Theme 9

Publishing digitally with XML

XHTML

- E**x**tensible **H**yper**T**ext **M**arkup **L**anguage

- Identical to HTML (Stricter)

- Immediate popularity of XML…
- …led to the idea of redefining HTML in terms of XML.

- The idea is to help development make transition from HTML to XML.

Benefit of XHTML

- Document can be processed by regular XML tools
- Documents can be edited in XML-aware software
- Can contain “non-HTML” content
  - XML vocabulary (MathXML, SVG)
- **Does not replace but sit along side it**

Writing an XHTML

- Start with a <!DOCTYPE>

- Strict, Transitional, Formal and 1.1

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

- The xmlns must be provided in the <head> element.

HTML vs XHTML

- Elements that are mandatory
  - Html, head, title, body

- XML gramma rules applies to all HTML elements
  - Element **must** be in lower case

- XML naming conventions applies to attribute
  - Names **must** be in lower case
  - Minimization is **forbidden**

Convert from HTML to XHTML

- Add an XHTML <!DOCTYPE> to the first line of every page
- Add an xmlns attribute to the html element of every page
- Change all element names to lowercase
- Close all empty elements
- Change all attribute names to lowercase
- Quote all attribute values

Why use XHTML

- XHTML documents are validated with standard XML tools.
- It is easily to maintain, convert, edit document in the long run.
- It is used to define the quality standard of web pages.
- XHTML is an official standard of the W3C, your website becomes more compatible and accurate with many browsers.

EPUB 3

- **E**lectronic **Pub**lication (latest version 3.2)
- The epub format is an open standard for e-books created by the International Digital Publishing Forum (IDPF)
- A file format for publishing books and other content in a “reflowable” fashion.
- Uses **XHTML** to **construct** the content of a publication.

EPUB 3

- EPUB 3 is the second major revision of the standard.
- HTML5 and CSS3 adoption
  - Caters to new elements introduced in the specifications
- Improved navigation
  - Uses HTML drop-down from NCX
- Ability to add media as well as scripting
  - Add support for video, audio and JavaScript

EPUB 3

- More sophisticated metadata
  - Much richer, standardized amount of metadata
- The ability to create fixed-format eBooks
  - Files that mimicked the look and layout of paper-and-ink book
- Backward compatibility

Specification that makes EPUB work

- **Open Packaging Format** (OPF)
  - This part of the specs deals with structural information such as metadata, the manifest and the table of contents. These data are all embedded using XML.
- **Open Container Format** (OCF)
  - The OCF specs define how all those files end up being packaged in one single container file. ZIP compression is used for this. If you take an EPUB file and change the .epub extension to .zip, you can decompress the publication and take a look at all those files.
- **EPUB Content Documents**
  - This specification defines profiles of HTML, SVG, and CSS for specific use in the context of EPUB

Open Container Format

- Mimetype file - Format of the publication

```
Application/epub+zip
```

- META-INF folder – A single container.xml
  - container files, signatures, encryption, rights, in the format of xml files

- OEBPS Content folder – Main content of EPUB

Open Container Format (OEBPS folder)

- TOC file
  - An XHTML file that contains the table of content file

- OPF file (Open Packaging Format)
  - An XML file that describes an EPUB publication.
  - It identifies all other files in the publication and provides descriptive information about them.

- Container
  - An XML file that translate the location of the OPF file to the reading system.

Example of TOC file

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:epub="http://www.idpf.org/2007/ops">
      <head>
            <title>Accessible EPUB 3</title>
            <link rel="stylesheet" type="text/css“ href=“…" />
      </head>
      <body>
            <nav epub:type="toc" id="toc">
            </nav>
            ...
```

Open Packaging Format (OPF)

- The Package Document (OPF file) specifies all the publication's content, required resources, reading order, and navigation information. The structure of the XML document requires (not limited to) three main element:
  - A **metadata** element
    - Including and/or referencing metadata applicable to the entire publication and particular resources within it.
  - A **manifest** element
    - Identifies and describes the set of resources that constitute the EPUB Publication.
  - A **spine** element
    - Defines the default reading order of the publication.

Example of OPF

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<package xmlns="http://www.idpf.org/2007/opf"
      xmlns:dc="http://purl.org/dc/elements/1.1/"
      unique-identifier="pub-identifier">
      <metadata>
            <!—Dublin core metadata -->
      </metadata>
      <manifest>
            <!—All content use must be define here-->
      </manifest>
      <spine>
            <!—Ordering the Epub-->
      </spine>
</package>
```

Create an EPUB from HTML

- Build your content
  - Convert your html files to XHTML
- Create a MIME Type File
- Add additional content needed
  - CSS, images, fonts, etc.
- Build Your Table of Contents
- Add a container XML file
- Create the OPF

SVG

- **SVG** stands for **S**calable **V**ector **G**raphics
- SVG defines vector-based graphics in XML format.

```
<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4‟ fill="yellow" />
</svg>
```

Benefits of SVG

SVG-based images **do not lose quality** when resized

SVG images can be animated using JavaScript or CSS

SVG images integrate well with DOM very well

SVG images can be indexed by search engines, which is perfect for SEO purposes

Creating SVG

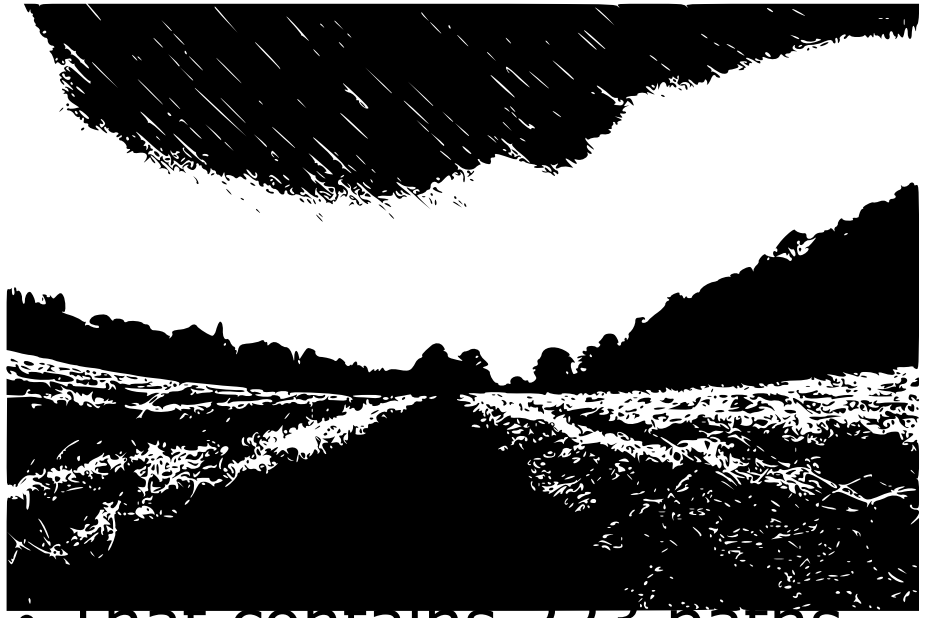
- SVG are created with the combination of eight different shape type
  - squares, rectangles, circles, ellipses, lines, polylines, polygons and paths,

- These shape are be essentially divided into:
  - `<rect/> <ellipse/> <line/> <polyline/> <polygon/> <path/>`

- SVG can be defined anywhere within your HTML page

SVG Sample

```
<rect x="50" y="20"
rx="20" ry="20"
width="150" height="150"
style="fill:red;stroke:black;stroke-
width:5;opacity:0.5" />
```



- That contains 223 paths
- 1700 lines of code

Animating SVGs

- SVG is an xml file (think DOM)

- Add script (JS) that directly changes the SVG model
- **or**
- For more simple animation use the built-in animation tag
  - `<animateTransform>`

Animate Example

```
<svg width="120" height="120" viewBox="0 0 120 120“
xmlns="http://www.w3.org/2000/svg">
    <polygon points="60,30 90,90 30,90">
        <animateTransform attributeName="transform"
                          attributeType="XML"
                          type="rotate"
                          from="0 60 70"
                          to="360 60 70"
                          dur="10s"
                          repeatCount="indefinite"/>
    </polygon>
</svg>
```