

<YOUTUBE 페이지 코딩하기>

1. index.html

```
index.html X JS script.js
YoutubeApi > index.html > html > body > div.infoAndUpnext > section.info > div.metadata
14 <div class="menu">
15 <i class="fas fa-bars"></i>
16 </div>
17 <div class="logo">
18 <i class="fab fa-youtube"></i>
19 <span class="ylogo">YouTube</span>
20 </div>
21 <!-- 검색창 추가 -->
22 <div class="search">
23 <input type="text" id="search" name="search" placeholder="검색">
24 <button id="button">
25 <i class="fas fa-search"></i>
26 </button>
27 </div>
28 <!-- 아이콘 추가 -->
29 <div class="icons">
30 <i class="fas fa-video"></i>
31 <i class="fas fa-th"></i>
32 <i class="fas fa-bell"></i>
33 </div>
34 </header>
35
36 <!-- video player -->
37 <section id="videoPlayer">
38 <!-- 비디오 플레이어 iframe 태그로 변경 -->
39 <iframe src="//www.youtube.com/embed/67stn7Pu7s4?autoplay=1" width="100%" height="400px" frameborder="0" allowfullscreen></iframe>
40 </section>
41
```

- Header 부분에 검색창과 아이콘 추가
- Videoplayer를 iframe 태그로 변경

2. script.js

```
index.html JS script.js X
YoutubeApi > JS script.js > [e] optionParams
1 //Youtube open api 이용, 검색기능 사용하기
2 const search = document.getElementById('search');
3 const button = document.getElementById('button');
4 const v = document.getElementById('videoPlayer');
5 const vTitle = document.getElementById('vTitle');
6 const cName = document.getElementById('cName');
7 const date = document.getElementById('date');
8
9 var optionParams = {
10   part:"snippet",
11   key:"AIzaSyAcYCF3S1cnwmgS4wfyiY59_3Aw0yKR1MA",
12   type:"video",
13   maxResults:10,
14   regionCode:"KR",
15   videoDuration:"short"
16 };
17
18 var url="https://www.googleapis.com/youtube/v3/search?";
19 for(var option in optionParams){
20   url += option + "=" + optionParams[option] + "&"; //open api에 요청할 url값 초기화
21 }
22
23 button.addEventListener('click', getInputData); //검색 버튼 눌렀을 경우 getInputData 실행
24
```

```

index.html JS script.js X
YoutubeApi > JS script.js > [0] optionParams
24
25 function getInputData() {
26     var yUrl = url;
27     var name = "";
28     if(search.value=="") {
29         alert("검색어를 입력하세요.");
30         search.focus();
31         return false; //검색어가 입력되지 않았을 경우 알림창 띄우기
32     } else {
33         console.log(search.value);
34         name = encodeURIComponent(search.value);
35         yUrl += "q=" + name;
36         console.log(name);
37         console.log(yUrl); //검색어가 입력되었을 경우 url에 포함시켜서 전달
38     }
39
40     sendHttpRequest('GET', yUrl); //json type으로 데이터 받아오는 함수 호출
41 }
42
43 function sendHttpRequest(method, url) {
44     var xhr = new XMLHttpRequest();
45     xhr.open(method, url);
46     xhr.responseType = 'json';
47     xhr.onload = () => {
48         var data = xhr.response;
49         dataList = data.items;
50         jsonData = data.items[0].id.videoId;
51         console.log(jsonData);
52         updateVideo(jsonData); //videoPlayer에서 재생할 video 보여주는 함수
53         updateInfo(data.items[0]); //videoPlayer에서 재생할 video의 제목과 channel 보여주는 함수
54         addList(dataList); //다음 비디오(upNext)의 썸네일 목록 불러오는 함수
55     };
56     xhr.send();
57 }
58
59 function updateVideo(id) {
60     v.innerHTML = '<iframe src="//www.youtube.com/embed/' + id + '?autoplay=1" width="100%" height="400px" frameborder="0" allowfullscreen></iframe>';

```

```

index.html X JS script.js X
YoutubeApi > JS script.js > [0] optionParams
56     xhr.send();
57 }
58
59 function updateVideo(id) {
60     v.innerHTML = '<iframe src="//www.youtube.com/embed/' + id + '?autoplay=1" width="100%" height="400px" frameborder="0" allowfullscreen></iframe>';
61     //open api를 통해서 받아온 video의 id를 주소와 합쳐서 iframe에 넣김
62 }
63
64 function updateInfo(item) {
65     vtitle.innerHTML = item.snippet.title + " | " + item.snippet.description;
66     //videoPlayer에서 재생할 video의 title과 description을 vtitle 자리에 출력
67     cname.innerHTML = item.snippet.channelTitle;
68     //channel의 name을 channel의 metadata자리에 출력하기
69     date.innerHTML = item.snippet.publishTime.substr(0,10);
70     //video의 업로드 시간 중 년월일 부분만을 출력
71 }
72
73 function addList(itemList) {
74     var i;
75     for(i=1; i<4; i++) { //i=0은 main player에서 재생되는 영상, 목록 업데이트는 1번부터 진행
76         var item = document.getElementById("item"+i);
77         var thum = document.getElementById("thum"+i);
78         var title = document.getElementById("title"+i);
79         var channel = document.getElementById("channel"+i);
80         var thumImg = itemList[i].snippet.thumbnails.medium.url;
81         thum.innerHTML = "<img src='" + thumImg + "' style='max-width: 100%; height: auto;'>";
82         //다음 동영상 목록에 검색 결과로 나온 video의 썸네일 사진 출력하기
83         title.innerHTML = itemList[i].snippet.title; //동영상 목록의 title 업데이트
84         channel.innerHTML = itemList[i].snippet.channelTitle; //동영상 목록의 channel name 업데이트
85         item.appendChild(thum);
86     }
87 }
88

```

- Youtube open api를 사용해서 제한적이지만 실제 동영상을 검색하는 기능을 넣었습니다.
- 검색버튼을 누르면 실제 Youtube 사이트에서 동영상을 검색해서 제일 첫번째 동영상은 videoPlayer에서 재생하도록 하고 나머지 목록 중 3개는 다음 동영상 목록에 띄우도록 했습니다.

- ♦ 다만 동영상과 관련된 정보는 동영상의 제목과 채널 이름, 썸네일 이미지만 가져오게 하였습니다..

이 외에도 구현해보고 싶은 기능들이 많지만 시험 기간이라 시간이 촉박하여 조금 생략했습니다. 시험 기간이 끝나면 기능을 추가해서 다시 한 번 만들어보겠습니다.

감사합니다. 복 많이 받으세요!