

Opinnäytetyö (AMK)

Tietotekniikka

Mediatekniikka

2014

Markus Räikkä

# VIDEON KÄYTTÖMAHDOLLISUUDET WWW-YMPÄRISTÖSSÄ

– virtuaalisen video-oppimisympäristön luominen



TURUN AMMATTIKORKEAKOULU  
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietotekniikka | Mediatekniikka

2014 | Sivumäärä 37

yliopettaja, FT Mika Luimula

Markus Räikkä

## VIDEON KÄYTTÖMAHDOLLISUUDET WWW-YMPÄRISTÖSSÄ

Insinööriyön tavoitteena oli tuottaa video-oppimisympäristö Turun yliopiston lääketieteen tiedekunnan opiskelijoiden käyttöön. Yliopiston lähtökohtana oli antaa opiskelijoille mahdollisuus syventää luennoilla saamia tietoja verkossa ja näin laajentaa opiskelutapoja. Oppimisympäristöä on tarkoitus pystyä muokkaamaan jälkepäin lisäämällä ja vaihtamalla sisältöä.

Teoriaosuudessa paneuduttiin www-trendien historiaan ja nykypäivään. Käytiin läpi HTML5:n tuomat uudet ominaisuudet ja se miten niitä käyttämällä pystyy esittämään interaktiivisella tavalla lisätietoa videon aikana. Lisäksi tutkittiin muita vastaavanlaisia video-oppimisympäristöjä joita on tarjolla. Vertailtiin niiden ominaisuuksia, käytettävyyttä, ulkoasua ja toiminnallisuutta ja pohdittiin miten tämä vertailu on vaikuttanut insinööriyöni suunnitteluun ja toteutukseen. Teoriaosuuden loppuvaiheessa käytiin läpi videosoittimen toteutus käyttäen HTML5 ja JavaScript -kieltä.

Video-oppimisympäristöjä luodessa on huomioitava sivuston käytettävyys ja sisällön vaihtaminen ja lisääminen on ylläpitäjälle oltava helppoa ja nopeaa. Video-oppimisympäristöt tuovat uusia mahdollisuuksia opetukseen ja monipuolistavat oppilaiden tapoja oppia. Uusien oppimisympäristöjen avulla pystytään nykyteknologia tuomaan osaksi opetusta.

ASIASANAT:

video-oppimisympäristö, JavaScript, HTML5

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information technology | Media Technology

2014 | Total number of pages 37

Principal Lecturer, Ph.D. Mika Luimula

Markus Räikkä

# POSSIBILITIES OF VIDEO-BASED LEARNING ON THE WEB

The purpose of this thesis was to produce a video-based learning environment for medical students at the University of Turku, and this environment should be easily adjustable afterwards. The purpose of the University of Turku was to give their students the opportunity to deepen their knowledge and skills online on topics covered in lectures and this way extend their learning techniques.

The theoretical part describes web design trends, past and present. This part describes the new features of HTML5 and how they provide an interactive way to show information during the video. The thesis also introduces other similar video -based learning environments that are available compares their features, usability, appearance and functionality, and discusses the project design and implementation. Finally the practical part of this thesis presents the video player design and implementation using HTML5 and JavaScript languages.

When creating a video-based learning environment, it is important to consider usability and adaptability. In other words, the administrator should be able to edit and add content easily and quickly. Video-based learning environments create new opportunities for teaching and enrich the students' ways of learning. New learning environments help to bring modern technology into teaching.

KEYWORDS:

video-based learning environment, HTML5, JavaScript

# SISÄLTÖ

<b>1 JOHDANTO</b>	<b>7</b>
<b>2 WWW-SIVUSTOJEN TEKEMINEN ENNEN JA NYT</b>	<b>9</b>
2.1 WWW-trendien historiaa	9
2.2 HTML5	10
2.3 HTML5:n uudet ominaisuudet	11
<b>3 HTML5:N VIDEOSOITTIMEN HALLINTA JAVASCRIPTILLÄ</b>	<b>13</b>
3.1 Video elementin rakenne	13
3.2 Videon hallintafunktiot	14
3.2.1 Videoformaatin valintafunktio	14
3.2.2 PlayPause-funktio	15
3.2.3 Äänenvoimakkuuden liukusäädinfunktio	15
3.2.4 Vidmute-funktio	16
3.2.5 JumpToTime-funktio	16
3.2.6 Kokonäyttö-funktio	17
<b>4 VASTAAVANLAISET OPPIMISYMPÄRISTÖT JA NIIDEN OMINAISUUDET</b>	<b>18</b>
4.1 Yhteenveto oppimisympäristöistä	24
<b>5 VIDEO-OPPIMISYMPÄRISTÖN SUUNNITTELU</b>	<b>26</b>
5.1 Oman ympäristön ominaisuudet	26
5.2 Videosoitin suunnittelu	28
<b>6 VIDEO-OPPIMISYMPÄRISTÖN TOTEUTUS</b>	<b>29</b>
6.1 Vaihtuvien tekstien lisääminen	29
6.2 Thumbnail-kuvien lisääminen ja niiden ajankohdan määrittäminen	30
6.3 Videosivun artikkelihaku	30
6.4 Hakusivun ominaisuudet	31
6.5 WordPressin käyttö sisällönhallintajärjestelmänä	31
6.6 Sivuston responsiivisuus ja testaus	32
<b>7 YHTEENVETO</b>	<b>34</b>
<b>LÄHTEET</b>	<b>36</b>

## LIITTEET

Liite 1. JavaScript-funktiot.

## KUVAT

Kuva 1. NCSA Mosaic -selain 1990-luvun alussa. (WIRED 2014.)	9
Kuva 2. 3PlayMedia-näkymä.	19
Kuva 3. LearnItIn5 -interaktiivinen video.	20
Kuva 4. TED-Ed-videoluentonäkymä. (TED-Ed 2014)	21
Kuva 5. Video.not-muistiinpanosovellus.	23
Kuva 6. EduCanon-video-opetusnäkyvä.	24
Kuva 7. Vaihtuva sisältö.	29
Kuva 8. Hakusivun ominaisuudet.	31

## TAULUKOT

Taulukko 1. Video-elementin rakenne. (MSDN Microsoft 2014.)	13
Taulukko 2. Video-oppimisympäristöjen vertailu.	25

## SANASTO

neurologia	Neurologia eli hermotautioppi on lääketieteen osa-alue, joka tutkii ja hoitaa keskus- ja ääreishermoston sairauksia, poikkeavuuksia ja vammoja.
Ogg	Avoimen standardin multimediasäiliötiedostomuoto
SGML	Metakieli, jonka avulla määritellään dokumenttien merkintäkieliä. (Standard Generalized Markup Language)
WebM	Googlen kehittämä rajoittamattoman videotiedostomuoto, jonka tavoitteena on tuottaa korkealaatuinen avoin standardi HTML5-videolle.

# 1 JOHDANTO

Insinööriyön tavoitteena oli tuottaa video-oppimisympäristö Turun yliopiston lääketieteellisen tiedekunnan opiskelijoiden käyttöön. Yliopiston tarkoituksena oli tuottaa opiskelijoille oppimisympäristö, jossa on mahdollisuus syventää luennoilla saamia tietoja ja taitoja verkossa ja näin laajentaa opiskelutapoja. Video käsittelee neurologiaa, joka aiheena on hyvin laaja ja siksi tarkoituksena on tehdä ympäristö, jota on mahdollista muokata jälkeenpäin lisäämällä oppimateriaalia.

Maailmalla käytetään varsinkin opetuksessa jo paljon videoita ja palveluita, jotka mahdollistavat saamaan videosta kaikki hyödyt opetukseen. Mahdollisuus lisätä tekstiä, kuvia tai vaikka kysymyksiä videon ympärille saa katsojan syvennymään paremmin videon aiheeseen, ja pohtimaan tarkemmin videon käsittelemää aihetta. Lisäksi mahdollisuus keskustella videon käsittelemästä aiheesta on hyvä tapa oppia aiheesta lisää. Tämän tyyppisiä palveluita on tarjolla paljon ja opinnäytetyössä käydään läpi niistä muutamia. Vertaillaan ja tutkitaan niiden ominaisuuksia, käytettävyyttä, ulkoasua ja toiminnallisuuksia. Lisäksi pohditaan, miten tämä vertailu on vaikuttanut oppimisympäristön suunnitteluun ja toteutukseen.

Työn alkuosassa paneuduttiin www-trendien historiaan ja nykypäivään. Työssä käydään pikaisesti läpi www-kehityksen historiaa ja miten on päästy nykypäivän tasolle. Kerrotaan minkälaisia www-sivustoja oli 1990-luvun alussa, ja mitä sisältöä sinne oli mahdollista syöttää. Listaan HTML5:n tuomat uudet ominaisuudet, ja miten nämä helpottavat huomattavasti www-sivujen luomista.

Videosoitin toimii avainasemassa video-oppimisympäristössä, joten työssä käsitellään videosoitimen rakenne ja toteuttaminen. Mitä erilaisia funktioita ja ominaisuuksia sen toteuttaminen vaatii ja miten siitä saa yhtä käyttäjäystävällisen, kuten muutkin yleisimmät videosoitimet. Käytössä olevalla WordPress-sisällönhallintajärjestelmällä olisi ollut tarjota monia valmiita videosoitin toteu-

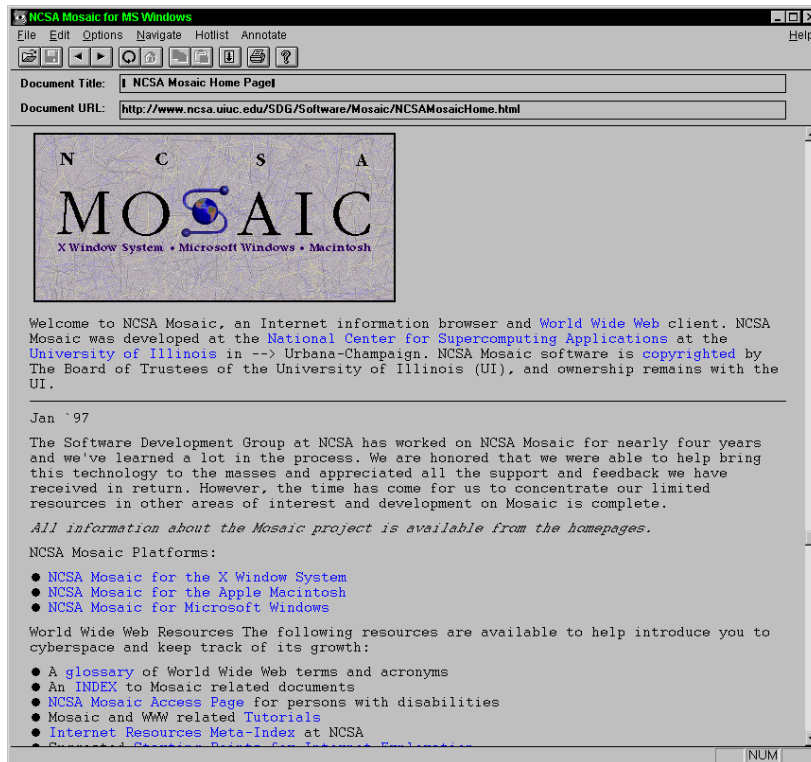
tuksia, jonka takia selostetaan myös miten hyödyttiin toteuttamalla soitin alusta asti itse.

Työn loppupuolella käydään läpi itse oppimisympäristön suunnittelu ja toteutus, jonka tavoitteena oli saada aikaan interaktiivisen sivusto, syventämään opiskelijan jo oppimaansa tietoa aiheesta. Selostetaan, miten HTML5/JavaScriptin tuomia mahdollisuuksia hyödynnettiin oppimisympäristön luomisessa. Lisäksi käydään läpi mitä ohjelmia ja ohjelmointikieliä oppimisympäristön luomisessa käytettiin. Paneudutaan WordPress-sisällönhallintajärjestelmän tuomiin sivuston rakentamista helpottaviin ominaisuuksiin, kuten lisäosiin ja sisällön tuottamiseen koskeviin aiheisiin. Kerrotaan myös työn aikana vastaan tulleista ongelmista ja testauksesta.

## 2 WWW-SIVUSTOJEN TEKEMINEN ENNEN JA NYT

### 2.1 WWW-trendien historiaa

WWW-historia lähtee liikkeelle 1990-luvun alusta, jolloin Tim Berners-Lee kehitti HTML-sisällönmerkkaukielen. Se perustui SGML-merkkaukieliin, joka on yleisesti tunnistettavissa oleva tekstin merkitsemismenetelmä. Vuosi tästä Tim Berners-Lee kehitti ensimmäisen version selaimestaan. Tähän aikaan HTML-kielissä oli paljon rajoitteita, joten kehittäjät pystyivät lisäämään ainoastaan hyvin yksinkertaista tekstiä www-sivuille. Tästä ei mennyt kauankaan, kun Hewlett-Packardin www-kehittäjä Dave Ragget julkaisi HTML+ :n. HTML+ sisälsi myös IMG-tägin, joka mahdollistaa myös kuvien lisäämisen www-sivuille. Kuva 1 esittää 1990-luvun alussa NCSA Mosaic -selaimen näyttämiä sivuja. (Addison Wesley Longman Limited 1998.)



Kuva 1. NCSA Mosaic -selain 1990-luvun alussa. (WIRED 2014.)

Kehitys jatkui nopeaa vauhtia, ja lisää ominaisuuksia ja parempia selaimia ilmestyi usein. HTML 2.0 ilmestyi kesäkuussa 1994, ja sitä seurasi HTML 3.2 tammikuussa 1997. HTML 3.2 sisälsi jo ominaisuuksien puolesta muun muassa taulukot, tekstin lisäämisen kuvien ympärille sekä ylä- ja alaindeksoinnit. Tämän jälkeen julkaistiin nopeasti HTML 4.0, joka sisälsi style- ja class-elementit ulkoasun parempaan muokkaamiseen ja erilaisten luokkien nimeämiseen. Selainten kehitys on aina tullut hieman myöhässä ja eri selaimet näyttävät sisällön jonkin verran eri tavalla. Tulevaisuudessa olisi hyvä, että kehittäjien ei tarvitsisi ottaa huomioon liian montaa ulkoasuun vaikuttavaa seikkaa vaan voivat keskittyä olennaiseen eli itse www-sovellusten kehittämiseen. (Addison Wesley Longman Limited 1998.)

## 2.2 HTML5

WWW-kehitys on muuttunut paljonkin sen alkuajoista ja nykyään on tarjolla valtavasti erilaisia ominaisuuksia ja tapoja tehdä näyttäviä www-sovelluksia. Nykyään on normaalia, että liikutellessa hiirtä kuvaruudulla aina jotain ilmestyy tai muuttuu sivustolla, joten käyttäjä tavallaan kommunikoi sivuston kanssa. Hienoimmissa HTML5-toteutuksissa näkee, miten sivusto reagoi hiiren liikkeeseen, kuten esimerkiksi sivustolla (<http://www.ro.me/>) on interaktiivinen musiikkivideo, joka elää sen mukaan, mihin suuntaan hiirtä kuvaruudulla viedään. Tämän tyyppistä musiikkivideota ei pelkällä HTML5:llä saada aikaan, mutta HTML5:n avulla pystytään katsomaan, missä kohtaa kuvaruutua hiiri on ja näin ohjelmoimaan halutut toiminnot. (ROME 2014.)

HTML5 toi mukanaan paljon uusia www-kehitystä helpottavia ominaisuuksia. HTML5:een liittyvät myös CSS3 ja JavaScript, joiden avulla saadaan tyyliä ja lisättyä toiminnallisuuksia. Näin ollen HTML5:llä yleensä tarkoitetaan näiden kielten kokonaisuutta. HTML5:n monipuolisuus ja eri laitteilla ja sovelluksilla toimivuus tekee siitä tällä hetkellä suosittua kielen esittää www-sisältöä. HTML5:n tärkeimpinä etuina pidetään sitä, että se toimii kaikilla laitteilla, joissa on käytetyimpien selaimien päivitetty versio. HTML5 on avointa lähdekoodia,

joka tarkoittaa että siitä on tietoa tarjolla ja se vaikuttaa kielen suosioon www-kehittäjien parissa. HTML5 ei myöskään tarvitse erillisiä lisäosia toimiakseen, kuten esimerkiksi Flash. (Addison Wesley Longman Limited 1998; F+W Media 2014; W3Schools 2014.)

### 2.3 HTML5:n uudet ominaisuudet

HTML5:n tuomat uudet ominaisuudet helpottavat huomattavasti www-sivujen luomista. Aikaisemmin dokumenttityylin määrittely XHTML tavalla oli esimerkiksi

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

HTML5:n keinoin ei tarvitse kuin kirjoittaa kuin

```
<!DOCTYPE html>
```

(Tuts Plus 2014.)

Yksi ohjelmointia helpottava ominaisuus on se, että ei tarvitse muistaa laittaa attribuutteja lainausmerkkien sisään. Näin säästyy paljon aikaa, kun ei tarvitse etsiä merkkäusvirheitä koodista. Muitakin koodia lyhentäviä ominaisuuksia on tullut, kuten type-attribuutin jääminen pois, kun viitataan script- tai link-tagisiin. HTML5:n avulla pystyy myös helposti lisäämään mihin tahansa elementtiin oman data-attribuutin, jota voi tämän jälkeen käyttää moniin eri tarkoituksiin, kun haluaa lisätä tiettyjä arvoja elementeille. Liukusäätimen luomiseen HTML5 tuo range-arvon, jolla pystyy määrittelemään säätöväliä. Näin ohjelmoija voi määrittää esim. äänenvoimakkuuden arvovälin 0–100 tai 0–10, ja käyttää tämän jälkeen näitä arvoja koodissa itse toiminnallisuutta luodessaan. (W3Schools 2014 HTML <video> Tag.)

Yksi HTML5:n täysin uusista ominaisuuksista on canvas-elementti, joka mahdollistaa grafiikan piirtämisen samanaikaisesti videotoston aikana. Piirtämiseen canvas-elementti tarvitsee ohjelmointia JavaScriptillä ennen kuin näytölle ilmestyy mitään. Toinen täysin uusi ominaisuus on media-elementit video ja audio,

joiden kanssa voi määrittää ja lisätä ääntä ja kuvaa www-sivuille. (W3Schools 2014 HTML <video> Tag.)

HTML5 on tuonut myös suuren määrän uusia rakenteellisia ja semanttisia elementtejä, jolloin aina ei tarvitse käyttää div-tagia, vaan on mahdollista käyttää siihen tarkoitukseen sopivaa tagia. Esimerkkinä luotaessa navigaatiota tai alapalkkia kannattaa käyttää niille olemassa olevia omia tageja nav ja footer. Tällöin rakenne pysyy selkeämpänä ja helpommin muokattavana. (W3Schools 2014 HTML5 Introduction.)

## 3 HTML5:N VIDEOSOITTIMEN HALLINTA JAVASCRIPTILLÄ

### 3.1 Video elementin rakenne

Seuraavassa taulukossa (Taulukko 1) on listattuna kaikki video-elementin attribuutit kuvauksineen.

Taulukko 1. Video-elementin rakenne. (MSDN Microsoft 2014.)

Attribuutti	Kuvaus
<b>src</b>	Merkkijono, joka kuvaa videon URL -osoitetta
<b>controls</b>	Boolean muuttujalla määritettävä HTML5:n valmis videon hallintapaneeli. Sisältää play- ja pausepainikkeen, edistymisosoittimen ja äänenvoimakkuuden säädön.
<b>poster</b>	Merkkijono kuvan URL -osoitteeseen, joka ilmestyy videon tilalle, kun videota ei voida toistaa tai se on vielä latautumassa.
<b>loop</b>	Boolean muuttuja, joka kuvaa toistetaanko videota niin kauan kunnes pausepainiketta on painettu.
<b>muted</b>	Boolean muuttuja, jonka avulla voidaan määrittää, että video ei toista lainkaan ääntä
<b>autoplay</b>	Boolean muuttuja, jonka avulla video alkaa toistua automaattisesti kun sen sisältö on riittävästi latautunut.
<b>preload</b>	Boolean muuttuja, joka kuvaa kuinka paljon videon tarvitsee latautua, että sitä voidaan toistaa.
<b>height</b>	Arvo, jolla määritetään videosoittimen korkeus pikseleinä.
<b>width</b>	Arvo, jolla määritetään videosoittimen leveys pikseleinä.

Video-elementin sisään pystytään määrittämään kaikki taulukossa 1 olevat attribuutit seuraavanlaisella koodilla

```
<video src="demo.mp4" controls autoplay loop muted preload="auto"
height="400px" width="600px" poster="demo.jpg" ></video>
```

## 3.2 Videon hallintafunktiot

### 3.2.1 Videoformaatin valintafunktio

Yksi tärkeimmistä funktioista käsiteltäessä HTML5-videosoitinta on funktio, jossa tarkistetaan, mitä videon muotoa käytössä oleva selain tukee. Näin vältetään ongelmilta ja pystytään opastamaan käyttäjää valitsemaan selain, jonka kanssa video on mahdollista toistaa. Video-elementtiä lisättäessä täytyy olla tarkkana, että jokaiselle selaimelle löytyy oma formaattinsa, jota se pystyy toistamaan. Kolme formaattia, joita yleisimmät selaimet tukevat, ovat MP4, WebM ja Ogg. Koodissa olisi hyvä määrittää ainakin formaateille MP4 ja Ogg omat videoelementtirivinsä, koska näin saadaan kaikissa yleisimmissä selaimissa video näkymään (Esimerkki 1). Chrome, Internet Explorer ja Safari tukevat MP4-formaattia, mutta Firefox ja Opera Ogg-formaattia. Google Chrome -selain on näistä formaatin yhteensopivuuden suhteen paras, sillä se tukee näistä kaikkia kolmea formaattia. Videoformaatin valintaan ei välttämättä tarvita erillistä funktiota, vaan voidaan lisätä suoraan video-tagin sisään useampi scr-attribuutti. Funktion avulla saadaan tehtyä kuitenkin sama ja lisäetuna voidaan antaa käyttäjälle palautetta ja ohjeita, mikäli video ei toimi käytössä olevassa selaimessa. (W3Schools 2014 HTML <video> Tag.)

#### Esimerkki 1. Video-elementti

```
<video>
  <source src="video.mp4" type="video/mp4">
  <source src="video.ogg" type="video/ogg">
</video>
```

### 3.2.2 PlayPause-funktio

PlayPause-funktion avulla kontrolloidaan videon toistumista ja joko pysäytetään tai toistetaan videota ja lisäksi samanaikaisesti saadaan annettua käyttäjälle palautetta toiminnon suorittamisesta. Palaute on yleensä videosoittimen paneelissa olevan painikkeen vaihtuminen ja ruudussa välähtää pause- tai play-ikoni (Esimerkki 2). Tämä funktio on hyvä liittää itse painikkeeseen, kuvaruutuun ja kokotilanäytössä vielä näppäimistön välilyöntipainikkeeseen.

#### Esimerkki 2. PlayPause-funktio

```
function playPause(){
  if(vid.paused){
    vid.play();
    playbtn.style.background = "url(pause.png)";
  } else {
    vid.pause();
    playbtn.style.background = "url(play.png)";
  }
}
```

### 3.2.3 Äänenvoimakkuuden liikusäädinfunktio

HTML5:n avulla saadaan helposti tehtyä soittimelle liikusäädin, jonka avulla voidaan videon äänenvoimakkuutta muuttaa halutulle tasolle (Esimerkki 3). Input-elementissä voidaan määrittää sen minimi- ja maksimiarvot ja value-arvoksi voidaan määrittää videon äänenvoimakkuuden alkuarvo. Step-arvo määrittää kuinka tarkasti äänenvoimakkuutta voidaan säätää. Elementissä olevaa liukuarvoa muuttamalla koodi kutsuu funktiota ja asettaa halutun äänenvoimakkuuden tason.

#### Esimerkki 3. Setvolume-funktio

```
<input id="volumeslider" type="range" min="0" max="100" value="100"
step="1">

function setvolume(){
  vid.volume = volumeslider.value / 100;
}
```

### 3.2.4 Vidmute-funktio

Vidmute-funktion käyttämällä voidaan yhtä painiketta painamalla määrittää videon äänenvoimakkuus tasolle nolla (Esimerkki 4). Samalla funktio laittaa muistiin sen hetkisen äänenvoimakkuuden arvon, jota käytetään kun painiketta painetaan uudestaan. Näin ollen äänenvoimakkuus palautuu aikaisemmin olleelle tasolle, kun halutaan ääni takaisin videoon. Funktio vaihtaa myös painikkeen ikonia kun sitä painetaan, joten käyttäjä näkee onko ääni päällä vai ei.

#### Esimerkki 4. Vidmute-funktio

```
function vidmute(){
  if(vid.muted){
    volumeslider.value = cachedVolume;
    vid.muted = false;
    mutebtn.style.background = "url(mute.png)";
  } else {
    cachedVolume = volumeslider.value;
    vid.muted = true;
    mutebtn.style.background = "url(unmute.png)";
    volumeslider.value = 0;
  }
}
```

### 3.2.5 JumpToTime-funktio

Tämän funktion avulla pystytään määrittämään aloituskohta kuvalle, tekstille tai mille tahansa objektille, josta halutaan videon jatkavan toistamista. Näin ollen mikäli videossa on eri aiheita käsitteleviä kohtia, pystytään helposti hyppimään näiden välillä. Esimerkissä 5 kuvalle on annettu title-arvo, joka on muotoa MM|SS eli kaksi ensimmäistä merkkiä kuvaa minuutteja ja kaksi viimeisintä sekunteja. Kuvaa painettaessa suoritetaan funktio ja määritetään videon sen hetkinen aika (currentTime). Funktiossa olevilla muuttujilla lasketaan (currentTime) sekunteina, jolloin koodi ymmärtää ajankohdan mihin siirrytään.

#### Esimerkki 5. JumpToTime-funktio

```
$(".jumpPic img").click(function () {
```

```

//Määritetään muuttujaan kuvan title -arvon kaksi ensimmäistä
//merkkiä ja kerrotaan se 60, jolloin saadaan minuutit sekunteina.
var imgStartMin = parseInt($(this).attr('title').substring(0,2)*60);

//Määritetään muuttujaan kuvan title -arvon kaksi viimeisintä
//merkkiä, jotka ovat valmiiksi sekunteina.
var imgStartSec = parseInt($(this).attr('title').substring(3,5));

//Lasketaan nämä kaksi muuttujaa yhteen ja saadaan aika sekunteina.
var imgStart = imgStartMin + imgStartSec;

//Määritetään videon ajankohta johon hypätään.
vid.currentTime = imgStart;
});

```

### 3.2.6 Kokonäyttö-funktio

Tällä funktiolla voidaan määrittää videon hallintapaneelissa olevalle ikonille, että video vaihdetaan kokonäyttötilaan. Funktio tarkistaa mitä selainta käytetään, sillä kaikki selaimet eivät käytä samaa selainmoottoria. Esimerkistä 6 käy ilmi miten funktio toimii ja mitä kaikkea se tarkistaa. (MDN 2014.)

#### Esimerkki 6. ToggleFullScreen-funktio

```

function toggleFullScreen() {
    //Tarkistetaan onko selaimella ollenkaan selainmoottoria
    if(vid.requestFullScreen) {
        vid.requestFullScreen();

        //Tarkistetaan käyttääkö Trident -selainmoottoria (IE)
    } else if (vid.msRequestFullscreen) {
        vid.msRequestFullscreen();

        //Tarkitetaan käyttääkö Gecko -selainmoottoria (Firefox)
    } else if(vid.webkitRequestFullScreen){
        vid.webkitRequestFullScreen();

        //Tarkitetaan käyttääkö WebKit -selainmoottoria (Safari ja Chrome)
    } else if(vid.mozRequestFullScreen) {
        vid.mozRequestFullScreen();
    }
}

```

## 4 VASTAAVANLAISET OPPIMISYMPÄRISTÖT JA NIIDEN OMINAISUUDET

Nykypäivänä opiskelu ei ole pelkästään luennoilla istumista ja opettajan kuuntelemista. Opetuksessa pyritään käyttämään hyväksi kaikkia teknologian tuomia mahdollisuuksia. Puhutaan käänteisestä opetuksesta, joka tarkoittaa perinteisestä opetustavasta poikkeavaa opiskelua, jossa opettaja opetti jonkin asian ja tämän jälkeen opiskelijat tekivät tehtäviä ja opiskelivat kotona lisää aiheesta. Käänteisessä opetuksessa opiskelijat saavat etukäteistehtäväkseen tutustua yleensä internetissä olevaan videoon katsomalla ja kommentoimalla sitä. Videon käsittelemästä aiheesta jatketaan yhdessä oppitunnilla, koska opiskelijoilla on taustatietoa jo aiheesta, opettajan ja opiskelijoiden väliselle vuorovaikutukselle jää enemmän aikaa. Käänteisessä opetuksessa korostuu kolme asiaa, jotka ovat 1) nykyteknologian ja www-materiaalin käyttö, 2) opiskelijan oma aktiivisuus opiskeluun, 3) opettajan ja opiskelijoiden vuorovaikutus. (Knewton 2014.)

Vastaavanlaisia oppimisympäristöjä ja palveluita on tarjolla paljonkin, mutta suurin osa näistä on maksullisia tai ei ollenkaan kaupallisessa levityksessä. Vastaavanlaisista oppimisympäristöistä puhuttaessa tarkoitan oppimisympäristöjä, joissa on videon mukana vaihtuvaa sisältöä ja joka toimii interaktiivisesti katsojan kanssa videon aikana. Video voi antaa lisätietoa tekstin tai tehtävien muodossa tai se ylipäätensä auttaa opiskelijaa oppimaan paremmin ja se olisi jopa miellyttävää. Vaihtuvia tekstejä pystyisi videon päälle luomaan suoraan itse videotiedostoonkin esim. Adoben ohjelmilla Premiere ja After Effects. Tällä tavalla tekstejä ei pysty tosin jälkeinpäin helposti muokkaamaan ja ne saattavat häiritä itse videon katsomista olemalla videokuvan päällä.

3PlayMedian tarjoamista ominaisuuksista tärkein on tekstitys, joka tarjoaa käyttäjälle mahdollisuuden tekstittää video. Tilanteessa, jossa videon ääni on huonolaatuinen, käyttäjä pystyy kuitenkin lukemaan puhutun tekstin. Ohjelma käy läpi videon ja tuottaa tekstin automaattisesti sen pohjalta mitä videossa puhu-

taan tai vaihtoehtoisesti sen pystyy itsekin syöttämään. Tekstin pystyy myös kääntämään automaattisesti muille kielille ohjelman avulla ja muokkaamaan, kuten mitä tahansa tekstiä ennen sen liittämistä videoon, kuten kuvasta 2 huomaat. Ohjelman avulla pystyy myös hakemaan tiettyjä kohtia tekstistä ja palaamaan siihen kohtaan videota. Käyttäjän halutessa videon ja tekstit pystyy helposti hakukoneoptimoimaan, jolloin ne ovat helpommin löydettävissä. (3PlayMedia 2014.)

The screenshot displays the 3PlayMedia 'Upload For Processing' interface. On the left is a navigation menu with categories like 'HOW IT WORKS', 'Overview', 'Video Tutorials', 'FAQs', 'Webinars', 'White Papers', 'How-To Guides', 'Process', 'Security', and 'For Developers'. The main content area is titled 'VIDEO TUTORIALS' and shows a video player with a progress bar at 00:27 / 02:14. The video content shows a 'Upload For Processing' form with the following steps:

1. Choose Folder → Select Service → Confirm Upload
2. Which service would you like?
  - Transcription and Captioning (Default)
  - Alignment Only: I will upload my transcripts (\$1.25 per min)
3. Select Turnaround Time:
 

TURNAROUND LEVEL	DELIVERY DATE (ET)	ADDITIONAL COST PER MIN
Standard	8:00 PM, 12/8/12	
<b>Fast</b>	<b>8:00 PM, 12/11/12</b>	<b>\$0.75</b>
Push	8:00 PM, 12/5/12	\$1.00

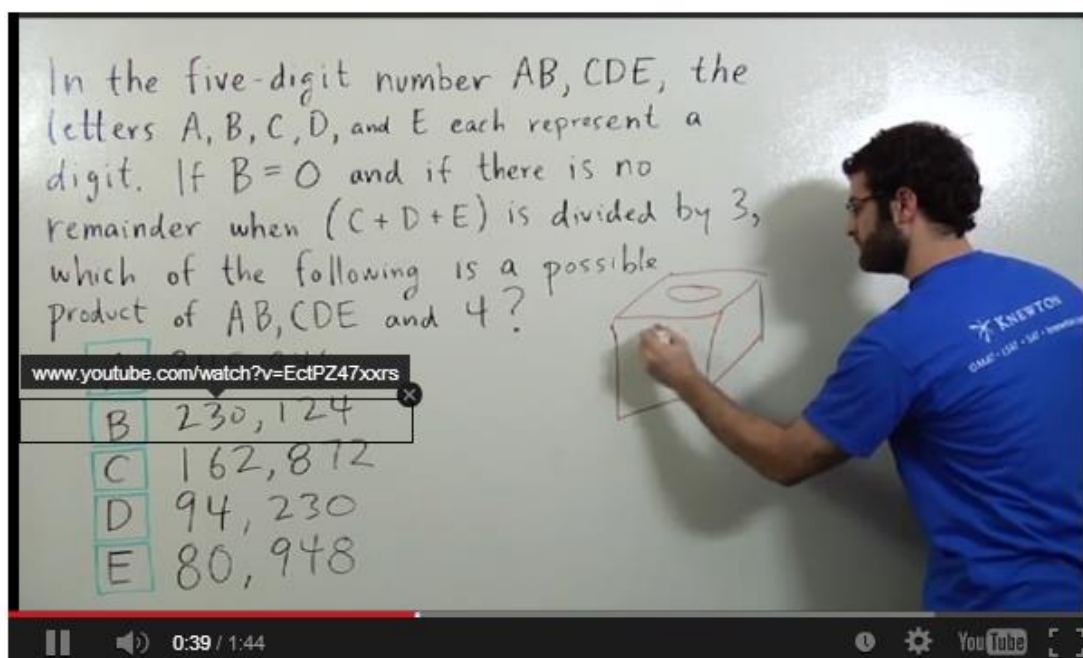
Below the video player is a search bar labeled 'Search Video' and a text block explaining the process: 'Next, specify the turnaround time and transcription settings. In particular, make sure you select the appropriate speaker identification settings. As soon as your files are uploaded, the transcription and captioning process will begin. If you already have transcripts, choose Alignment Only. This is a faster and less expensive way to create captions and interactive video plug-ins. From the My Files page, you can monitor the job status, organize your files, and download a multitude of captions and transcript formats. You also have the option to create translations and multilingual subtitles from English into many...'

Kuva 2. 3PlayMedia-näkymä.

On olemassa myös paljon muitakin palveluja ja tapoja käyttää videota opetus-tarkoituksessa, kuten sivusto SchoolMedia. Tämä sivusto tarjoaa videoita monista eri aihe-alueista ja videon katsottua voi tehdä tehtäviä aiheesta. Tämä sivusto ei tarjoa toisaalta sitä ominaisuutta, että videon aikana pystyisi tekemään tai näkemään kuin itse videon, vaan pystyy ainoastaan tekemään yhtä asiaa kerralla. (SchoolMedia Interactive 2014.)

LearnItIn5 tarjoaa todella interaktiivisen ja hauskan tavan opiskella videon avulla. Tämä palvelu tarjoaa mahdollisuuden tehdä itse videosta tehtävän, jossa käyttäjä pystyy videota painamalla vastaamaan esitettyihin kysymyksiin ja tällä

tavalla etenemään videosta toiseen. Tällaisen videokokonaisuuden tekeminen toisaalta vaatii paljon aikaa ja suunnitelmallisuutta, sillä kaikki mahdolliset vastausvaihtoehdot täytyy ottaa huomioon. Kuvasta 3 huomaa kuinka LearnItIn5 kanssa toimiminen tapahtuu. (LearnItIn5 2012.)

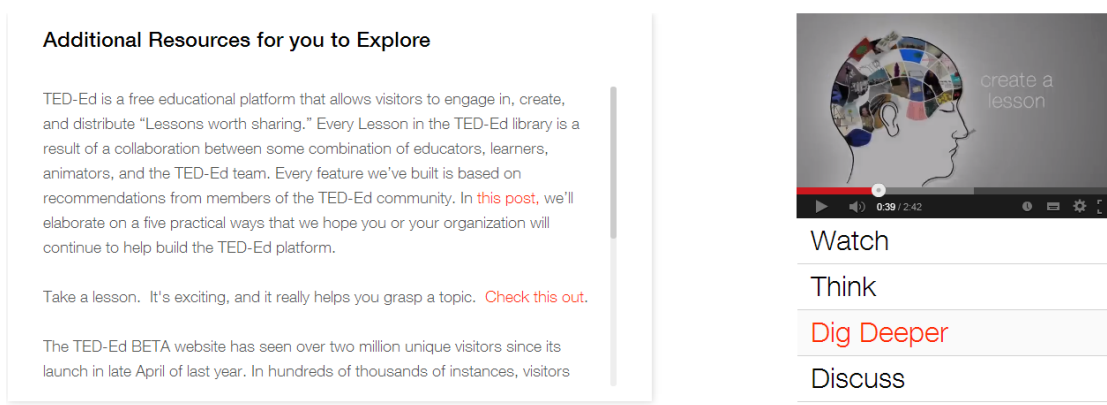


Kuva 3. LearnItIn5 -interaktiivinen video.

Zaption on maksullinen palvelu, joka on erikoistunut kuvatekstien liittämiseen videoihin. Zaptionin avulla voi videoihin liittää kyselyitä, joihin videon katsoja voi videon aikana vastata. Videoihin voi liittää myös muutakin sisältöä, kuten kuvia, tekstiä tai keskusteluja. Zaptioniin voi liittää videoita Youtubesta, Vimeosta tai omista tiedostoistaan. Zaptionin avulla videoiden lisääjä pystyy tarkastelemaan esimerkiksi, mitä videon katsojat ovat vastanneet kyselyissä tai kuka kirjautuneista katsojista videoita on katsonut. Tämä toimii mainiona opetusvälineenä opettajille, jotka haluavat jakaa ja tehdä video-opetusmateriaaleja ja seurata, että kaikki opiskelijat nämä videot katsoneet ja vastanneet kysymyksiin videon pohjalta. (Zaption 2014.)

TED-Ed on ominaisuuksiltaan hyvin samanlainen palvelu kuin Zaption, mutta se on tehty näyttämään vielä hienommalta ja helpommalta käyttää. TED-Ed:n videon katselupohja on monipuolisempi ja siinä pystyy siirtymään eri näkymien

välillä. Mikäli käyttäjä haluaa katsoa vain videota, sen saa näkymään suurena näytöllä, mutta jos haluaa tehdä videon aikana samalla siihen pohjautuvaa kyselyä, voi valita kyselynäkymän. Kyselynäkymässä video siirtyy oikeaan reunaan pienemmäksi ja itse kysely näkyy suurempana vasemmassa reunassa. Näin käyttäjä näkee paremmin pienemmälläkin näytöllä kyselyssä olevat tekstin ja kysymykset. Kolmanteen näkymään, joka näkyy kuvassa 4, voi lisätä katsojalle lisäpohdittavaa aiheesta ja lisätä mitä tahansa sisältöä, kuten tekstiä, kuvia tai vaikka linkkejä toisiin videoihin tai sivustoihin. Viimeisessä näkymässä on keskustelualue, johon voi lisätä videon aiheeseen liittyen keskusteluita ja herättää katsojissa ajatuksia. TED-Ed:n avulla muun muassa opettaja pystyy luomaan opiskelijoilleen videoluentoja ja seuraamaan heidän toimintaansa videoiden parissa. Videoita voi lisätä samoista lähteistä kuten Zaptionissa ja jakaa oppilaiden kanssa suoralla linkillä opettajan luomiin videoluentoihin. (TED-Ed 2014.)



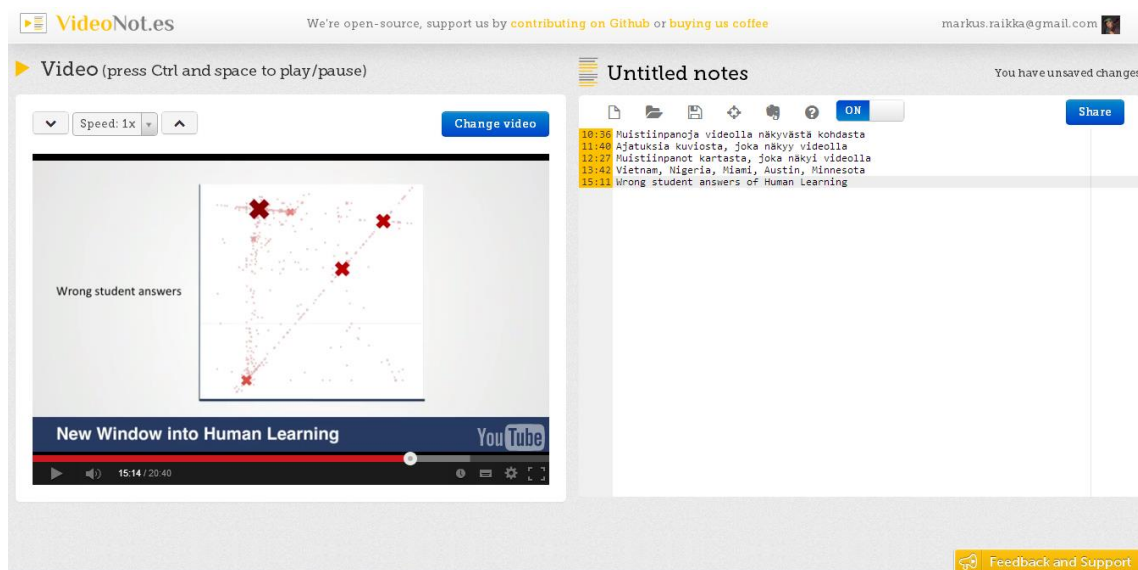
Kuva 4. TED-Ed-videoluentonäkymä. (TED-Ed 2014)

Vialogues on ilmainen www-palvelu, joka on tarkoitettu luomaan videon ympärille keskustelua. Palvelun käyttäjä pystyy lisäämään tekstiä haluamaansa kohtaan videota ja katsojan on tarkoitus reagoida ja vastata tähän tekstiin videota katsellessa. Tämän avulla olisi tarkoitus luoda keskustelua videon ympärille ja syventää videon käsittelemää tietoa aiheesta. Palvelun käyttäjä pystyy kutsu- maan katsojia videolleen ja keskustelemaan videon aikana. Videon aikana käy-

ty keskustelu voidaan jälkeinpäin jakaa helposti muun muassa blogeissa tai nettisivuilla. (EdLab 2014.)

Metta.io-palvelu tarjoaa mahdollisuuden tehdä video-oppimisympäristön, johon voi lisätä muun muassa lyhyitä videoita, kuvia tai tekstiä. Näistä sisällöistä luodaan suurempi kokonaisuus, jota katsojan olisi tarkoitus seurata luennon tapaan. Palvelu on tavallaan ohjelma, jonka avulla käyttäjä pystyy luomaan dia-sarjan, jossa aina yhdellä dialla on joku video ja tämän aikana vaihtuvaa kuva- ja tekstisisältöä. Jokaisella sivulla ei ole pakko olla kaikkia mainittuja sisältö mahdollisuuksia vaan käyttäjä saa itse muokata sivuista haluamansa näköisiä. Videosivujen väliin on myös mahdollista lisätä kysymyssivuja, joihin katsoja voi vastata. Kyselyistä saadut vastaukset tallentuvat palveluun ja ympäristön haltija pystyy näitä tarkastelemaan jälkeinpäin. Maksuttomassa versiossa palvelu kerää ainoastaan 10 ensimmäistä vastausta, mutta siirryttäessä maksulliseen versioon rajoituksia ei ole. Maksullinen versio ei tuo mukanaan lisäominaisuuksia sivukokonaisuuksien tekemiseen, mutta siinä on mahdollista tehdä useampia kuin yksi katsojaryhmä ja riippuen maksullisesta versiosta, on käytössä joko 1 Gt:n tai 10 Gt:n tallennustila. (Metta 2014.)

VideoNot on Google Driveen lisättävä maksuton ohjelma. Tämän avulla käyttäjä pystyy lisäämään videoihin muistiinpanoja niin, että ne korostuvat tietyssä kohdassa videota. Palvelu on hyödyllinen kaikille opiskelijoille, mutta erityisesti auttamaan opiskelijoita, joiden äidinkieli ei ole sama kuin videossa puhuttava kieli tai joilla on vaikeuksia oppia pelkästään kuuntelemalla ja katsomalla. Näin ollen opiskelija pystyy lukemaan jo lisäämäänsä tekstiä tai lisäämään uusia muistiinpanoja samalla katsoen videota. Palvelun avulla opiskelijan oppimistavat monipuolistuvat ja tieto jää paremmin muistiin. Käyttäjä pystyy hyppimään videon kohdasta toiseen ja päättämään helposti kohdan videosta, johon tekstiä lisää. VideoNot toimii ikään kuin aikataulutettuna muistiona, jonka avulla opiskelija voisi miellyttävällä tavalla esimerkiksi kerrata koealuetta ennen koetta, kuten alla olevasta kuvasta huomaa. (VideoNot 2013.)



Kuva 5. Video.not-muistiinpanosovellus.

EduCanonin on täysin maksuton palvelu opettajien käyttöön, jolla saa kaikki hyödyt irti videoista. Tämän avulla opettaja pystyy helposti lisäämään sisältöä, kuten monivalintakysymyksiä niin, että ne ilmestyvät uutena palkkina videon vasempaan laitaan ja samalla video pysähtyy. Opiskelijan vastattua kysymyksen ohjelma antaa välittömästi palautteen, onko vastaus oikein vai väärin, minkä jälkeen palkki poistuu näkyvistä ja video jatkuu. Lisäksi opettaja pystyy lisäämään kysymyksiä (Kuva 6), joihin opiskelijoiden olisi tarkoitus vastata sanallisesti. Palvelu on tarkoitettu varsinkin käänteiseen opetukseen, jossa opiskelijat perehtyvät seuraavalla luennolla käsiteltävään aiheeseen ennen varsinaista luentoa videon avulla. Tämän palvelun avulla opettaja pystyy lisäämään sisältöä videoihin monipuolistaakseen opetusta. (EduCanon 2014.)

The image shows a screenshot of the EduCanon interface. On the left, a white panel contains the text: "You can even ask free response questions to see the thought processes of your students." Below this text is an orange button with a keyboard icon and the text "Free response". Underneath the button is a large empty text input box. At the bottom of this panel is an orange "Submit" button. On the right, a dark-themed interface shows a search bar with a question mark icon and the text "anon". Below the search bar are two checked items: "What is Flipped Learning?" and "TWBAT understand why I flipped my classroom". A dropdown menu shows "6th-8th", "Math", and "Pre-algebra". Below this is a search bar containing the URL "http://www.youtube.com/watch?v=9aGul...". A grid of six video thumbnails is displayed, each with a title and view count. The thumbnails include: "Teaching for Tomorrow: Flipped Learning" (9000 Magazine, 42374 views), "How the Flipped Classroom works" (MediaOnLearning, 3605 views), "Why I Flipped My Classroom" (bookstodgy, 14792 views), "Flipped Class Video Tour" (LearningMastery, 4923 views), "The Flipped Classroom", and "Flipped Class 101".

Kuva 6. EduCanon-video-opetusnäkyä.

#### 4.1 Yhteenveto oppimisympäristöistä

Video-oppimisympäristöt ovat keskenään ominaisuuksiltaan hyvinkin samanlaisia, mutta jotkut sopivat paremmin tiettyjen asioiden tekemiseen toiset. Toiset ympäristöt mahdollistavat kysymyksiin vastaamisen pysäyttämättä videota, mutta muun muassa Metta.io-ympäristössä kysymykset lisätään videon tiettyihin kohtiin, jolloin kysymys ilmestyy videon tilalle. Maksullisissa versioissa ei välttämättä ole lisää ominaisuuksia ja mahdollisuuksia videon muokkauksen suhteen, vaan yleensä ne tuovat suuremman tallennuskapasiteetin ja laajemman käyttäjähallinnan. Laajempi käyttäjähallinta tuo ainakin opettajille mahdollisuuden jakaa eri opetusryhmien kesken vain tiettyjä videoluentoja. Suurempi tallennuskapasiteetti auttaa ainakin silloin, jos palvelu ei tue muihin lähteisiin, kuten Youtubeen tai Vimeoon tallennettuja videoita. Seuraavassa taulukossa on yhteenveto käydyistä video-oppimisympäristöistä ja niiden ominaisuuksista.

Taulukko 2. Video-oppimisympäristöjen vertailu.

	1	2	3	4	5	6	7	8	9
<b>Maksullinen</b>	X	X		X	X		X		
<b>Automaattinen tekstitys</b>	X								
<b>Vastausten tallentuminen</b>		X		X		X	X		
<b>Linkkien lisääminen videoon</b>			X						
<b>Vaihtuva sisältö</b>									
<b>Teksti</b>	X	X		X	X	X	X	X	X
<b>Kuvat</b>	X	X		X	X	X	X		X
<b>Kyselyt</b>		X		X	X	X	X		X
<b>Keskustelu</b>					X	X			

- 1 3PlayMedia
- 2 SchoolMedia
- 3 LearnItIn5
- 4 Zaption  
TED-
- 5 Ed
- 6 Vialogues
- 7 Metta.io
- 8 VideoNot
- 9 EduCanon

## 5 VIDEO-OPPIMISYMPÄRISTÖN SUUNNITTELU

Projektin suunnittelun suhteen lähdettiin liikenteeseen kirjaamalla asiakkaan vaatimat tarpeet ja toiveet. Asiakkaan toiveena oli toteuttaa opiskelun ohelle oppimisympäristö, jossa Turun yliopiston lääketieteellisen tiedekunnan opiskelijat voisivat kehittää tietojaan luennoilla jo käydyissä aihealueissa. Sivustolle tulisi yksi pidempi video, joka käsittelee koko potilaan statuksen määrittelyn, ja oppilas voisi lukemalla lisämateriaalia myös syventämään tämän määrittelyn taustalla olevia tietoja. Videossa pitäisi pystyä hyppimään kohdasta toiseen, sen suhteen mitä aihetta oppilas haluaisi oppia eikä joutuisi välttämättä koko videota katsomaan, vaan ainoastaan hänelle olennaista aluetta. Sivuston oikeaan laitaan tulisi lisätietoa, joka vaihtuisi videon kohdasta riippuen ja tätä tekstiä pystyisi jälkeinpäin helposti muokkaamaan ja näkymisajankohtaa vaihtamaan. Sivuston vasempaan laitaan tulisi syventäviä aihealueita, joita painamalla opiskelijalle aukenisi ponnahdusikkuna, jossa olisi tekstiä, kuvia ja mahdollisesti linkkejä muihin aihetta koskeviin sivustoihin. Näiden ikkunoiden sisältöä tulisi myös pystyä muokkaamaan jälkeinpäin. Lisäksi sivustolle tulisi hakusivu, jonka avulla käyttäjä pystyy helposti tarkastelemaan, mitä kaikkea sisältöä sivustolla on ja hakemaan juuri hänelle tarpeellista tietoa.

### 5.1 Oman ympäristön ominaisuudet

Toimeksiantajalta saatiin paljonkin ideoita oppimisympäristön suhteen. Tärkeimmät ominaisuudet kuitenkin liittyvät itse videon katseluun ja oppimateriaalin esittämiseen. Sivuston tärkein elementti on video, jossa esitetään neurologian lääkärin tekemä status potilaasta. Tämä status tarkoittaa henkilöstä tehtävää toimenpiteiden sarjaa, siitä mikä potilaalla on vialla.

Oppimistarkoituksessa tärkeitä ominaisuuksia on lisämateriaalin esittäminen ja olemassa olevien tietojen syventäminen. Videon ympärillä esitettävä lisämateriaali videon kohdasta riippuen on sivuston tärkein ominaisuus. Tällä tavoin opiskelija pystyy lukemaan lisämateriaalia halutessaan aiheesta. Lisämateriaalin

sisällöksi voidaan lisätä muun muassa linkkejä muihin videoihin sivustolla tai vaikka Youtubessa oleviin videoihin. Toinen tärkeä ominaisuus on mahdollisuus siirtyä videossa opiskelijalle ajankohtaiseen kohtaan vaivattomasti. Ylläpitäjien tulee pystyä itse vaikuttamaan jälkeempään siihen, mitä tekstiä videon ympärillä esitetään ja mihin aikaan, koskematta sivuston taustalla olevaan koodiin. Tämä ominaisuus vaati eniten suunnittelua, sillä aikaisempaa kokemusta tästä ei ollut. Ylläpitäjän tulee pystyä lisäämään ainakin tekstiä, kuvia ja linkkejä ja muokkaamaan näitä halutunlaiseen muotoon. Halutessaan ylläpitäjien tulee myös pystyä vaihtamaan sivustolla oleva video ja vaihtamaan tällä olevien siirtymiskuvakkeiden kuvia ja siirtymiskohtia videossa.

Muihin olemassa oleviin ympäristöihin verrattaessa toteutetun videooppimisympäristön ominaisuudet ovat hyvinkin samanlaisia kuin muissakin ympäristöissä. Kuvatekstien, pikkukuvien ajankohdan ja sisällön vaihtaminen onnistuu yhtä helposti kuin olemassa olevissa ympäristöissä. Kuvateksteihin voidaan laittaa mitä tahansa sisältöä, joka on kaikissa muissakin ympäristöissä ollut avainasemassa. Muutamissa ympäristöissä on mahdollisuutena tarkastella katsojien tekemiä vastauksia ja nähdä, ketkä ovat videon katsoneet. Näitä ominaisuuksia ympäristössäni ei ole, koska näitä ei nähty tarpeellisiksi ja sivuston käyttäjät eivät ole opettajia vaan opiskelijoita.

VideoNot-muistiinpano-ominaisuus sopisi sivustolle, mutta toteutuksen kannalta se jäisi kuitenkin vajavaiseksi. Se vaatisi jokaiselle käyttäjälle oman tallennustilan, jonne tallentaa muistiinpanonsa, jotta niitä pystyy myöhemmin katsomaan ja muokkaamaan. Tämä ominaisuus päädyttiin jättämään kokonaan pois.

Videoiden liittäminen olemassa olevista videonjakamispalveluista, kuten YouTubeista ja Vimeosta, sivustolle onnistuu. Toisaalta pelkkä suora linkki tekisi saman asian, sillä näihin videoihin ei ole tarkoitusta lisätä edellä mainittuja ominaisuuksia, vaan videota voisi vain katsella. Näin ollen muut videot toimisivat pelkkänä lisämateriaalina, toisin kuin sivustolla oleva pidempi video, jonka pohjalle koko sivusto perustuu.

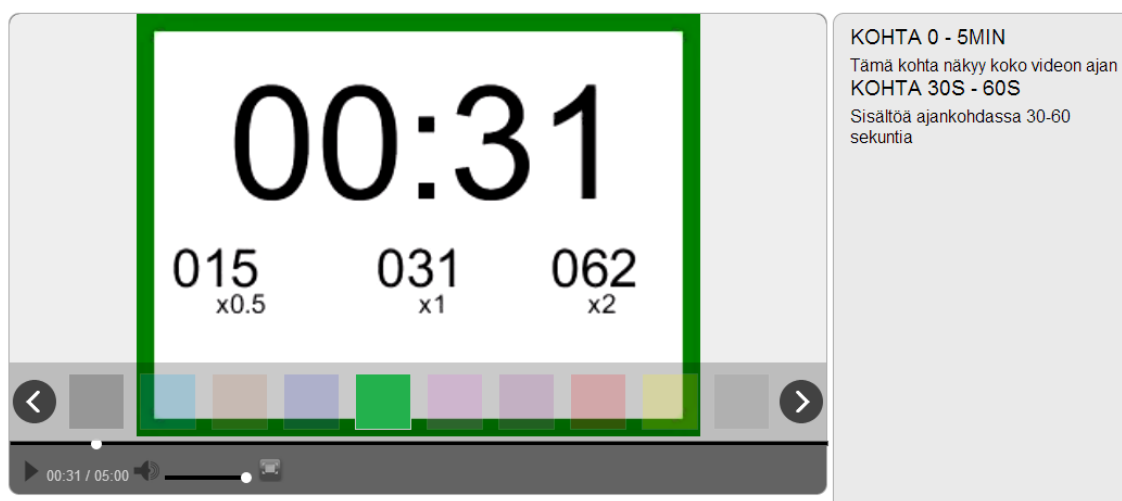
## 5.2 Videosoittimen suunnittelu

Videosoitinta suunniteltaessa kirjattiin, mitä soittimen välttämättömiä ominaisuuksia tarvittaisiin ja mitä ominaisuuksia soittimessa ei välttämättä tarvitsisi olla. Välttämättömät soittimen ominaisuudet ovat playpause-painike, videon pituuden ja kohdan näyttäminen sekunteina, muteunmute-painike, äänenvoimakkuuden säätö ja kokonäyttöpainike. Lisäsin myös edistymisosoittimen, jota painamalla pääsee haluamaansa kohtaan videossa. Soittimen palasia yhteen liittäessä ja taustalla olevaa koodia kirjoitellessa etsittiin paljon tietoa vastaavanlaisista toteutuksista. Näitä vastaavanlaisia toteutuksia tutkimalla ja aihetta opiskelemalla saatiin koottua ominaisuuksiltaan halutun näköinen soitin. jQuery JavaScript -kirjaston avulla soittimesta saa hyvinkin omaperäisen ja hienon, mutta suurin osa näistä ominaisuuksista on tehty toimimaan hiirelle ja sivustoa tulisi pystyä myös käyttämään kosketusnäytöllä, joten soitin pyrittiin saamaan toimimaan molemmissa ympäristöissä.

## 6 VIDEO-OPPIMISYMPÄRISTÖN TOTEUTUS

### 6.1 Vaihtuvien tekstien lisääminen

Vaihtuvien tekstien sisältö tulee WordPressin videosivun alasivuilta ja näille on määritelty aikaväli, jolla sisältö näkyy videosivun oikeanpuoleisessa palkissa. Koodissa haetaan kaikki alasivut ja määritetään data-start ja data-end arvot näytettävälle sisällölle. Data-arvot määritetään WordPressin kautta More Fields -lisäosan avulla, jolla pystyy lisäämään sivuille ja artikkeleille lisämuuttujakenttiä. Nämä kentät haetaan ja syötetään koodissa elementille, joka halutaan näyttää. Tämän jälkeen koodi syöttää kaikkien alasivujen aloitus- ja lopetusajat taulukkoon, josta se tarkistaa sekunnin välein sopiiko aloitusaika videon sen hetki- sen ajan kanssa yhteen, kuten kuvasta 7 huomaa. Arvojen sopiessa yhteen sivun sisältö näytetään tai poistetaan näkyvistä. Lisäksi vaihtuvan sisällön otsikon avulla voi halutessaan palata videossa siihen kohtaan, jolloin tiettyä aihetta alettiin käsitellä.



Kuva 7. Vaihtuva sisältö.

Jatkokäyttäjän ei siis tarvitse koskea koodiin lainkaan, sisältöä tai sen näyttämisaikakohtaa vaihdettaessa, vaan kaiken saa muutettua WordPressin hallintapaneelin sivut-osioissa. Data-arvoja lisättäessä käyttäjä ei myöskään pysty li-

säämään arvoja väärässä muodossa, sillä More Fields -lisäosan avulla on määriteltävä, että arvot täytyy olla muodossa MM:SS.

## 6.2 Thumbnail-kuvien lisääminen ja niiden ajankohdan määrittäminen

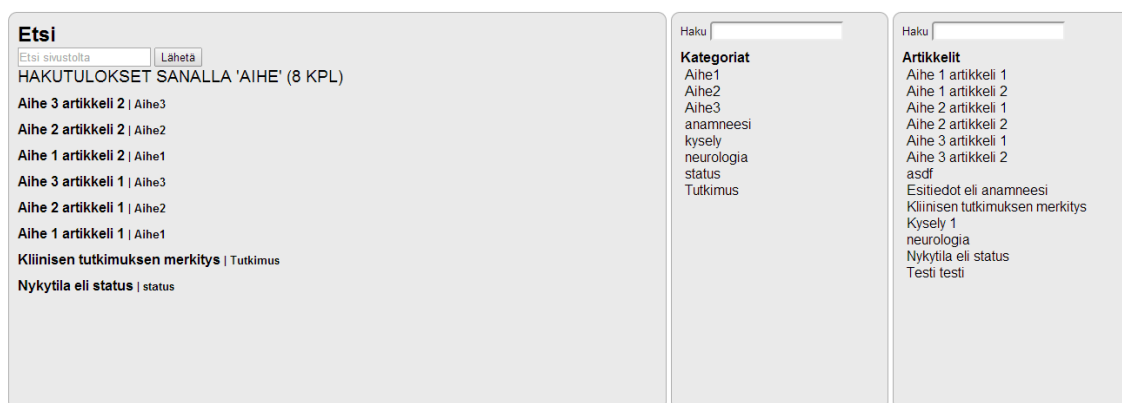
Videon alareunassa olevien thumbnail-kuvien lisääminen on toteutettu Thumbnail Slider -lisäosan avulla, jota on muokattu paremmin käyttötarkoituksiin sopivaksi. Liukupalkkiin voi lisätä niin monta pikkukuvaa kuin on tarpeen eikä kuvien koolla ole suurta merkitystä, vaan lisäosa muokkaa kuvat juuri oikean kokoisiksi. Pikkukuvia näkyy kerralla liukupalkissa kymmenen, mutta palkkia pystyy vierittämään nuolilla oikealle ja vasemmalle, jotta voi liikkua pikkukuvien välillä. Riippuen siitä, missä kohtaa ajallisesti videossa ollaan, sitä käsittelevän kohdan pikkukuva on korostunut ja muut ovat lähes läpinäkyviä. Kuville on määriteltävä aikaväli, jolloin ne ovat korostuneita, syöttämällä kuvien otsikoksi haluttu näkymisaikaväli. Tämä otsikko täytyy olla muotoa MM:SS|MM:SS, jossa ensimmäinen aika on se, jolloin kuva korostuu, ja toinen tarkoittaa aikaa, jolloin kuva palaa takaisin lähes läpinäkyväksi. Koodissa tämän otsikon lukeminen on toteutettu käyttäen JavaScriptin substring-metodia, jonka avulla pystytään lukemaan ainoastaan halutut merkit merkkijonosta.

## 6.3 Videosivun artikkelihaku

Videosivulla on videon vasemmassa reunassa artikkelihaku, jonka avulla opiskelija voi hakea samalla videota katsoessa lisätietoa aiheesta. Hakupalkissa on listattuna kaikki sivustolla olevat artikkelit ja kirjoittaessa hakupalkkiin koodi poistaa ne artikkelit, joiden nimestä ei kirjoitettuja kirjaimia löydy. Painaessa artikkelin nimeä artikkeli avautuu ponnahdus-ikkunaan ja samalla video menee automaattisesti taukotilaan ja sulkiessa ikkunan video jatkaa toistoa. Tämän ominaisuuden ansiosta opiskelijan ei tarvitse hypätä sivulta toiselle, halutesaan nopeasti vilkaista jotain tiettyä artikkelia kesken videon.

## 6.4 Hakusivun ominaisuudet

Hakusivulla on kolme hakumahdollisuutta. Siellä pystyy hakemaan artikkeleita kategorioittain, ja ne on listattu keskimmaiseen palkkiin. Kirjoitettaessa kategoriahakukenttään koodi alkaa poistaa heti niitä kategorioita, joissa ei kirjoitettua kirjainta tai kirjainyhdistelmää löydy. Samalla oikeanpuoleisimmasta listasta, johon on listattu kaikkien kategorioiden alla olevat artikkelit aakkosjärjestyksessä, poistuvat ne artikkelit, joiden kategoriata ei enää kategoriapalkissa ole listattuna. Tämä hakuominaisuus toimii molempiin suuntiin niin, että artikkelin nimen mukaan hakiessa kategorialistasta poistuu myös ne kategoriat, joita ei enää artikkelilistassa ole. Artikkelin sisältöä pystyy lukemaan suoraan hakusivulta painamalla artikkelin nimeä, jolloin avautuu ponnahtus-ikkuna, johon sisältö tulee näkyviin. Kolmas hakuominaisuus mahdollistaa sanallisen haun artikkelien sisällön mukaan. Esimerkiksi neuro-sanalla, tämä hakee kaikki artikkelit ja sivut, joiden kategoriasta, otsikosta tai sisällöstä tämä sana löytyy. Haku näyttää tulokset myös samalla sivulla ja sille on varattu suurin alue, kuten kuvasta 8 näkee.



Kuva 8. Hakusivun ominaisuudet.

## 6.5 WordPressin käyttö sisällönhallintajärjestelmänä

WordPress toimii sisällönhallintajärjestelmänä valtavan hyvin tämän tyyppisen sivuston toteutukseen. Sen lisäosat tuovat suurta apua siihen, että jatkokäyttäjät

pystyy pelkästään WordPressin hallintapaneelia käyttämällä muokkaamaan sivuston sisältöä. Suurikokoisen videon lisääminen omalta koneelta tuo ainoan hankaluuden loppukäyttäjälle. Video täytyy siirtää FTP-tiedonsiirto-ohjelmalla palvelimelle ja tämän jälkeen WordPressin Add From Server -lisäosalla vasta sivuston käyttöön. Tämän lisäosan avulla pystyy minkä kokoisia tiedostoja tahansa tuomaan WordPressiin, kun tavallisesti maksimitiedostokoko on 2 Mt.

Soitinta luotaessa päädyttiin toteuttamaan se alusta asti itse, sillä mikäli olisi käytetty jotain valmista lisäosaa, vaihtuvien pikkukuvien ja kuvatekstien kanssa olisi saattanut tulla suuria ongelmia. Näin päästiin lopulta helpommalla, säästettiin aikaa ja saatiin juuri sen näköinen soitin kuin oli tarkoituskin.

## 6.6 Sivuston responsiivisuus ja testaus

Responsiivisuus eli erikokoisille päätelaitteille mukautuvuus on nykypäivänä tärkeitä asioita www-koodauksessa. (Karukka, M. & Inkilä, T. 2013.) Toimeksiantajan mukaan sivuston ei tarvitse mukautua kännykän näytölle, mutta jo tabletin näytössä sivustoa tulisi jo hyvin pystyä käyttämään. Responsiivisuus toteutettiin CSS ja JavaScriptin keinoin, joiden avulla pystyy helposti toteuttamaan kaikenkokoisille näytöille skaalautuvuuden. Prosentuaaliset leveysarvot mahdollistavat sen, että minkä tahansa resoluution omaavalla laitteella sivusto näyttää samalta. Näin ollen mentäessä pienempiin näyttöihin sisältökin vaihtuu pienemmäksi. Yleensä tehdään myös rakenteellisia muutoksia, kun suunnitellaan sivuston skaalautuvuutta myös matkapuhelimille. Toteutettua sivustoa ei ole tarkoitus käyttää matkapuhelimella, joten rakenteellisia muutoksia pieniin näyttöihin ei tarvinnut huomioida. Lisäksi määriteltiin joitain korkeusarvoja JavaScriptin avulla, niin että koodi tarkistaa näytön korkeuden suhteen, sen kuinka korkeita tietyt alueet pitäisivät olla. Tämän avulla sivuston sisällön saa aina sopimaan näytölle oikean kokoiseksi. JavaScriptin avulla pystytään määrittämään muuttujan kokoruudun korkeus ja tästä vähentämällä ylä-, ala- ja navigaatiopalkit saadaan esimerkiksi määritettyä korkeus, johon sisällön tulisi mah-  
tua.

Sivuston testaus on tärkeimpiä asioita hyvän lopputuloksen kannalta ja sitä olisi hyvä toteuttaa myös ulkopuolisilla tahoilla. Ulkopuolista testausta ei toteutettu, mutta toimeksiantaja testasi sivustoa itse aina tietyin aikaväleihin. Jokaisen testauskerran jälkeen aina jotain huomioitavaa ilmeni, jota ohjelmointivaiheessa ei ollut huomattu. Ohjelmoijan oma testaus ohjelmoitaessa tapahtuu kokoajan, ja sitä olisi hyvä tehdä niin monipuolisesti kuin mahdollista. Itse ohjelmoija saattaa tulla tavallaan sokeaksi ja ei itse huomaa kaikkia asioita miten sivuston saisi toimivuudeltaan parhaimmaksi. Toimeksiantajan testauskommenteista ja myös omaa testausta käyttäessä tuli huomattua monia virheitä sivuston toiminnallisuuksissa. Nämä virheet kirjattiin ja saatiin korjattua pienillä muutoksilla koodissa. Testausta tehtäessä on ainakin hyvä huomioida se, millaiseen käyttöön sivusto tulee tulevaisuudessa, ja miten sivustoa tullaan jälkeinpäin muokkaamaan. Näin ollen sivustoa täytyisi testata mahdollisimman suurella määrällä dataa ja erilaisilla tavoilla. (OAMK.)

## 7 YHTEENVETO

Video-oppimisympäristöt ovat yleistyneet opetuksessa valtavasti lähivuosina. Käänteinen opetus on tullut yhdeksi opetustavaksi kouluissa, joka monipuolistaa oppilaiden tapoja oppia. Oppimisympäristöjä on suuri määrä, ja niillä jokaisella on omat hyvät puolensa. (Suomen Kuvalehti 2012.)

Omassa työssäni oli tarkoitus toteuttaa video-oppimisympäristö tietyillä ominaisuuksilla, ja tavoitteena oli saada se liitettyä osaksi syventämään oppilaiden jo luennoilla saamiaan tietoja. Teoriaosuudessa vertailtiin ja tutkittiin muita vastaavia video-oppimisympäristöjä ja pohdittiin kuinka tämä vertailu vaikutti oman työni suunnitteluun ja toteutukseen. Lisäksi käsiteltiin HTML5-videosoittimen luominen käyttäen hyväksi JavaScriptin toiminnallisuuksia.

Kokonaisuudessaan opinnäytetyö eteni hyvin ja pysyttiin sovitus- aikataulussa. Haasteita ja vaikeuksia projektissa olivat itse ympäristön jälke- enpäin muokattavuuden toteuttaminen. Tärkeää oli se, että jälkikäyttäjä pystyy helposti vaihtamaan ja tuottamaan sisältöä ja, että sisältö on ulkoasultaan halutunlaisessa muodossa koskematta taustalla olevaan koodiin. Projektin suhteen päästiin tavoitteisiin ja saatiin aikaan vaatimusten mukaisen video-oppimisympäristö. Yhtenä ajatuksena suunnitteluvaiheessa oli myös mahdollisuus lisätä kyselyitä videon yhteyteen, mutta rajallisen aikataulun ja haasteellisuuden vuoksi tätä ominaisuutta ei sivustolle saatu lisättyä.

Jatko-kehitys mahdollisuuksia projektilla on paljonkin, kuten useamman videon upottaminen sivustolle näillä kaikilla toteutetuilla ominaisuuksilla. Lisäksi käyttäjätietojen tallentuminen toisi paljon lisäarvoa opettajille. Tarkoituksena olisi jatkaa kehittämistä ainakin sivuston ulkoasun ja käytettävyyden suhteen. Tarkoituksena olisi vielä tulevaisuudessa, että sivusto tulisi Turun yliopiston palvelimella ja toimisi lääketieteellisen tiedekunnan opiskelijoiden yhtenä apuvälineenä heidän opiskelussaan. Jos olisi mahdollista tehdä jotain toisin niin, olisin voinut käyttää suunnitteluun vielä enemmän aikaa, sillä toteutusvaiheessa eteni

tuli paljon ongelmatilanteita, jotka olisin voinut välttää laajemmalla suunnittelulla.

Video-oppimisympäristöjen käyttö on vielä suhteellisen vähäistä opetuksessa. On olemassa paljon videoita muun muassa Youtubessa ja Vimeossa monista eri aiheista, mutta vaihtuvan sisällön arvo videon aikana on suuri oppimisen kannalta. Ei tarvitse pelkästään kuunnella ja katsoa videota vaan voi vastata kyselyihin ja syventää videossa käsiteltävää aihetta lukemalla lisää aiheesta muualta klikkaamalla lisämateriaalilinkkejä muihin sivustoihin.

## LÄHTEET

Addison Wesley Longman Limited 1998 A history of HTML. Viitattu 1.4.2014 <http://www.w3.org/People/Raggett/book4/ch02.html>

EdLab 2014 Vialogues. Viitattu 14.4.2014 <https://vialogues.com/>

EduCanon 2014 Unleash Video's Potential. Viitattu 14.4.2014 <http://www.educanon.com/>

F+W Media 2014 Digital Book World: HTML5, The Future — And Now — of Publishing. Viitattu 2.4.2014 <http://www.digitalbookworld.com/2013/html5-the-future-and-now-of-publishing/>

Karukka, M. & Inkilä, T. 2013. Responsiivinen verkkosivujen suunnittelu mukauttaa sisällön eri päätelaitteille. ePooki. Oulun seudun ammattikorkeakoulun tutkimus- ja kehitystyön julkaisut 6. Viitattu 26.5.2014 <http://urn.fi/urn:nbn:fi-fe201302221895>

Knewton 2014 Flipped Classroom. Viitattu 16.4.2014 <http://www.knewton.com/flipped-classroom/>

LearnItIn5 2012 How-to video: YouTube interactive videos make learning fun. Viitattu 3.4.2014 <http://learnitin5.com/YouTube-interactive-videos>

MDN 2014 Using fullscreen mode. Viitattu 23.4.2014 [https://developer.mozilla.org/en-US/docs/Web/Guide/API/DOM/Using\\_full\\_screen\\_mode](https://developer.mozilla.org/en-US/docs/Web/Guide/API/DOM/Using_full_screen_mode)

MSDN Microsoft 2014 Developer Network Viitattu 22.4.2014 [http://msdn.microsoft.com/en-us/library/hh924820\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/hh924820(v=vs.85).aspx)

Metta 2014 Make'em Curious With Your Video Lessons. Viitattu 14.4.2014 <http://www.metta.io/>

OAMK Web-sovelluksen testaus. Viitattu 13.5.2014 [https://www.oamk.fi/sbc/testaus/web\\_sovelluksen\\_testaus.htm](https://www.oamk.fi/sbc/testaus/web_sovelluksen_testaus.htm)

ROME 2014 3 Dreams of Black. Viitattu 7.5.2014 <http://www.ro.me/>

SchoolMedia Interactive 2014. Viitattu 3.4.2014 <http://schoolmediainteractive.com/>

Suomen Kuvalehti 2012 Hyvästi piirtoheitin: Uudet oppimismenetelmät yleistyvät korkeakouluissa. Viitattu 2.6.2014 <http://suomenkuvalehti.fi/jutut/kotimaa/hyvasti-piirtoheitin-uudet-oppimismenetelmat-yleistyvat-korkeakouluissa/>

TED-Ed 2014 Lenssons Worth Sharing. Viitattu 14.4.2014 <http://ed.ted.com/>

TED-Ed 2014 What's new on TED-Ed. Viitattu 17.4.2014 <http://ed.ted.com/lessons/dear-subscribers#digdeeper>

Tuts Plus 2014 28 HTML5 Features, Tips, and Techniques you Must Know. Viitattu 3.4.2014 <http://code.tutsplus.com/tutorials/28-html5-features-tips-and-techniques-you-%20must-know--net-13520>

VideoNot 2013 The easiest way to take notes synchronized with videos. Viitattu 14.4.2014 <http://www.videonot.es/>

WIRED 2014 Mosaic Browser Lights Up Web With Color, Creativity. Viitattu 2.1.2014 <http://www.wired.com/2010/04/0422mosaic-web-browser/>

W3Schools 2014 HTML <video> Tag. Viitattu 3.4.2014 [http://www.w3schools.com/tags/tag\\_video.asp](http://www.w3schools.com/tags/tag_video.asp)

W3Schools 2014 HTML5 Introduction. Viitattu 2.4.2014  
[http://www.w3schools.com/html/html5\\_intro.asp](http://www.w3schools.com/html/html5_intro.asp)

Zaption 2014 Making Video Interactive. Viitattu 14.4.2014 [www.zaption.com](http://www.zaption.com)

3PlayMedia 2014 Services and features – overview. Viitattu 3.4.2014  
<http://www.3playmedia.com/services-features/overview/>

## JavaScript-funktiot

```

<script>
    var vid, playbtn, seekslider, curtimetext, durtimetext, mutebtn,
    volumeslider, fullscreenbtn;
    var cachedVolume, article;

    function initializePlayer(){
        // Set object references
        vid = document.getElementById("movies");
        playbtn = document.getElementById("playpausebtn");
        seekslider = document.getElementById("seekslider");
        curtimetext = document.getElementById("curtimetext");
        durtimetext = document.getElementById("durtimetext");
        mutebtn = document.getElementById("mutebtn");
        volumeslider = document.getElementById("volumeslider");
        fullscreenbtn = document.getElementById("fullscreenbtn");
        article = document.getElementById("testarticle");
        // Add event listeners
        playbtn.addEventListener("click", playPause, false);
        vid.addEventListener("click", playPause, false);
        seekslider.addEventListener("change", vidSeek, false);
        vid.addEventListener("timeupdate", seektimeupdate, false);
        mutebtn.addEventListener("click", vidmute, false);
        volumeslider.addEventListener("change", setvolume, false);
        fullscreenbtn.addEventListener("click", toggleFullScreen, false);
    }

    window.onload = initializePlayer;

    function playPause(){
        if(vid.paused){
            vid.play();
            playbtn.style.background = "url(wp-content/themes/neuroTheme/images/pause_button.png)";
        } else {
            vid.pause();
            playbtn.style.background = "url(wp-content/themes/neuroTheme/images/play_button.png)";
        }
    }

    function vidSeek(){
        var seekto = vid.duration * (seekslider.value / 100);
        vid.currentTime = seekto;
    }

    function seektimeupdate(){
        var nt = vid.currentTime * (100 / vid.duration);

```

```

seekslider.value = nt;
var curmins = Math.floor(vid.currentTime / 60);
var cursecs = Math.floor(vid.currentTime - curmins * 60);
var durmins = Math.floor(vid.duration / 60);
var dursecs = Math.floor(vid.duration - durmins * 60);
if(cursecs < 10){ cursecs = "0"+cursecs; }
if(dursecs < 10){ dursecs = "0"+dursecs; }
if(curmins < 10){ curmins = "0"+curmins; }
if(durmins < 10){ durmins = "0"+durmins; }
curtimetext.innerHTML = curmins+":"+cursecs;
durtimetext.innerHTML = durmins+":"+dursecs;
}

function vidmute(){
  if(vid.muted){
    volumeslider.value = cachedVolume;
    vid.muted = false;
    mutebtn.style.background = "url(wp-
content/themes/neuroTheme/images/mute_button.png)";
  } else {
    cachedVolume = volumeslider.value;
    vid.muted = true;
    mutebtn.style.background = "url(wp-
content/themes/neuroTheme/images/unmute_button.png)";
    volumeslider.value = 0;
  }
}

function setvolume(){
  vid.volume = volumeslider.value / 100;
}

function toggleFullScreen(){
  if(vid.requestFullScreen){
    vid.requestFullScreen();
  } else if(vid.webkitRequestFullScreen){
    vid.webkitRequestFullScreen();
  } else if(vid.mozRequestFullScreen){
    vid.mozRequestFullScreen();
  }
}

function sync() {
  $('.sliderimgLi img').attr('style', 'padding: 0px !important; width: 50px; height: 50px;');
  $('.sliderimgLi img').addClass( "snapShotImg" );
  var now = 0;
  var timestamps = [],
      snapshotStamps = [],
      last = 0,
      all = 0,

```

```

        snapShots = 0,
        old = 0,
        i = 0,
        j = 0;;

        $('article').each(function() {
            if ($(this).attr('data-start').length == 5) {
                var startMin = parseInt($(this).attr('data-
start').substring(0,2) * 60);
                var startSec = parseInt($(this).attr('data-
start').substring(3,5));
                var artStart = startMin + startSec;

                var endMin = parseInt($(this).attr('data-
end').substring(0,2) * 60);
                var endSec = parseInt($(this).attr('data-
end').substring(3,5));
                var artEnd = endMin + endSec;

                timestamps.push({
                    start : +artStart,
                    end : +artEnd,
                    elm : $(this)
                })
            }
        });

        all = timestamps.length;

        $('img').each(function() {
            if ($(this).attr('title')) {
                var imgStartMin = par-
seInt($(this).attr('title').substring(0,2) * 60);
                var imgStartSec = par-
seInt($(this).attr('title').substring(3,5));
                var imgStart = imgStartMin + imgStartSec;

                var imgEndMin = par-
seInt($(this).attr('title').substring(6,8) * 60);
                var imgEndSec = par-
seInt($(this).attr('title').substring(9,11));
                var imgEnd = imgEndMin + imgEndSec;

                snapshotStamps.push({
                    first : +imgStart,
                    last : +imgEnd,
                    ment : $(this)
                })
            }
        });
    });

```

```

        snapShots = snapshotStamps.length;

        $(".sliderimgLi img").click(function () {
            var imgStartMin = par-
seInt($(this).attr('title').substring(0,2) * 60);
            var imgStartSec = par-
seInt($(this).attr('title').substring(3,5));
            var imgStart = imgStartMin + imgStartSec;

            vid.currentTime = imgStart;

        });

        vid.addEventListener("timeupdate",function(event){
            var fullSec = parseInt(vid.currentTime)
            var now = fullSec;
            if (fullSec >= now) {
                showsection(now, all, timestamps, snapShots,
snapshotStamps);
                now++;
            }
        },false);
    }

    function showsection(now, all, timestamps, snapShots,
snapshotStamps) {
        for(i=0;i<all;i++){
            if(now >= timestamps[i].start && now <= times-
tamps[i].end){
                timestamps[i].elm.addClass( "testarticleCurrent" );
            } else {
                timestamps[i].elm.removeClass( "testarticleCurrent"
);
            }
        }
        for(j=0;j<snapShots;j++){
            if(now >= snapshotStamps[j].first && now <= snapshot-
Stamps[j].last){
                snapshotStamps[j].ment.addClass( "snapShotCurrent"
);
            } else {
                snapshotStamps[j].ment.removeClass( "snapShotCur-
rent" );
            }
        }
    }

    function popUp(id) {
        vid.pause();
        playbtn.style.background = "url(wp-
content/themes/neuroTheme/images/play_button.png)";
    }

```

```

        $('#content' + id).css({'display':'block'})
        $('#opacityBack' + id).css({'display':'block'})
        $('#popUp' + id).css({'display':'block'})
    }

    function closepopup(id){
        vid.play();
        playbtn.style.background = "url(wp-
content/themes/neuroTheme/images/pause_button.png)";
        $('#popUp'+ id).hide();
        $('#opacityBack' + id).css({'display':'none'});
        $('#content' + id).css({'display':'none'})
    }

    function doSearchHaku() {

        //Vasemman puoleinen hakulaatikko (kategorian mukaan)
        var q = document.getElementById("hakuLeft");
        var v = q.value.toLowerCase();
        var rows = document.getElementsByClassName("hakurivitLeft");
        var m = 0;
        var nm = document.getElementById("noresultsLeft");
        var leftDispArray = [];

        for ( var i = 0; i < rows.length; i++ ) {
            var fullname = rows[i].getElementsByTagName("a");
            fullname = fullname[0].innerHTML.toLowerCase();
            if ( fullname ) {
                if ( v.length == 0 || (v.length < 1 && fullna-
me.indexOf(v) == 1) || (v.length >= 1 && fullname.indexOf(v) > -1 ) )
            {
                rows[i].style.display = "";
                leftDispArray.push(rows[i].id);
                m++;
            } else {
                rows[i].style.display = "none";
            }
        }
        if ( m == 0 && nm ) {
            nm.style.display = "";
            document.getElementById("noresLeft").innerHTML =
q.value;
        } else {
            nm.style.display = "none";
        }
    }

    //Oikean puoleinen hakulaatikko (post sisällön mukaan)
    var g = document.getElementById("haku");
    var w = g.value.toLowerCase();
    var rowsOther = document.getElementsByClassName("hakurivit");

```

```

var n = 0;
var nv = document.getElementById("noresults");
var rightDispArray = [];

for ( var j = 0; j < rowsOther.length; j++ ) {
    var insertti = rowsOther[j].getElementsByName("a");
    insertti = insertti[0].innerHTML.toLowerCase();
    if ( insertti ) {
        if ( w.length == 0 || (w.length < 1 && insert-
ti.indexOf(w) == 1) || (w.length >= 1 && insertti.indexOf(w) > -1 ) )
        {
            rowsOther[j].style.display = "";
            rightDispArray.push(rowsOther[j].id);
            n++;
        } else {
            rowsOther[j].style.display = "none";
        }
    }
    if ( n == 0 && nv ) {
        nv.style.display = "";
        document.getElementById("nores").innerHTML = g.value;
    } else {
        nv.style.display = "none";
    }
}
compareLists(v, leftDispArray, rightDispArray);
compareListsRight(w, leftDispArray, rightDispArray);
}

function compareLists (v, leftDispArray, rightDispArray) {
    if ( v.length != 0 ) {
        $(".hakurivit").hide();
        for (var i = 0; i < leftDispArray.length; i++) {
            for (var j = 0; j < rightDispArray.length; j++) {
                if (leftDispArray[i] == rightDispArray[j]) {
                    var x = rightDispArray[j];
                    $("#"+ x + ".hakurivit").show();
                }
            }
        }
    }
    else {
        for (var j = 0; j < rightDispArray.length; j++) {
            $('.' + rightDispArray[j]).show();
        }
    }
}

function compareListsRight (w, leftDispArray, rightDispArray) {
    if ( w.length != 0 ) {
        $(".hakurivitLeft").hide();
    }
}

```

```
    for (var i = 0; i < rightDispArray.length; i++) {
        for (var j = 0; j < leftDispArray.length; j++) {
            if (leftDispArray[j] == rightDispArray[i]) {
                var x = leftDispArray[j];
                //console.log("#"+ x + ".hakurivitLeft");
                $("#"+ x + ".hakurivitLeft").show();
            }
        }
    }
}
else {
    for (var j = 0; j < leftDispArray.length; j++) {
        $("#."+ leftDispArray[i]).show();
    }
}
} </script>
```