Box Office 조회하기

201924105 김규리

1. 사용 시나리오

* Date 입력 창에 원하는 날짜를 입력한다.  
  ex) 2019-07-18 입력
* 검색 버튼 누르면 2019년 07월 18일의 Box Office Ranking 1~10위가 출력된다.
* 더 알아보고 싶은 영화제목을 클릭하면 감독과 제작사, 개봉일, 출연 배우 등의 정보가 출력되고 youtube에서 검색된 예고편 영상이 재생 준비된다.
* 줄거리 버튼을 클릭하면 줄거리가 알림창으로 출력된다.  
  포스터 버튼을 클릭하면 포스터 이미지가 팝업창으로 출력된다.  
  더보기 버튼을 클릭하면 해당 영화에 대한 한국영화 데이터 베이스 사이트의 공식 검색결과가 열린다.

2. 구현한 서비스와 기능

* BoxOffice를 조회할 날짜를 date input box로 입력 받는다.
  + 조회할 수 있는 가장 오래된 날짜는 2010년 01월 01일이다.
  + Input의 min property를 이용해서 그 이전의 날짜는 입력받지 못하도록 한다.
  + 만약 입력창의 날짜를 수정하지 않고 검색버튼을 누를 경우 2019년 07월 18일이 기본값으로 검색된다.
* 검색 버튼을 누르면 BoxOffice Open Api에 입력한 날짜에 해당하는 BoxOffice 정보를 요청한다.
  + 응답받은 데이터 중 그날의 BoxOffice 순위 1~10위에 해당하는 영화 제목을 화면에 출력한다.
  + BoxOffice 목록의 맨 위에는 검색 결과가 몇년 몇월 몇일의 BoxOffice 순위인지 알려주는 메세지를 출력한다.
* 출력된 BoxOffice 목록 중 하나를 클릭하면 BoxOffice Api의 응답 결과를 바탕으로 해당 영화의 제목과 개봉일 데이터를 얻어온다.
  + BoxOffice Api로부터 응답받은 개봉일 데이터의 형식은 yyyy-mm-dd 형식이다.
  + yyyymmdd형식으로 개봉일 데이터를 바꾸어준다..
  + 클릭한 BoxOffice 버튼의 색깔을 보라색으로 변경한다.
  + 클릭하지 않은 BoxOffice 버튼의 색은 하얀색으로 초기화한다.
  + 영화 상세정보 api와 youtube 재생목록 api에 data를 요청하는 함수를 호출한다.
* 영화 제목과 개봉일 데이터를 바탕으로 영화 상세정보 Open Api에 영화 상세정보 데이터를 요청한다.
  + 영화 제목만으로 상세 정보를 검색하면 동일한 제목의 영화가 검색될 수 있으므로 중복을 피하기 위해 개봉일 데이터도 추가한다.
* showInfo(infoList) : 응답받은 영화정보 데이터를 infoList 변수로 전달하여 화면에 출력하는 함수 showInfo 함수를 호출한다.
  + 영화의 영어 제목과 개봉일을 하나의 문자열로 합쳐서 화면에 출력한다.
  + 배열에 저장된 출연한 배우들의 정보를 하나의 문자열로 합쳐서 화면에 출력한다.
* ‘영화의 제목 + 예고편’을 검색어로 Youtube Open Api에 재생목록 데이터를 요청한다.
  + 한글로 검색하면 에러 발생의 가능성이 있기 때문에 (‘영화제목 + 예고편)을 encode 해서 url로 전달
  + 응답받은 재생목록 중 맨 처음 동영상의 아이디를 youtube 주소와 합쳐서 iframe의 src property 값으로 전달한다.
  + ‘영화 제목 + 예고편’으로 검색했을 때 제일 첫번째로 나오는 동영상이 자동으로 재생 준비된다.
* openPoster() : 포스터 버튼을 누르면 해당 영화의 포스터 이미지가 팝업창으로 열린다.
* openSummary() : 줄거리 버튼을 클릭하면 해당 영화의 줄거리가 알림창으로 출력된다
* openSite() : 더보기 버튼을 클릭하면 해당 영화의 한국영화 database 사이트 검색결과가 새 창에서 열린다.
* 오류 처리
  + 만약 2010년 01월 01일 이전의 날짜가 입력될 경우 알림창을 통해 ‘날짜는 2010년 이후여야 합니다.’라는 메시지와 함께 닐찌 입력창으로 focus를 이동시킨다.
  + 영화 제목을 선택하지 않은 상태에서 줄거리/포스터/더보기 버튼을 클릭할 경우 ‘검색 결과가 없습니다.’라는 메시지와 함께 날짜 입력창으로 focus를 이동시킨다
  + 만약 open api에서 결과를 받아오지 못하면 알림창으로 어떤 api에서 문제가 생겼는지. status가 어떤지 알려준다..

3. 개념 적용 & 응용한 부분

* HTML
  + 시멘틱 태그
    - <header> : 페이지 맨 위 상단에 로고 삽입
    - <section> : 본문에 사용, 결과창의 그룹화
    - <iframe> : JavaScript를 통해 구현되는 태그. 페이지 안에서 Youtube 재생창을 그대로 띄워준다..
    - <footer> : &copy entity를 사용해서 copyright 정보 표시
  + <img> : 동영상 재생되기 전에 youtube 로고 이미지를 삽입
    - src=” image\youtube.png” 속성을 이용해 이미지의 상대적인 주소를 지정
    - alt=“유튜브” 속성을 이용해 이미지가 보이지 않을 때 나타날 텍스트 설정
* CSS
  + css 문서 적용 방법
    - 외부에 따로 style.css 문서 작성하여 <link>사용하여 html 문서 안에 포함시켰다.
  + selector
    - #id를 사용하여 특정 아이디를 가진 element 선택
    - .class를 사용하여 특정 클래스를 가진 element 선택
    - header \* 를 사용하여 header 태그 밑에 있는 모든 자손 태그 선택
    - header span:nth-child(2)를 사용하여 header 태그의 2번째 자식 태그인 span태그 선택, logo에 있는 ‘BoxOffice’ 텍스트 중 ‘Box’ 선택
    - :hover 가상클래스를 이용하여 마우스가 버튼 위에 올랐을 때 버튼 색이 반전되도록 한다.
    - .searcher > #datebox를 사용하여 class가 searcher 인 태그의 자식태그 중 id가 datebox인 태그 선택
  + color와 font
    - font-family property를 이용해서 선택한 element에서 사용할 폰트를 지정한다.
    - font-style을 사용해서 이탈릭체, font-weight를 사용해서 굵은 글씨로 바꾼다.
    - #rrggbb Hex코드를 이용해서 색깔 지정
    - :root를 이용하여 자주 사용하는 색상코드를 변수로 지정.
      * var(변수명)으로 색상을 호출할 수 있도록 함.
  + Border과 margin 속성
    - 줄거리. 포스터 버튼에 padding 속성을 사용하여 button 자체의 크기 변경
    - 버튼에 margin 속성을 사용하여 button과 button 사이의 간격을 조절
  + Flex box
    - element들을 정렬하는 display 방식으로 float가 아닌 flex 박스 사용
    - flex-direction: column 으로 지정하면 내부 element들이 가로로 정렬
  + cursor: pointer
    - 커서가 해당 element 위에 올라가면 화살표에서 포인터로 바뀜
* JavaScript
  + Js 문서 적용 방법
    - script.js로 따로 작성한 js 문서를 <script> 태그와 src attribute를 사용하여 html문서에 포함
  + 변수와 연산자
    - const로 값이 변하지 않는 상수를 선언
    - var로 값이 변할 수 있는 변수를 선언
    - “문자열”과 “문자열”을 + 연산자로 결합시킨다.
  + 함수
    - function showInfo(infoList) : open api에서 응답받은 data를 infoList라는 이름의 parameter로 함수에 전달
    - showInfo(infoList) : 함수명(매개변수) 꼴로 작성한 함수를 호출
    - preDate.split("-") : preDate라는 이름의 문자열을 ‘-‘문자를 기준으로 잘라서 배열로 만든다.
    - date.focus(); : focus를 date변수에 지정한 element로 이동시킴
    - alert("검색결과가 없습니다.") : 알림창에 “문자열” 내용을 출력
    - boxBtn.addEventListener('click', getTitleAndDate); : boxBtn라는 변수에 지정된 element를 클릭할 경우 getTitleAndDate함수를 실행하도록 한다.
    - temp.setAttribute(‘style’, ‘속성’) : temp라는 이름으로 지정된 element의 속성을 지정한다.
    - list.appendChild(temp); : list라는 이름으로 지정된 element 내부에 temp를 추가한다.
    - var OpenWindow = window.open(pstUrl, '\_blank', 'width = ' + 250 + ', height = '+ 350 +', menubars = no, scrollbars = auto');
      * 전달한 pstUrl을 새 창에서 열리게 하는 함수
      * Width와 height를 지정하였기 때문에 팝업창으로 열림
      * \_blank로 지정하였기 때문에 창을 끄지 않고 포스터 버튼을 누르면 새 창이 계속해서 생기는 것을 알 수 있다.
      * \_blank가 아닌 다른 이름으로 지정하면 새 창이 열리지 않고 이미 열려있는 창에서 새로 로드되는 것을 알 수 있다.
  + if(dateArray[0]<2010) {  
     alert("날짜는 2010년 이후여야 합니다.");  
     return;}
    - if(조건문)으로 조건문의 결과가 true일 때 실행할 코드 작성
    - alert() 함수로 알림창 띄우기
    - return; 으로 함수 종료
  + for(var i=0; i < dateArray.length; i++) {  
     targetDate += dateArray[i];}
    - for 반복문을 사용해서 targetDate라는 변수에 dateArray라는 배열의 값을 저장
    - dateArray.length로 배열의 길이를 받아 자동으로 모든 배열이 저장되도록 한다.
  + jQuery & Ajax & json
    - const info = $('#metainfo') : id가 metainfo인 element 선택한다.
    - $(list).empty() : list를 초기화하는 함수
    - const xhr1 = new XMLHttpRequest();  
      xhr1.open(method, url);  
      xhr1.send();
      * open api 서버에 json data를 요청하는 함수
    - xhr1.onload = () => {  
       var data = xhr1.response;  
       boxData = data.boxOfficeResult.dailyBoxOfficeList;};
      * 받아온 json data를 쓰기 쉬운 형태로 변경해서 변수에 저장

❖ 사용한 Open Api (중간보고서와 달라졌습니다.)

영화 박스오피스 DB

영화 정보 DB

Youtube Data Api