

Il formato EPUB 3

Editoria digitale

Maria A. Alberti
Informatica per la comunicazione digitale
AA 17/18

HTML5 i tag nuovi...

- Tags per definire i contenuti
 - `audio`, `video`, `embed`, `canvas`
- Tags per definire la struttura del documento
 - `article`, `aside`, `figure`, `header`, `hgroup`, `footer`, `nav`, `section`, `figcaption`
- Tags vari
 - `meter`, `progress`, `time`, `details`, `command`, `menu`, `datalist`, `mark`, `output`, `rp`, `ruby`, `source`, `summary`, `keygen`
- Integrazione di API
 - `Audio/video`, `offline application`, `inline editing`, `drag & drop`, `history`, `protocols`

Le risorse multimediali

- Definito su HTML5, EPUB 3 inserisce le risorse *audio* e *video* nel flusso del documento
- Per l'audio è richiesto il formato **mp3**
- Per il video le cose sono un poco più complesse. Al momento sono supportati i formati **H.264** e **WebM**.
 - H.264 è molto diffuso ma potrebbe essere coperto da diritti
 - WebM è libero da diritti ma non è molto diffuso e non è neppure chiaro se lo diventerà
 - Quindi per essere sicuri che un video sia leggibile dai lettori, che supportano alcuni ma non tutti i formati video, occorre stabilire il proprio target di lettori

Le risorse multimediali - 2

- Come sempre, il problema delle risorse multimediali sta nella loro dimensione
- Per questo, unica eccezione in EPUB, le risorse possono anche risiedere fuori dal contenitore
 - Se si mette tutto nel contenitore i tempi di download e di upload sui lettori può essere lungo ma l'esperienza di lettura senza interruzioni
 - Se le risorse risiedono fuori sul web, il download è veloce si inizia subito a leggere ma si possono incontrare problemi nell'apertura delle risorse con lettori non sempre connessi o canali di distribuzione non sempre disponibili

Inserire audio

```
<audio id="motivo">
  <source src="motivo.wav" type="audio/wav"/>
  <source src="motivo.mp3" type="audio/mp3"/>
  <source src="motivo.ogg" type="audio/ogg"/>
  <em>audio non supportato dal lettore</em>
</audio>
```

Elemento **audio** è un contenitore di 3 file audio di diverso formato, come specificato dall'attributo **type**. Il lettore esegue il primo file del formato che supporta e se non ne trova rimanda all'istruzione di fallback

Inserire audio - 2

Se il file audio è solo di formato mp3, l'istruzione può essere semplificata in:

```
<audio id="motivo" src="motivo.mp3" type="audio/mp3"/>
  <em>audio non supportato dal lettore</em>
</audio>
```

Per controllare l'esecuzione del file sono disponibili alcuni attributi booleani

- controls** visualizza una barra di controllo con i bottoni per l'esecuzione
- autoplay** l'esecuzione avviene automaticamente appena il file è caricato
- loop** l'esecuzione è ripetuta in ciclo

Inserire audio – 3

Se il file audio è solo di formato mp3, l'istruzione può essere semplificata in:

```
<audio id="motivo" src="motivo.mp3" type="audio/mp3"/>  
  <em>audio non supportato dal lettore</em>  
</audio>
```

Gli attributi booleani sono previsti da HTML5 ma non supportati da molti lettori. Quindi è consigliabile usare la precedente sintassi valida in EPUB

```
controls = "controls"  
autoplay = "autoplay"  
loop = "loop"
```

Inserire video

```
<video id="clip" width="300" height="300">  
  <source src="clip.mp4" type="video/mp4"/>  
  <source src="clip.webm" type="video/webm"/>  
  <em>video non supportato dal lettore</em>  
</video>
```

Elemento **video** contiene 2 file video di diverso formato, come specificato dall'attributo **type**. Il lettore esegue il primo file del formato che supporta e se non ne trova rimanda all'istruzione di fallback

Come per l'elemento audio è possibile la forma semplificata in presenza di un unico file e degli attributi booleani o comunque degli attributi per il controllo dell'esecuzione

Inserire video - 2

Se la risorsa **video** è remota:

```
<video id="clip" src="http://..." />
```

Nel caso di risorsa video o audio remota il file che lo riferisce deve dichiararlo nella sezione **manifest** del content document tramite l'attributo **properties**

```
<item id="capX"
  src="capX.xhtml"
  media-type="application/xhtml+xml"
  properties="remote-resources"/>
```

E il file clip audio o video deve essere in lista anche se esterno

```
<item id="clip"
  src="http://....webm"
  media-type="video/webm"/>
```

HTML5 supporta SVG

- Il frammento SVG va inserito nel file XHTML
 - Per inclusione
 - O mediante link a file esterno
- Il frammento SVG deve essere dichiarato nella sezione **manifest** del file **content.opf**

```
<manifest>
  <item id="cap1"
    href="capitolo1.xhtml"
    media-type="application/xhtml+xml"
    property="svg"
    media-overlay="cap1_audio"/>
```

Media overlays

- Oltre alle risorse multimediali integrate nel testo, si possono avere risorse multimediali sincronizzate con il testo: un modo per avere uno strato MM sopra i contributi tradizionali.
 - Parole evidenziate durante la lettura, audio sovrapposto al testo
 - Mentre si esegue l'audio uno stile CSS è applicato alla parte di testo corrispondente
 - In futuro si potrà anche sovrapporre video al testo
- Se l'audio è supportato dai lettori *dovrebbero* esserlo anche i media overlays

Media overlays - 2

Si realizza via un profilo SMIL

- L'informazione dell'overlay è completamente separata dal contenuto
- Diversamente dall'attuale metodo adottato da DAISY (Digital Talking Book)
- Una specie di mappa che stabilisce la corrispondenza a diversi livelli di granularità tra testo e audio.
- Richiede un tag di markup per ogni oggetto della granularità desiderata
- La sincronizzazione è effettuata da SMIL tramite l'indicazione del tempo della traccia audio e del testo

I tag principali implicati:

- <par>** eventi paralleli
- <seq>** eventi in sequenza

Esempio

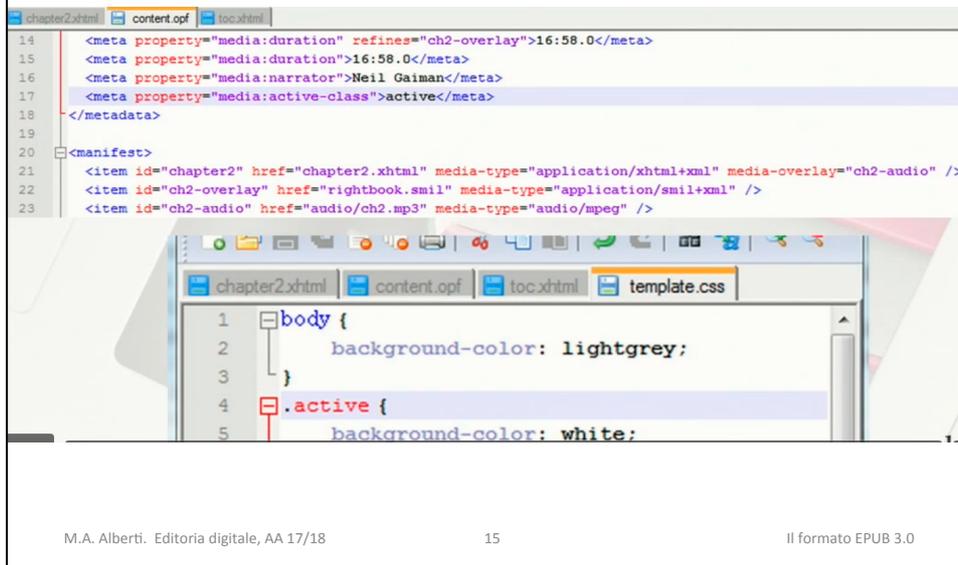
Decisa la granularità, marcati con identificatori ogni oggetto da sincronizzare

```
<section id="ch2" epub:type="chapter">
  <p class="chapter" id="title">The Right Book </p>
  <p class="body" id="sent1">Now (-ish) </p>
  <p class="body"><span id="sent2">The thing that Arthur liked best about
owning his own shop was that he could stock whatever he pleased, and if
you didn't like it, you could just shop somewhere else.</span> <span id=
"sent3">So there in the window were four ancient Cluedo sets rescued
from a car-boot sale in Sussex; a pair of trousers sewn from a salvaged
WWII bivouac tent; a small card advertising the availability of
artisanal truffles hand-made by an autistically gifted chocolatier in
Islington; a brick of Pu'er tea that had been made in Guyana by a
Chinese family who'd emigrated a full century previous; and, just as of
now, six small, handsomely made books.</span> </p>
```

SMIL file

```
<smil xmlns="http://www.w3.org/ns/SMIL" xmlns:epub="http://www.idpf.org/2007/ops"
version="3.0">
  <body>
    <seq id="ch2" epub:textref="chapter2.xhtml#ch2" epub:type="chapter">
      <par id="title">
        <text src="chapter2.xhtml#title" />
        <audio src="audio/ch2.mp3" clipBegin="0:0s" clipEnd="0:24.50s" />
      </par>
      <par id="sent1">
        <text src="chapter2.xhtml#sent1" />
        <audio src="audio/ch2.mp3" clipBegin="0:24.50s" clipEnd="0:28.50s" />
      </par>
      <par id="sent2">
        <text src="chapter2.xhtml#sent2" />
        <audio src="audio/ch2.mp3" clipBegin="0:28.50s" clipEnd="0:39.0s" />
      </par>
      <par id="sent3">
        <text src="chapter2.xhtml#sent3" />
        <audio src="audio/ch2.mp3" clipBegin="0:39.0s" clipEnd="0:01:11.7" />
      </par>
      <par id="sent4">
        <text src="chapter2.xhtml#sent4" />
        <audio src="audio/ch2.mp3" clipBegin="0:01:11.7" clipEnd="0:01:14" />
      </par>
    </seq>
  </body>
</smil>
```

Modificare content.opf e css



The image shows a code editor with two files open: content.opf and template.css. The content.opf file contains the following code:

```
14 <meta property="media:duration" refines="ch2-overlay">16:58.0</meta>
15 <meta property="media:duration">16:58.0</meta>
16 <meta property="media:narrator">Neil Gaiman</meta>
17 <meta property="media:active-class">active</meta>
18 </metadata>
19
20 <manifest>
21 <item id="chapter2" href="chapter2.xhtml" media-type="application/xhtml+xml" media-overlay="ch2-audio" />
22 <item id="ch2-overlay" href="rightbook.smil" media-type="application/smil+xml" />
23 <item id="ch2-audio" href="audio/ch2.mp3" media-type="audio/mpeg" />
```

The template.css file contains the following code:

```
1 body {
2   background-color: lightgrey;
3 }
4 .active {
5   background-color: white;
```

At the bottom of the slide, there is a footer with the following text:

M.A. Alberti. Editoria digitale, AA 17/18 15 Il formato EPUB 3.0