음식들을 보고 즐거워하며 허겁지겁 먹어대다가 돼지로 변해버린다. 겁에 질려 당황하는 치히로에게 낯선 소년 하쿠가 나타나 빨리 이곳을 나가라고 소리치는데...",  },  //5. 허스토리  {  img: "herstory.jpeg",  color: "#FFE4B5",  title: "허스토리",  year: "2018",  genre: "드라마",  synopsis: "남모를 비밀을 간직한 할머니들이 드디어 세상 밖으로 나왔다. 위안부에 대한 사과와 배상을 일본 정부에 요구한 그녀들. 그 용기 있는 6년간의 여정을 그린 감동 실화. 과연 할머니들은 진정한 사과를 받을 수 있을까. 6년의 기간, 23번의 재판, 10명의 원고단, 13명의 변호인. 시모노세키와 부산을 오가며 일본 재판부에 당당하게 맞선 할머니들과 그들을 위해 함께 싸웠던 사람들의 뜨거운 이야기가 시작된다.",  },  ] |

|  |  |
| --- | --- |
| 실 습 결 과 | **<< 리액트 코드 (파일명: movie.jsx ) 주석포함, 코드넣을 공간이 부족하면 양식을 더 만들어서 추가!** >>  import React from "react"; //React 임포트  import "./movie.css" //movie.jsx에서 쓰일 css 스타일 코드 임포트  //하나의 영화 섹션을 보여주는 Movie 컴포넌트  function Movie(props) { //props를 통해 MovieStory 컴포넌트에서 상품 정보를 받아옴  return(  <div className="movieContainer"> {/\* div태그 (Movie 컴포넌트 전체를 감쌈) \*/}  <img src={props.img} className="movieImg" alt="movieImg" /> {/\* img태그 (영화 포스터 이미지) \*/}  <div className="infoContainer" style={{backgroundColor: props.color}}> {/\* div태그 (영화 상세정보를 담는 박스, 배경색은 영화마다 다름) \*/}  <p className="titleTxt">{props.title}<span className="yearTxt">{props.year}</span></p> {/\* p태그 (영화 제목), span태그 (개봉년도) \*/}  <p className="genreTxt">{props.genre}</p> {/\* p태그 (영화 장르) \*/}  <p className="synopTxt">{props.synopsis}</p> {/\* p태그 (영화 개요) \*/}  </div>  </div>  );  }  export default Movie; //Movie 컴포넌트 (element객체) 내보냄 |

|  |  |
| --- | --- |
| 실 습 결 과 | **<< css 코드 (파일명: movie.css ) 주석포함, 코드넣을 공간이 부족하면 양식을 더 만들어서 추가!** >>  .movieContainer {  margin: 0; /\* 바깥쪽 여백 없앰 \*/  display: flex; /\* 아이템들을 flex 형태로 배치 \*/  width: 100vw; /\* 현재 실행중인 스크린 너비에 맞춰 상대적 너비를 반환 (전체너비) \*/  height: 90vh; /\* 현재 실행중인 스크린 높이에 맞춰 상대적 높이를 반환 (전체높이의 90%) \*/  }  .movieImg {  width: 30%; /\* 넓이를 부모 넓이의 30%로 설정 \*/  object-fit: cover; /\* 가로세로 비율을 유지한 채 사이즈가 조절, 비율이 맞지 않으면 이미지를 확대해 컨테이너 완전히 채움 \*/  }  .infoContainer {  padding: 1.7rem 3.8rem 1.7rem 3.8rem; /\* 안쪽 여백 설정 \*/  width: 70%; /\* 넓이를 부모 넓이의 70%로 설정 \*/  }  .titleTxt {  margin: 0; /\* 바깥쪽 여백 없앰 \*/  font-size: 3rem; /\* 글씨 크기 설정 \*/  color: lemonchiffon; /\* 글씨 색상 설정 \*/  }  .yearTxt {  margin-left: 1.5rem; /\* 바깥쪽 여백 설정 \*/  font-size: 1.9rem; /\* 글씨 크기 설정 \*/  color: black; /\* 글씨 색상 설정 \*/  }  .genreTxt {  margin: 0.8rem 0 0.8rem 0; /\* 바깥쪽 여백 설정 \*/  font-size: 1.9rem; /\* 글씨 크기 설정 \*/  }  .synopTxt {  margin: 1.3rem 0 1.3rem 0; /\* 바깥쪽 여백 설정 \*/  font-size: 1.4rem; /\* 글씨 크기 설정 \*/  } |

|  |  |
| --- | --- |
| 실 습 결 과 | **<< 리액트 코드 (파일명: moviestory.jsx ) 주석포함, 코드넣을 공간이 부족하면 양식을 더 만들어서 추가!** >>  import React from "react"; //React 임포트  import "./moviestory.css"; //moviestory.jsx에서 쓰일 css 스타일 코드 임포트  import Movie from "./movie"; //Movie 컴포넌트 임포트  import { Infos } from "./infos"; //영화 정보들을 담고있는 Infos 임포트  //반복되는 Movie 컴포넌트들을 하나씩 배치하는 MovieStory 컴포넌트  function MovieStory(props) {  return(  <div className="wrapper"> {/\* div태그 (MovieStory 컴포넌트 전체를 감쌈) \*/}  <header className="movieHeader"> {/\* header태그 (고정 헤더) \*/}  <h2 className="headerTitle">무비스토리</h2> {/\* h2태그 (페이지 제목) \*/}  </header>  <div className="movieBody"> {/\* div태그 (Movie 컴포넌트가 들어갈 자리) \*/}  {/\* map함수를 이용해 반복되는 컴포넌트 렌더링 \*/}  {/\* Infos에서 영화 정보를 받아와 map함수를 이용해 index를 key로 item을 불러오고 item의 각 요소들을 배치함 \*/}  {Infos.map((item, index) => {  return(  <Movie  key={index} //index를 key로 사용해 어떤 원소가 변경되었는지 빠르게 감지하고 이에 맞는 item을 불러옴  img={item.img} //Infos의 index번째 item의 img(item.img)를 img에 넣어줌  color={item.color} //Infos의 index번째 item의 color(item.color)를 color에 넣어줌  title={item.title} //Infos의 index번째 item의 title(item.title)을 title에 넣어줌  year={item.year} //Infos의 index번째 item의 year(item.year)를 year에 넣어줌  genre={item.genre} //Infos의 index번째 item의 genre(item.genre)를 genre에 넣어줌  synopsis={item.synopsis} //Infos의 index번째 item의 synopsis(item.synopsis)를 synopsis에 넣어줌  />  )  })}  </div>  </div>  );  }  export default MovieStory; //MovieStory 컴포넌트 (element객체) 내보냄 |

|  |  |
| --- | --- |
| 실 습 결 과 | **<< css 코드 (파일명: moviestory.css ) 주석포함, 코드넣을 공간이 부족하면 양식을 더 만들어서 추가!** >>  .wrapper {  margin: 0; /\* 바같쪽 여백 없앰 \*/  }  .movieHeader {  position: fixed; /\* 헤더를 페이지 상단에 고정 \*/  display: flex; /\* 아이템들을 flex 형태로 배치 \*/  align-items: center; /\* 글씨 세로 중앙 정렬 \*/  /\* 헤더가 여백없이 페이지 맨 위쪽에 오게 하기 위해 위, 왼쪽, 오른쪽 여백을 없애줌 \*/  top: 0;  left: 0;  right: 0;  padding: 0 0 0 2rem; /\* 안쪽 여백 설정 \*/  height: 10vh; /\* 현재 실행중인 스크린 높이에 맞춰 상대적 높이를 반환 (전체높이의 10%) \*/  background-color: blueviolet; /\* 배경색 설정 \*/  }  .headerTitle {  margin: 0.5rem; /\* 바깥쪽 여백 설정 \*/  font-size: 2rem; /\* 글씨 크기 설정 \*/  color: white; /\* 글씨 색상 설정 \*/  }  .movieBody {  margin-top: 10vh; /\* 바깥쪽 여백 설정 \*/  display: grid; /\* 아이템들을 grid 형태로 배치 \*/  gap: 0; /\* 그리드 아이템들 사이의 여백 없앰 \*/  grid-template-rows: 1fr; /\* 그리드 아이템을 동일 크기 행으로 배치 \*/  } |