

비대면 환경에서의 효과적인 교육 및 비즈니스를 위한 개인 맞춤형 융합 콘텐츠 생성 기반 기술

12주차 발표

김이홍조

2016112166 김운호

2017112085 이유경

2017112099 조민지

2017111718 홍은주

목차

1. 진행 상황 보고

2. 차후 진행 계획

자막 종류 및 구조

1) 자막 종류 및 구조

- .smi

- 1) Synchronized Accessible Media Interchange
- 2) 동영상 파일과 자막 파일의 이름이 동일해야 자막이 표시됨
- 3) 텍스트 편집기로 작성/편집 가능
- 4) HTML 태그 및 CSS 지원 : 폰트, 글씨색깔 등 적용 가능
- 5) 태그를 이용한 통합 자막 제작 가능 : 한글자막KRCC과 영자막ENCC 동시 출력 가능
- 6) 종료 싱크가 없음 : 다음 자막 출력까지 앞 자막이 계속 보여짐
- 7) 거의 한국에서만 사용, 호환성 낮음

자막 종류 및 구조

1) 자막 종류 및 구조

- .smi 구조

```
<SAMI>
  <HEAD>
    <Title> 자막제목 </Title>
    <STYLE TYPE = "text/css">
      <!--
      P{margin-left : 8pt; margin-right : 8pt; font-family : 굴림, Arial; font-weight : bold}
      .KRCC{name : 한국어; lang : kr-KR; SAMIType : CC;}
      .ENCC{name : English; lang : en-US; SAMIType : CC;}
      -->
    </STYLE>
  </HEAD>
  <BODY>
    <SYNC Start=0>
    <P Class=KRCC>한국어 자막</P>
    <P Class=ENCC>English Subtitle</P>
  </BODY>
</SAMI>
```

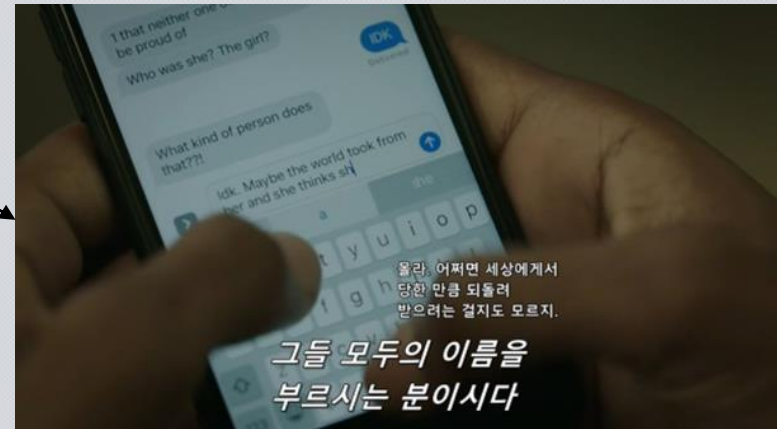
- <SYNC start (sec)> 시간, <p/> 자막, 시간은 sec으로만 표시 가능
- <BODY> 태그 안 내용 인식 탁월
- <SYNC start=싱크>가 핵심

자막 종류 및 구조

1) 자막 종류 및 구조

- .srt

- 1) 서버립 SubRip이라는 자막 추출 프로그램의 텍스트 자막 파일 확장자
- 2) 높은 호환성 : 유니코드 형식. 한자 등 글씨 깨짐 없음. 전세계적으로 사용.
- 3) HTML 태그 사용 가능
- 4) 형식 단순
- 5) 자막 별 시작, 종료 시간 설정 가능
- 6) 다른 종류의 자막도 동시 출력 가능



자막 종류 및 구조

1) 자막 종류 및 구조

- .srt 구조

```
1
00:00:00-->00:01:30
Hello
2
00:02:00-->00:03:00
world
```

- 순번
시간(시간:분:초,밀리초)
자막
으로 구성, 순번은 생략 가능

자막 종류 및 구조

1) 자막 종류 및 구조

- .vtt

- 1) 웹브라우저 상에서 지원하는 HTML5 자막 포맷 WebVTT
- 2) MIME 타입은 text/vtt
- 3) 파일 내에 주석 포함 가능
- 4) CSS 기반 스타일 지정 가능

자막 종류 및 구조

1) 자막 종류 및 구조

- .vtt 구조

```
0
00:00:01.000 --> 00:00:02.500
Hello
1
00:00:02.500 --> 00:00:03.000
world
```

- 순번은 생략 가능
- 시작 시간, 종료 시간 지정 가능



Smi같은 파일의 경우 간단하게 만들기에 한계가 있음.

Srt, vtt파일이 시간과 텍스트만으로 간단하게 생성 가능하기 때문에 두 종류의 파일을 이용하는 것으로 결정

webODF 사용

1) Python을 이용한 로컬 서버 사용

```
C:\Users\홍은주\OneDrive - dongguk.edu\DGU\2021_8\종설>py -m http.server  
Serving HTTP on :: port 8000 (http://[::]:8000/) ...  
|
```

보안이슈로 인해 일반 html 파일을 열때 webODF.js 로딩 불가



로컬서버 이용하여 해결

추후 사용시에는 서버 구축 후 사용할 것이므로 문제 없을것으로 예상

webODF 사용

2) html 파일 작성

```
1 <html>
2   <head>
3     <script src = "index.js" type="text/javascript"></script>
4     <script src = "webodf.js" type="text/javascript" charset="utf-8"></script>
5     <script type="text/javascript" charset="utf-8">
6       function init() {
7         var odfelement = document.getElementById("odf"),
8         odfcanvas = new odf.OdfCanvas(odfelement);
9         odfcanvas.load("myfile.odt");
10      }
11      window.setTimeout(init, 0);
12    </script>
13  </head>
14  <body>
15    <div id="odf"></div>
16  </body>
17 </html>
```

다운받은
webodf.js
라이브러리 사용

webODF 사용

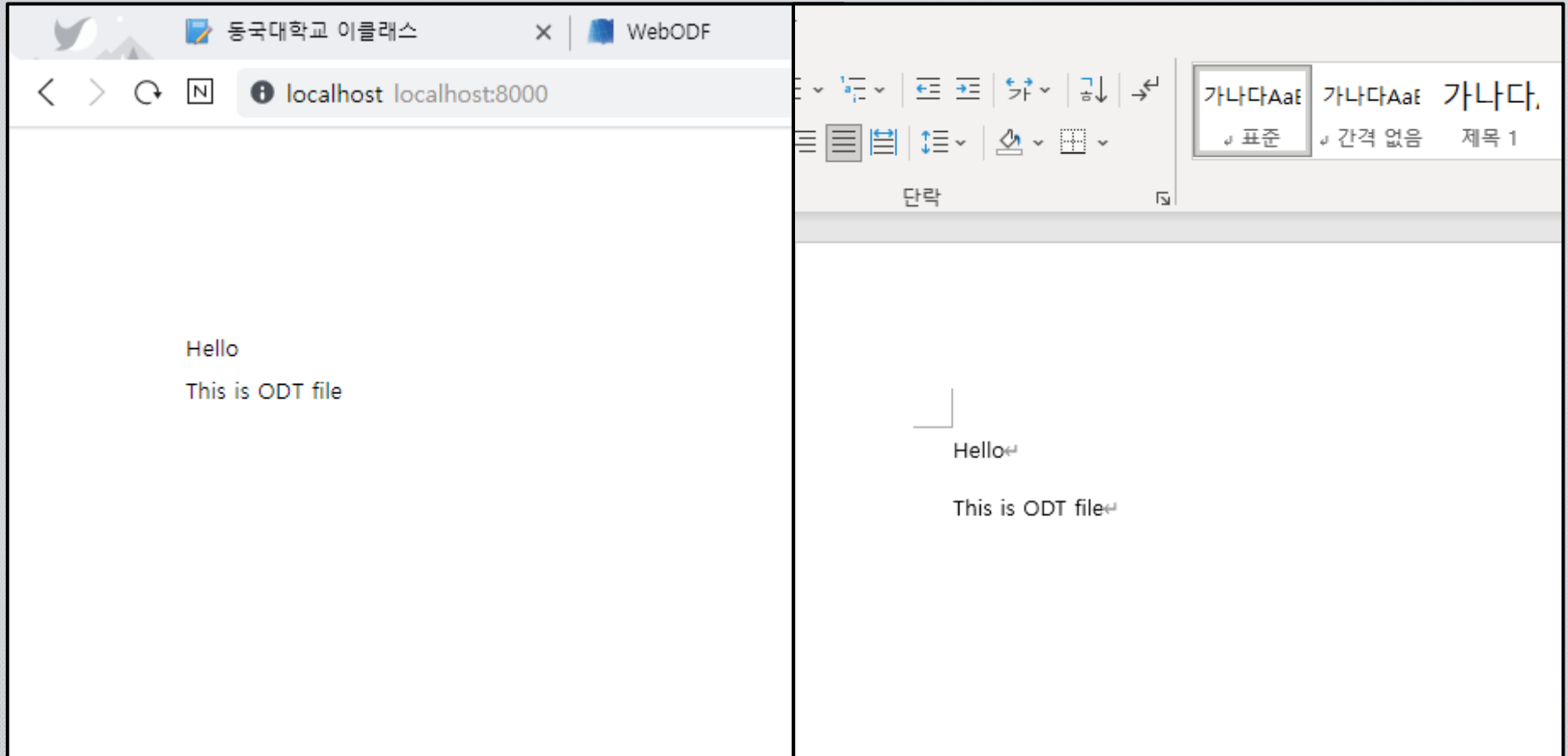
2) html 파일 작성

```
1  <html>
2  <head>
3      <script src = "index.js" type="text/javascript"></script>
4      <script src="webodf.js" type="text/javascript" charset="utf-8"></script>
5      <script type="text/javascript" charset="utf-8">
6          function init() {
7              var odfelement = document.getElementById("odf"),
8                  odffcanvas = new odf.OdfCanvas(odfelement);
9                  odffcanvas.load("myfile.odt");
10             }
11         window.setTimeout(init, 0);
12     </script>
13 </head>
14 <body>
15     <div id="odf"></div>
16 </body>
17 </html>
```

#odfelement에 odffcanvas(객체) 생성
→ load 함수를 이용해 로컬의 파일 불러오기

webODF 사용

3) 결과 확인



결과가 제대로 출력되는 것을 확인

Textract 사용

1) 설치

```
C:\Users\홍은주\OneDrive - dongguk.edu\DGU\2021_8\종설>npm install -g textract
```

npm 설치 후, npm i -g textract 명령을 통해 textract 설치
이때, -g(글로벌) 사용

2) 사용

```
C:\Users\홍은주\OneDrive - dongguk.edu\DGU\2021_8\종설>textract pathToFile
```

pathToFile에 파일 경로 입력

Textract 사용

2) 결과

```
C:\Users\홍은주\OneDrive - dongguk.edu\DGU\2021_8\종설>textract myfile.odt
Hello
This is ODT file

1
2
3
4
5
6
-----myfile
```

파일 내 텍스트만 제대로 추출 되는 것을 확인함

자막화 방법 모색



파일에서 텍스트 추출



웹 페이지로 가져오기



사용자가 시간정보를
가져올 수 있는 버튼 기능

시간정보와 매칭시켜 자막화

동영상의 시간 정보 가져오기

currentTime

```
<script>
    var vid = document.getElementById("myVideo");

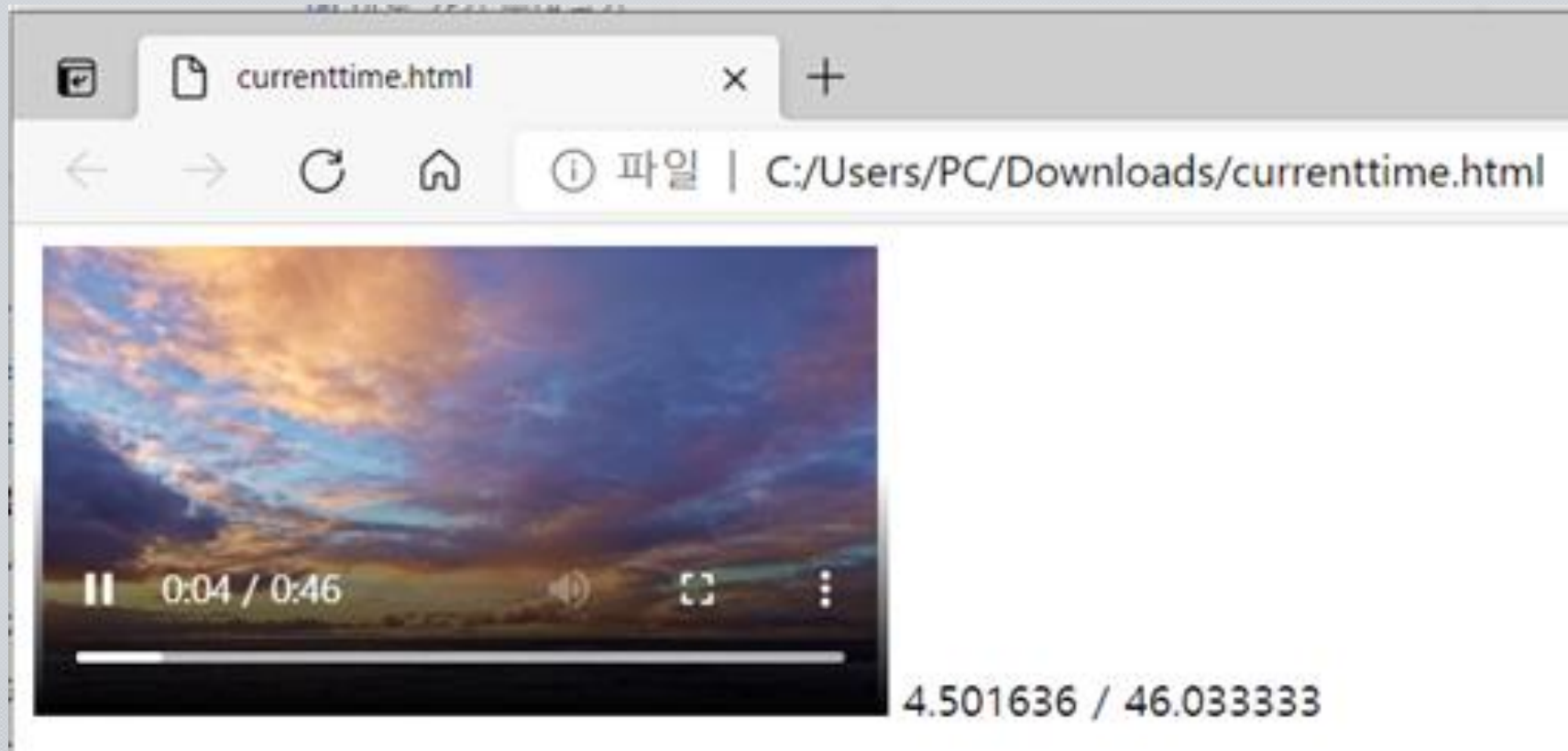
    vid.ontimeupdate = function() {myFunction2() };

    function myFunction2()
    {
        document.getElementById("test").innerHTML
            = vid.currentTime + "/" + vid.duration;
    }

</script>
```


동영상의 시간 정보 가져오기

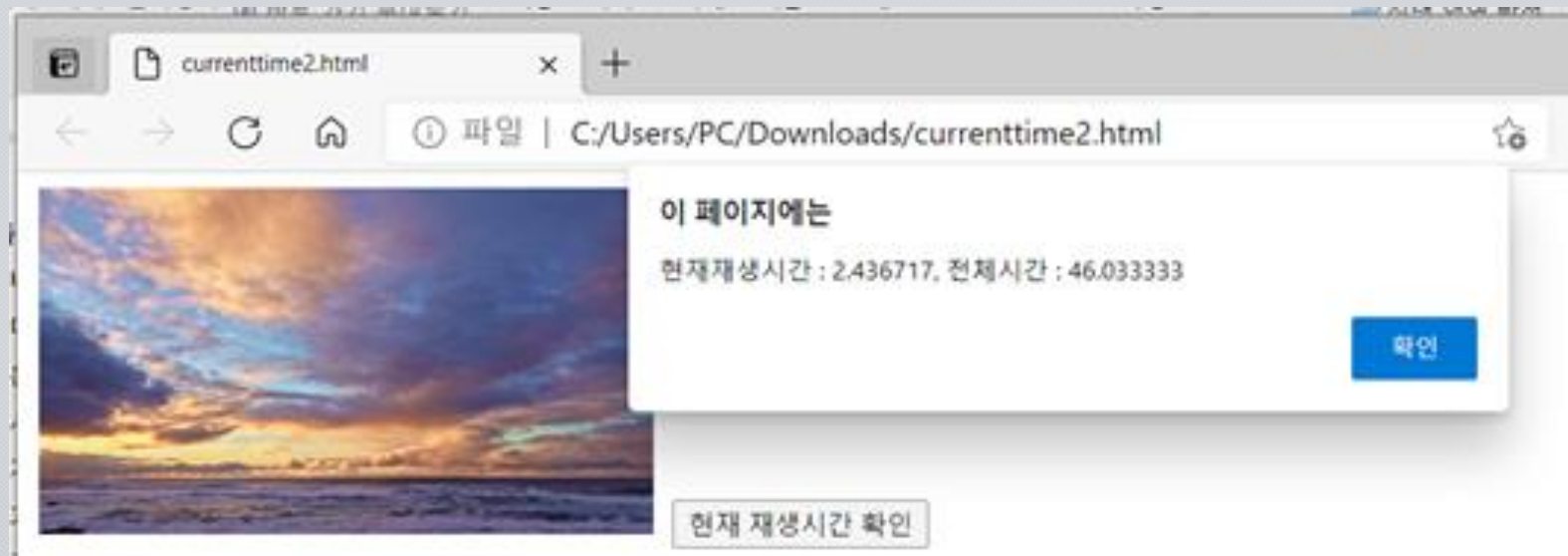
currentTime



동영상의 시간 정보 가져오기

currentTime

```
<button onclick="myFunction()" type="button">  
    현재 재생시간 확인  
</button>
```



차후 진행 계획

멘토링 (19일 수요일 예정)

- 현재까지 진행 상황 보고 및 검토
- 구현 방향성 확정

특허출원명세서 작성

- 진행한 내용들을 바탕으로 특허출원 명세서 작성 시작

감사합니다

김이홍조 12주차 발표
